Operations and Evaluation Measures for Learning Possibilistic Graphical Models

Christian Borgelt and Rudolf Kruse

Dept. of Knowledge Processing and Language Engineering
School of Computer Science
Otto-von-Guericke-University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany

Abstract: One focus of research in graphical models is how to learn them from a dataset of sample cases. This learning task can pose unpleasant problems if the dataset to learn from contains imprecise information in the form of sets of alternatives instead of precise values. In this paper we study an approach to cope with these problems, which is not based on probability theory as the more common approaches like, e.g., expectation maximization, but uses possibility theory as the underlying calculus of a graphical model. Since the search methods employed in a learning algorithm are relatively independent of the underlying uncertainty or imprecision calculus, we focus on evaluation measures (or scoring functions).

Keywords: graphical models, possibilistic networks, learning from data, evaluation measures

1 Introduction

In recent years graphical models [Whittaker 1990, Kruse *et al.* 1991, Lauritzen 1996] have become increasingly popular as means to structure complex domains and thus to facilitate reasoning in such domains. The idea underlying them is that under certain conditions, namely if conditional independences hold, a (probability or possibility) distribution on a high-dimensional domain can be decomposed into a set of (overlapping) distributions on subspaces, from which the whole distribution can be reconstructed. This decomposition and the conditional independences that make it possible are represented by a graph—hence the name "graphical model". In this graph there is a node for each attribute used to describe the domain under consideration. Edges connect attributes that are directly dependent on each other and also indicate the paths along which evidence has to be propagated if inferences are to be drawn from observations.

Among the best-known approaches in this direction are probabilistic graphical models like Bayes networks [Pearl 1988] and Markov networks [Lauritzen and Spiegelhalter 1988], but also the more general valuation-based networks [Shenoy 1992]. More recently, graphical models have also been studied with possibility theory as the underlying uncertainty calculus [Gebhardt and Kruse 1996a]. All of these approaches led to efficient implementations, for example HUGIN [Andersen *et al.* 1989], PATHFINDER [Heckerman 1991], PULCINELLA [Saffiotti and Umkehrer 1991], and POSSINFER [Gebhardt and Kruse 1996a].

Since it can be tedious and time consuming if human experts have to construct a graphical model "manually", a large part of recent research has been devoted to learning graphical models automatically from a dataset of sample cases. Despite the fact that some important instances of this learning task have been shown to be NP-hard in the general case [Dechter and Pearl 1992, Chickering *et al.* 1994], mainly because of the huge space of possible decompositions that has to be searched, several heuristic algorithms have been developed that lead to highly promising results in example applications [Cooper and Herskovits 1992, Heckerman *et al.* 1995, Gebhardt and Kruse 1995, Jordan 1998].

Early learning approaches, however, were restricted to learning from *precise* data. By this we mean that the tuple describing a sample case must not contain missing values or set-valued information: There must be exactly one value for each of the attributes used to describe the domain under consideration. In applications this condition is rarely met, though: Databases are notoriously incomplete and useful imprecise information, in the sense of a set of values for an attribute, is frequently available (even though it is often neglected, because standard database implementations cannot handle it adequately). Hence researchers were faced with the challenge to extend the existing learning algorithms to incomplete and imprecise data.

If probabilistic graphical models are studied, it is tried to meet this challenge with approaches that are based on the expectation maximization (EM) algorithm or on gradient descent [Dempster *et al.* 1977, Jamshidian and Jennrich 1993, Bauer *et al.* 1997, Friedman 1998]. Although these approaches are very promising, they have several drawbacks: In the first place, they are iterative procedures which can converge very slowly. This is especially true of the EM algorithm, which is known to converge slowly in the vicinity of the convergence point. Gradient descent is usually faster, but does not share the robustness of the EM algorithm. These disadvantages can be mitigated by using the EM algorithm first until a point close enough to the convergence point is reached, so that it is safe to switch to the faster gradient descent. Another drawback of these approaches is that they are costly to implement if a data tuple contains several missing values. In this case a joint distribution for all attributes with a missing value has to be maintained and estimated for the tuple. This leads to complex data structures to store the data to learn from and leads to a time complexity for each iteration that grows exponentially with the number of missing values per tuple.

Therefore we explore a different path in this paper, namely graphical models that are based on possibility theory [Gebhardt 1997, Borgelt 2000, Borgelt and Kruse 2002]. It has turned out that with this type of graphical models imprecise information can be handled very conveniently, so that learning from an imprecise dataset can easily be accomplished. No iterative procedure is necessary to estimate the needed possibility distributions and no complex data structures for the data tuples are needed to accomplish learning.

Of course, learning algorithms for possibilistic graphical models borrow heavily from the corresponding probabilistic algorithms. For example, the search methods employed can usually be transferred directly to the possibilistic case. However, due to the fact that the underlying uncertainty calculus differs, special evaluation measures (or scoring functions) have to be developed. This is the main subject of this paper: After a brief review of the theory of possibilistic graphical models we study several evaluation measures that can be used for learning possibilistic networks from data.

2 Graphical Models

As already indicated, graphical models describe decompositions of distributions. In the following, we first review the idea of decomposition and then how decompositions can be represented by graphs. Finally, we study a very simple example of a possibilistic network.

2.1 Decomposition

Decomposition is one of the key subjects in the theory of relational databases [Ullman 1988] and thus it is not surprising that relational database theory is closely connected to the theory of graphical models. In relational database theory it is studied whether a (high-dimensional) relation is *join-decomposable*, so that it can be stored with less redundancy and, of course, using less storage space. The idea is that a relation can often be reconstructed from certain *projections* by forming their *natural join*.

Formally, this can be described as follows: Let $U = \{A_1, \ldots, A_n\}$ be a set of attributes describing the modeled section of the world and let $\mathrm{dom}(A_i)$ be their respective domains. Furthermore, let r_U be a relation over U. Since it simplifies the transfer to probabilistic and possibilistic graphical models, we represent this relation by its *indicator function*, which is 1 for all tuples contained in the relation and 0 for all other tuples. The tuples themselves we represent as logical conjunctions $\bigwedge_{A_i \in U} A_i = a_i$ stating a value for each of the attributes.[1] Then a projection r_M of the relation r_U to a subset M of the attributes in U can be defined as

$$r_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big) = \max_{\substack{\forall A_j \in U-M: \\ a_j \in \mathrm{dom}(A_j)}} r_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big),$$

where the unusual notation w.r.t. the maximum means that the maximum has to be taken over all values of all attributes in $U - M$. With this notation a relation r_U is called *join-decomposable* w.r.t. a family $\mathcal{M} = \{M_1, \ldots, M_m\}$ of subsets of U iff

$$\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) :$$
$$r_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big) = \min_{M \in \mathcal{M}} r_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big).$$

Note that the minimum computed here is equivalent to the natural join of relational algebra. It is obvious that in such a situation it suffices to store the projections r_M in order to capture all information contained in the relation r_U, because the original relation can always be reconstructed.

The decomposition scheme just outlined is easily transferred to the probabilistic case: We only have to replace the projection and the natural join by the proper probabilistic operations. Thus we arrive at the following decomposition formula for a given probability distribution p_U:

$$\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) :$$
$$p_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big) = \prod_{M \in \mathcal{M}} \phi_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big).$$

[1] This representation of tuples and relations is more convenient for our purposes than the usual Cartesian product definition, since we do not need index mapping functions to describe projections.

Here the ϕ_M are functions that can be computed from the marginal distributions on the sets M of attributes (computed by summing over the values of the removed attributes, instead of taking the maximum as in the relational case), which shows that marginalization takes the place of projection. These functions are usually called *factor potentials* [Castillo *et al.* 1997]. From this formula we can also see that in the probabilistic case the minimum is replaced by the product.

The possibilistic case is even closer to the relational one. The decomposition formula is identical (except that we denote the possibility distribution by π instead of r):

$$\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) :$$
$$\pi_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big) = \min_{M \in \mathcal{M}} \pi_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big),$$

and the operation used to compute the projections π_M is also the same as in the relational case, namely the maximum. Thus the only difference is that the *possibility distributions* π_U and π_M are not restricted to values 0 and 1 as the indicator function we used to describe relations, but can assume any value in the interval $[0, 1]$. In this way a gradual possibility of a tuple is modeled and possibilistic graphical models can be developed as "fuzzifications" of relational graphical models.

2.2 The Context Model

Working with "gradual possibilities" raises, of course, the question of their interpretation, because in natural language the notion "possible" is two-valued: either a state, a situation, a circumstance etc. is possible or it is impossible, and hence there is no intuitive understanding of *degrees* of possibility. Therefore it is advantageous to touch at least briefly upon the interpretation provided by the *context model* [Gebhardt and Kruse 1993, Gebhardt and Kruse 1996a], on which we base our theory of possibilistic graphical models (a much more detailed exposition can be found in [Borgelt 2000, Borgelt and Kruse 2002]):

Suppose that for a description of the section of the world to be modeled we can distinguish between a set $C = \{c_1, \ldots, c_k\}$ of contexts. These contexts may be given, for example, by physical or observation-related frame conditions. Furthermore, suppose that we can assess the relative importance or frequency of occurrence of these contexts by assigning a probability $P(c)$ to each of them. Finally, suppose that we can state for each context c a set $\Gamma(c)$ of possible states—described by tuples, see above—the section of the world may be in under the conditions that characterize the context. We assume each set $\Gamma(c)$ to be the *most specific correct set-valued specification* of the state t_0 of the modeled section of the world, which we can give for the context c. By this we mean that we are sure that $\Gamma(c)$ contains t_0, but that we cannot state with certainty that a proper subset of $\Gamma(c)$ contains t_0, regardless of what subset we choose. Given these ingredients, we define the *degree of possibility* that a tuple t describes the actual state t_0 of the modeled section of the world as the weight (probability) of all contexts in which t is possible.

The above description can be made formally precise with the notion of a *random set* (i.e. a set-valued random variable) $\Gamma : C \to 2^T$, where T is the set of all possible tuples. This random set maps contexts to the so-called *focal sets* $\Gamma(c) \subseteq T$ and thus is an imperfect (i.e. imprecise and uncertain) specification of the actual state t_0 of the modeled section of the world. We can derive a *possibility distribution* from it by simply computing its

one-point coverage, which is defined as

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

In this interpretation possibility distributions represent uncertain *and* imprecise knowledge as can be seen by comparing them to probability distributions and to relations. 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 $\forall c \in C : |\Gamma(c)| = 1$, i.e., if for all contexts the specification of t_0 is precise. On the other hand, a relation represents *imprecise*, but *certain* knowledge. Thus, not surprisingly, a relation can also be seen as a special case of a possibility distribution in the interpretation given above, 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 described by a possibility distribution.

2.3 Graphical Representation

Graphs (in the sense of graph theory) are a very convenient tool to describe decompositions if we identify each attribute with a node of a graph. In the first place, graphs can be used to specify the sets M of attributes underlying the decomposition. How this is done depends on whether the graph is directed or undirected. If it is undirected, the sets M are the maximal cliques of the graph, where a clique is a complete subgraph and it is maximal if it is not contained in another complete subgraph.

If the graph is directed (but acyclic), we can be more explicit about the distributions in the decomposition: We can use conditional distributions, since we may use the direction of the edges to distinguish between the conditioned attribute and the conditions. Note, however, that this does not make much of a difference in the relational and the possibilistic case, because here we simply identify the conditional distributions with the corresponding marginal distributions, i.e.,

$$\pi\Big(A_j = a_j \Big| \bigwedge_{A_i \in M} A_i = a_i\Big) = \pi\Big(A_j = a_j \wedge \bigwedge_{A_i \in M} A_i = a_i\Big).$$

Of course, this is only one of several possibilities to define a conditional degree of possibility. However, it is the only one that can be justified with the context model, on which we base our theory of possibilistic graphical models. A detailed discussion, which is beyond the scope of this paper, can be found in [Borgelt 2000, Borgelt and Kruse 2002].

Secondly, graphs can be used to represent (conditional) dependence and independence relations between attributes via the concept of *node separation*. What is to be understood by "node separation" depends again on whether the graph is directed or undirected. If it is undirected, it is defined as follows: If X, Y, and Z are three disjoint subsets of nodes in an undirected graph, then Z separates X from Y iff after removing the nodes in Z and their associated edges from the graph there is no path, i.e. no sequence of consecutive edges, from a node in X to a node in Y. In other words: Z separates X from Y iff all paths from a node in X to a node in Y contain a node in Z.

For directed acyclic graphs the so-called *d-separation criterion* is used [Pearl 1988, Verma and Pearl 1990]: If X, Y, and Z are three disjoint subsets of nodes in a directed acyclic graph, then Z is said to *d-separate* X from Y iff there is no path, i.e. no sequence of consecutive edges (of any directionality), from a node in X to a node in Y along which the following two conditions hold:

1. every node, at which edges of the path converge (i.e., both edges are directed towards the node), either is in Z or has a descendant in Z,
2. every other node is not in Z.

These separation criteria may be used to define *conditional independence graphs*: A graph is a conditional independence graph w.r.t. a given multi-dimensional distribution if it captures by node separation only correct conditional independences between sets of attributes. Conditional independence means (for three attributes A, B, and C with A independent of B given C—the generalization to sets of attributes is obvious) that

$$P(A = a, B = b \mid C = c) \quad = \quad P(A = a \mid C = c) \cdot P(B = b \mid C = c)$$

in the probabilistic case and

$$\pi(A = a, B = b \mid C = c) \quad = \quad \min\{\pi(A = a \mid C = c), \pi(B = b \mid C = c)\}$$

in the possibilistic and the relational case. The latter is also well known under the name of *possibilistic non-interactivity* [Dubois and Prade 1988].

These formulae already indicate the close connection of conditional independence and decomposability (cf. section 2.1), because they are possible decomposition formulae for a three-dimensional probability or possibility distribution, respectively, on the space scaffolded by the attributes A, B, and C. Formally, the connection between conditional independence graphs and graphs that describe decompositions is brought about by theorems that show that a distribution is decomposable w.r.t. a given graph if and only if this graph is a conditional independence graph of the distribution. For the probabilistic setting, this theorem is usually attributed to [Hammersley and Clifford 1971], who proved it for the discrete case, although (according to [Lauritzen 1996]) this result seems to have been discovered in various forms by several authors. In the possibilistic setting similar theorems hold, although a certain restriction has to be introduced, namely that the graph must have hypertree structure [Gebhardt 1997, Borgelt 2000, Borgelt and Kruse 2002]. This restriction is harmless, though, because the preprocessing of the graph for the well-known join tree evidence propagation method involves the transformation into a graph with hypertree structure anyway.

Finally, the graph underlying a graphical model is very useful to derive evidence propagation algorithms, since evidence propagation can be reduced to simple computations of node processors that communicate by passing messages along the edges of a properly adapted graph. A detailed account, which is beyond the scope of this paper, can be found, for instance, in [Castillo *et al.* 1997].

2.4 A Simple Example

As an illustration of the decomposition of possibility distributions and reasoning in such decompositions we consider a very simple example. Figure 1 shows a three-dimensional possibility distribution on the joint domain of the attributes A, B, and C and its marginal distributions (maxima over rows/columns). This possibility distribution can be decomposed into the marginal distributions on the subspace scaffolded by the attributes A and B and the subspace scaffolded by the attributes B and C, because it can be reconstructed

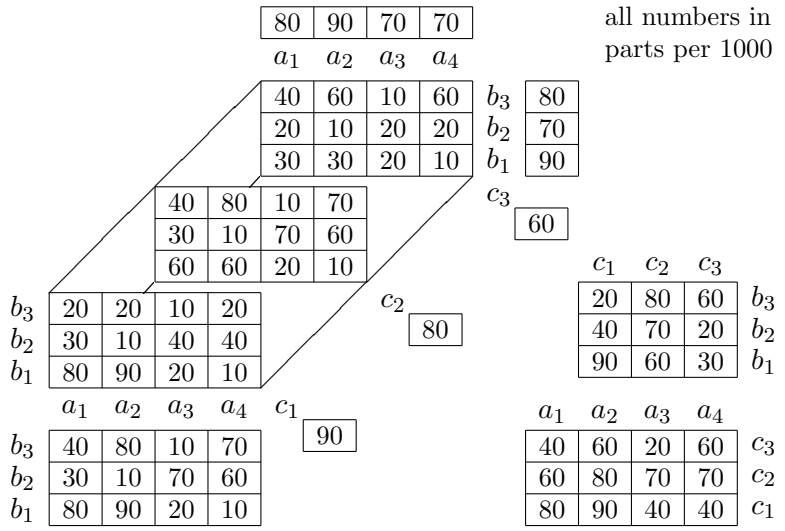


Figure 1: A three-dimensional possibility distribution with marginal distributions (maxima over rows/columns).

using the formula

$$\begin{aligned}
&\forall a \in \mathrm{dom}(A) : \forall b \in \mathrm{dom}(B) : \forall c \in \mathrm{dom}(C) : \\
&\quad \pi_{ABC}(A = a, B = b, C = c) \;=\; \min_{b \in \mathrm{dom}(B)} \{\pi_{AB}(A = a, B = b), \pi_{BC}(B = b, C = c)\} \\
&\qquad\qquad = \min_{b \in \mathrm{dom}(B)} \{ \max_{c \in \mathrm{dom}(C)} \pi_{ABC}(A = a, B = b, C = c), \\
&\qquad\qquad\qquad\qquad \max_{a \in \mathrm{dom}(A)} \pi_{ABC}(A = a, B = b, C = c).\}
\end{aligned}$$

In order to study reasoning in this example, let us assume that from an observation it is known that attribute A has value a_4. Obviously the corresponding (conditional) possibility distribution can be determined from the three-dimensional distribution by restricting it to the "slice" corresponding to $A = a_4$, i.e., by conditioning it on $A = a_4$, and computing the marginal distributions of that "slice". This is demonstrated in Figure 2. Note that the numbers in the "slices" corresponding to other values of attribute A have been set to zero, because these are known now to be impossible. Note also that the numbers in the "slice" corresponding to $A = a_4$ are unchanged, i.e., no renormalization takes place. This is an important difference to the probabilistic case, in which the probabilities have to be renormalized to sum 1.

However, the distributions on the two-dimensional subspaces pointed out above are also sufficient to draw this inference; see Figure 3. The information that $A = a_4$ is extended to the subspace scaffolded by A and B by computing the minimum of the prior degrees of possibility on this subspace (numbers in the upper half of the cells) and the posterior degrees of possibility of $A = a_i$, $i = 1, 2, 3, 4$. The result is shown in the lower half of the cells. Then the marginal distribution on B is determined by taking the maximum over the rows. In the same way the information of the new possibility distribution on B is propagated to C: The minimum of the prior distribution on the subspace scaffolded by B and C and the posterior distribution on B is computed and projected to attribute C by taking the maximum over the columns. It is easy to check that the results obtained in this

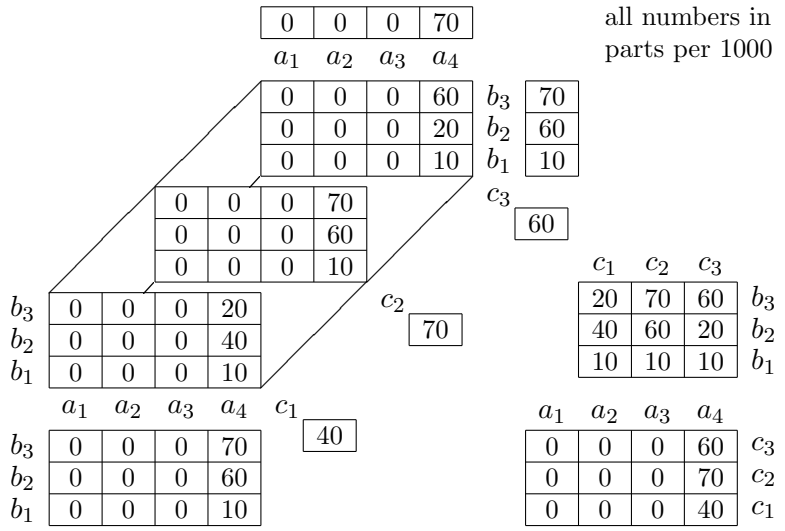


Figure 2: Reasoning in the domain as a whole.

way are the same as those that follow from the computations on the three-dimensional domain (see above).

2.5 Possibility versus Probability

From the simple example of a three-dimensional possibility distribution discussed above it should be clear that probabilistic and possibilistic networks exploit entirely different properties to decompose distributions. This leads, of course, to substantial differences in the interpretation of the reasoning results. To make this clear, we consider how, in the two calculi, the marginal distributions on single attributes relate to the joint distribution they are derived from. This is important, because the reasoning process produces only marginal distributions on single attributes (conditioned on the observations). Since the relation of these marginal distributions to the underlying joint distribution is very different for probability distributions compared to possibility distributions, one has to examine whether it is actually the joint distribution one is interested in.

The difference is, of course, due to the way in which marginal distributions are computed in the two calculi. In probability theory the summation over the dimensions to be removed wipes out any reference to these dimensions. In the resulting marginal distribution no trace of the attributes underlying these dimensions or their values is left: The marginal distribution refers exclusively to the attributes scaffolding the subspace marginalized to. The reason is, of course, that all values of the removed attributes contribute to the result of the marginalization w.r.t. their relative "importance", expressed in their relative probability.

In possibility theory this is different. Because the maximum is taken over the dimensions to be removed, not all values of the attributes underlying these dimensions contribute to the result of the marginalization. Only the values describing the elementary event or events having the highest degree of possibility determine the marginal degree of possibility. Thus not all information about the values of the removed attributes is wiped out. These attributes are implicitly fixed to those values describing the elementary event or events

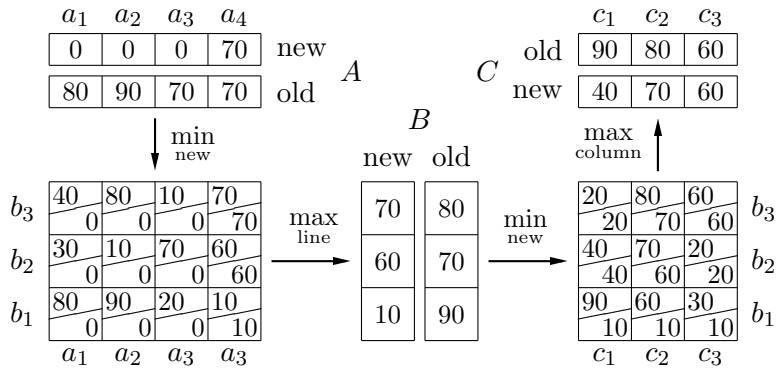


Figure 3: Propagation of the evidence that attribute A has value a_4 in the three-dimensional possibility distribution shown in Figure 1 using the projections to the subspaces $\{A, B\}$ and $\{B, C\}$.

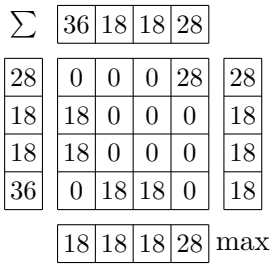


Figure 4: Possibility versus probability w.r.t. the interpretation of marginal distributions (all numbers are percent).

having the highest degree of possibility. It follows that—unlike marginal probabilities, which refer only to tuples over the attributes of the subspace marginalized to—marginal degrees of possibility always refer to *value vectors over all attributes of the universe of discourse*, although only the values of the attributes of the subspace are stated explicitly in the marginal distribution.

In other words, a marginal probability distribution states: "The probability that attribute A has value a is p." This probability is aggregated over all values of all other attributes and thus refers to a one element vector (a). A marginal possibility distribution states instead: "The degree of possibility of a value vector with the highest degree of possibility of all tuples in which attribute A has value a is p." That is, it refers to a value vector over all attributes of the universe of discourse, although the values of all attributes other than A are left implicit.

As a consequence of the difference just studied one has to ask oneself whether one is interested in tuples instead of the value of only a single attribute. An extreme example to illustrate this is shown in Figure 4, which, in the center square, shows a probability distribution over the joint domain of two attributes having four values each. The marginal probability distributions are shown to the left and above this square. Here selecting the tuple containing the values with the highest marginal probabilities decides on an impossible tuple. It follows that in the probabilistic case we may decide incorrectly if we rely exclusively on the marginal distributions (and, indeed, this is not a rare situation). To make the correct decision, we have to compute the joint distribution first or must apply other specialized techniques [Pearl 1988].

0	40	0	40
40	0	0	40
0	0	20	20
40	40	20	max

Figure 5: Maximum projections may lead to an incorrect decision due to an "exclusive-or" effect.

For possibility distributions, however, the situation is different. If in each marginal distribution on a single attribute there is only one value having the highest degree of possibility, then the tuple containing these values is the one having the highest degree of possibility. This is illustrated in Figure 4, where marginal possibility distributions computed by taking the maximum are shown below and to the right of the square (recall that, according to the context model, a probability distribution is only a special possibility distribution and thus we may use maximum projection for probability distributions, too). These marginal distributions indicate the correct tuple.

It should be noted, though, that in the possibilistic setting we may also choose incorrectly, due to a kind of "exclusive-or" effect. This is illustrated in Figure 5. If we decide on the first value for both attributes (since for attributes we have to choose between the first and the second value), we decide on an impossible tuple. Therefore, in this case, we are also forced to compute the joint distribution to ensure a correct decision.

Furthermore, if we are not interested in a tuple over *all* unobserved attributes that has the highest degree of possibility, the special properties of marginal possibility distributions can turn out to be disadvantageous. The reason is that—as indicated above—we cannot get rid of the implicitly fixed values of the attributes that were projected out. If we want to neglect an attribute entirely, we have to modify the universe of discourse and compute the possibility distribution and its decomposition on this modified universe.

3 Computing Projections

In this section we discuss an operation to derive marginal (or conditional) possibility distributions from a dataset of sample cases, which is a prerequisite for learning possibilistic graphical models. We call this operation a *projection*, because in the relational case it corresponds to the projection operation of relational algebra (see above). In the possibilistic case we have to compute a *maximum projection* of the possibility distribution that is induced by the given dataset. The problem is that there is no simple operation to compute such a maximum projection directly from the database to learn from, as we demonstrate with a simple example. Fortunately, however, the database to learn from can be preprocessed by computing its *closure under tuple intersection*, so that it becomes possible to derive any maximum projection with a simple and efficient operation [Borgelt and Kruse 1998, Borgelt 2000, Borgelt and Kruse 2002].

3.1 Databases of Sample Cases

Before we can state clearly the problems underlying the computation of maximum projections from a database of sample cases, it is helpful to define formally what we understand by a database, especially a database with imprecise tuples.

Up to now we considered only precise tuples, which we represented as logical conjunctions $\bigwedge_{A_i \in U} A_i = a_i$ stating a value for each of the attributes. In order to represent imprecise tuples we have to extend this representation, so that a set of possible values can be stated for an attribute. That is, we represent an general tuple as a logical conjunction $\bigwedge_{A_i \in U} A_i \in Q_i$, where $Q_i \subseteq \mathrm{dom}(A_i)$ and $Q_i \neq \emptyset$. For simplicity, we often write tuples similar to the usual vector notation. For example, a tuple t over $\{A, B, C\}$ in which $A \in \{a_1\}$, $B \in \{b_2, b_4\}$ and $C \in \{c_1, c_3\}$ is written $t_{ABC} = (\{a_1\}, \{b_2, b_4\}, \{c_1, c_3\})$.

With this notation a tuple can represent *imprecise* (i.e. set-valued) information about the state of the modeled section of the world. It is, however, restricted in doing so. It cannot represent *arbitrary* sets of instantiations of the attributes, but only such sets that can be defined by stating a *set of values* for each attribute. We chose not to use a more general definition (which would define a general tuple as a disjunction of normal tuples, i.e., of conjunctions stating one value for each of the attributes), because the above definition is usually much more convenient for practical purposes. It should be noted, though, that all definitions and especially the theorem we are going to prove can be transferred directly to the more general case, because the restriction of the above notation is not exploited.

We can now formally define the notions of a *precise* and an *imprecise* tuple: A tuple $t_U = \bigwedge_{A_i \in U} A_i \in Q_i$ over a set U of attributes is called *precise* iff $\forall A_i \in U : |Q_i| = 1$. Otherwise it is called *imprecise*. That is, in a precise tuple there is exactly one value for each of the attributes, while in an imprecise tuple there is at least one attribute for which more than one value is possible. The set of all tuples over X is denoted T_U, the set of all precise tuples over X is denoted $T_U^{(\mathrm{precise})}$. As usual we collect several tuples in a *relation*, which we define here in the usual way as a simple set of tuples (in contrast to the indicator function definition we used above). Or formally, $R_U \subseteq T_U$, i.e., a relation is a subset of the set of all tuples.

For both individual tuples and relations we need the notion of a *projection*, which we transfer directly from relational algebra: If t_X is a tuple over a set X of attributes and $Y \subseteq X$, then $t_X|_Y$ denotes the *restriction* or *projection* of the tuple t_X to Y. That is, the tuple $t_X|_Y$ contains only those terms of the logical conjunction which is the tuple t_X that refer to attributes in Y. Consequently, $t_X|_Y$ is a tuple over Y. If R_X is a relation over a set X of attributes and $Y \subseteq X$, then the *projection* $\mathrm{proj}_Y^X(R_X)$ of the relation R_X from X to Y is defined as

$$\mathrm{proj}_Y^X(R_X) = \{t_Y \in T_Y \mid \exists t_X \in R_X : t_Y \equiv t_X|_Y\}.$$

It is clear that a simple relation does not suffice to describe a dataset of sample cases. In a relation, as it is a *set* of tuples, each tuple can appear only once. In contrast to this, in a dataset of sample cases a tuple may appear several times, reflecting the frequency of the occurrence of the corresponding case. Since we cannot dispense with this frequency information, we need a mechanism to represent the number of occurrences of a tuple. As a consequence we define a *database* D_U over a set U of attributes as a pair (R_U, w_{R_U}), where R_U is a relation over U and w_{R_U} is a function mapping each tuple in R_U to a natural number, i.e. $w_{R_U} : R_U \to \mathbb{N}$. If the set U of attributes is clear from the context, we drop the index U. The function w_{R_U} is intended to indicate the number of occurrences of a tuple $t \in R_U$ in a dataset of sample cases. We call $w_{R_U}(t)$ the *weight* of the tuple t.

When dealing with imprecise tuples, it is helpful to be able to speak of a precise tuple being "contained" in an imprecise one or of one imprecise tuple being "contained" in another (w.r.t. the set of represented instantiations of the attributes). These terms are

made formally precise by introducing the notion of a tuple being *at least as specific* as another: A tuple $t_1 = \bigwedge_{A_i \in X} A_i \in Q_i^{(1)}$ over an attribute set X is called *at least as specific as* a tuple $t_2 = \bigwedge_{A_i \in X} A_i \in Q_i^{(2)}$ over X, written $t_1 \sqsubseteq t_2$ iff $\forall A_i \in X : Q_i^{(1)} \subseteq Q_i^{(2)}$.

Note that $\sqsubseteq$ is not a total ordering, since there are tuples that are incomparable. For example, $t_1 = (\{a_1\}, \{b_1, b_2\})$ and $t_2 = (\{a_1, a_2\}, \{b_1, b_3\})$ are incomparable, since neither $t_1 \sqsubseteq t_2$ nor $t_2 \sqsubseteq t_1$ holds. Note also that $\sqsubseteq$ is obviously transitive, i.e., if t_1, t_2, t_3 are three tuples over an attribute set X with $t_1 \sqsubseteq t_2$ and $t_2 \sqsubseteq t_3$, then also $t_1 \sqsubseteq t_3$. Finally, note that $\sqsubseteq$ is preserved by projection. That is, if t_1 and t_2 are two tuples over an attribute set X with $t_1 \sqsubseteq t_2$ and if $Y \subseteq X$, then $t_1|_Y \sqsubseteq t_2|_Y$.

3.2 Maximum Projections

If we rely on the context model interpretation of a degree of possibility (cf. Section 2.2), a given database is interpreted as a description of a random set. Each tuple is identified with a context and thus the relative tuple weight is the context weight. The sample space Ω is assumed to be the set $T_U^{(\text{precise})}$ of all precise tuples over the set U of attributes of the database. With these presuppositions the possibility distribution $\pi_U^{(D)}$ that is induced by a database D over a set U of attributes can be defined as follows: Let $D = (R, w_R)$ be a non-empty database (i.e. $R \neq \emptyset$) over a set U of attributes. Then

$$\pi_U^{(D)} : T_U^{(\text{precise})} \to [0, 1], \qquad \pi_U^{(D)}(t) \mapsto \frac{\sum_{s \in R, t \sqsubseteq s} w_R(s)}{\sum_{s \in R} w_R(s)},$$

is the *possibility distribution over U induced by D*. That is, the degree of possibility of each precise tuple t is the relative weight of those (imprecise) tuples that contain it.

It is obvious that for a precise database (i.e., a database containing only precise tuples) computing maximum projections is very simple: We traverse the tuples of the subspace X we want to project to. For each tuple t of this subspace we determine the maximum of the weights of those tuples in the database for which the projection to X is the tuple t. That this simple procedure is possible can easily be seen from the fact that for a precise database the numerator of the fraction in the definition of a database-induced possibility distribution is reduced to one term. Therefore we have

$$\forall t_X \in T_X : \quad \pi_X^{(D)}(t_X) = \max_{A \in U - X} \pi_U^{(D)}(t_U) = \frac{\max_{A \in U - X} w_R(t_U)}{\sum_{s \in R} w_R(s)}.$$

Unfortunately this simple procedure cannot be transferred to databases with imprecise tuples, because in the presence of imprecise tuples the sum in the numerator has to be taken into account. This sum poses problems, because the computation of its terms can be very expensive.

3.3 A Simple Example

To understand the problems that result from databases of imprecise tuples it is helpful to study a simple example that clearly shows the difficulties that arise. Consider the very simple database shown in Table 1 that is defined over two attributes A and B. The possibility distribution on the joint domain of A and B that is induced by this database is shown graphically in Figure 6. This figure also shows the marginal possibility

Database:	$(\{a_1, a_2, a_3\}, \{b_3\})$	: 1
	$(\{a_1, a_2\}, \{b_2, b_3\})$	: 1
	$(\{a_3, a_4\}, \{b_1\})$	: 1

Table 1: A very simple imprecise database with three tuples (contexts), each having a weight of 1.

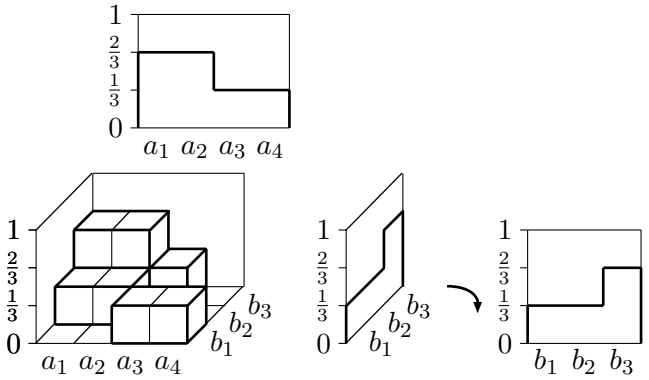


Figure 6: The possibility distribution induced by the three tuples of the database shown in Table 1.

distributions (maximum projections) for each of the two attributes. Consider first the degree of possibility that attribute A has the value a_3, which is $\frac{1}{3}$. This degree of possibility can be computed by taking the *maximum* over all tuples in the database in which the value a_3 is possible: Both tuples in which it is possible have a weight of 1. On the other hand, consider the degree of possibility that attribute A has the value a_2, which is $\frac{2}{3}$. To get this value, we have to *sum* the weights of the tuples in which it is possible. Since both a_2 and a_3 are possible in two tuples of the database, we conclude that neither the sum nor the maximum of the tuple weights can, in general, yield the correct result.

It should be noted that this problem of computing maximum projections results from the fact that we consider unrestricted random sets, i.e., random sets that may have arbitrary focal sets. If we confined ourselves to *consonant* random sets, i.e., random sets in which the focal sets can be ordered into an inclusion sequence $\Gamma(c_1) \subseteq \Gamma(c_2) \subseteq \ldots \subseteq \Gamma(c_k)$, summing over the tuple weights would always yield the correct result, because disjoint tuples (like the first and the third in the database), for which taking the maximum is necessary, are excluded. However, it is also clear that consonance of the focal sets is almost never to be had if random sets are used to interpret databases of sample cases.

Similarly, note that we could compute a maximum projection easily if the focal sets where pairwise disjoint, i.e., if $\forall i, j : \Gamma(c_i) \cap \Gamma(c_j) = \emptyset$. In this case taking the maximum over the tuple weights would always yield the correct result. However, this requirement is no better than the requirement that the focal sets must be consonant: It cannot be expected to be satisfied in applications. Therefore we have to face the challenge of finding an operation that allows us to compute a maximum projection even in a situation in which there are tuples that represent disjoint sets of instantiations as well as tuples that have instantiation of the attributes in common.

Database	
$(\{a_1, a_2, a_3\}, \{b_3\})$	: 1
$(\{a_1, a_2\}, \{b_2, b_3\})$	: 1
$(\{a_3, a_4\}, \{b_1\})$	: 1
3 tuples	

Support	
$(a_1, b_2) : 1$	$(a_3, b_1) : 1$
$(a_1, b_3) : 2$	$(a_3, b_3) : 1$
$(a_2, b_2) : 1$	$(a_4, b_1) : 1$
$(a_2, b_3) : 2$	
7 tuples	

Closure	
$(\{a_1, a_2, a_3\}, \{b_3\})$	: 1
$(\{a_1, a_2\}, \{b_2, b_3\})$	: 1
$(\{a_3, a_4\}, \{b_1\})$	: 1
$(\{a_1, a_2\}, \{b_3\})$	: 2
4 tuples	

Table 2: The maximum over tuples in the support equals the maximum over tuples in the closure.

Fortunately, the simple example shown in Figure 6 not only illustrates the problem that occurs w.r.t. computing maximum projections of database-induced possibility distribution, but also provides us with a hint how this problem may be solved. Obviously, the problem results from the fact that the first two tuples "overlap" or "intersect" on the precise tuples (a_1, b_3) and (a_2, b_3). If this intersection were explicitly represented—with a tuple weight of 2—we could always determine the correct projection by taking the maximum.

This is demonstrated in Table 2. The table on the left restates the database of Table 1. The table in the middle lists what we call the *support* of the database, which is itself a database. This database consists of all precise tuples that are contained in a tuple of the original database. The weights assigned to these tuples are the values of the numerator of the fraction in the definition of the database-induced possibility distribution. Obviously, the marginal degrees of possibility of a value of any of the two attributes A and B can be determined from this relation by computing the maximum over all tuples that contain this value (divided, of course, by the sum of the weights of all tuples in the original database), simply because this computation is a direct implementation of the definition. Therefore we can always fall back on this method of computing a maximum projection.

Note, however, that this method corresponds to a formal expansion of the database, where we expand each imprecise tuple into the set of precise tuples it represents. Unfortunately, this renders this method computationally infeasible in most cases, especially, if there are many attributes and several imprecise tuples. This problem is already indicated by the fact that even for this very simple example we need seven tuples in the support database, although the original database contains only three.

Consequently a better method than the computation via the support is needed. Such a method is suggested by the third column of Table 2. The first three tuples in this column are the tuples of the original database. In addition, this column contains an imprecise tuple that corresponds to the "intersection" of the first two tuples. Since this tuple is at least as specific as both the first and the second, it is assigned a weight of 2, the sum of the weights of the first and the second tuple. By adding this tuple to the database, the set of tuples becomes *closed under tuple intersection*, which explains the label *closure* of this column. That is, for any two tuples s and t in this database, if we construct the (imprecise) tuple that represents the set of precise tuples that are represented by both s and t, then this tuple is also contained in the database. It is easily verified that, in this example, the marginal degrees of possibility of a value of any of the two attributes A and B can be determined from this database by computing the maximum over all tuples that contain this value.

Hence, if we can establish this equality in general, preprocessing the database so that

it is closed under tuple intersection provides an alternative to a computation of maximum projections via the support database. This is especially desirable, since it can be expected that in general only few tuples have to be added in order to achieve closure under tuple intersection. In the example, for instance, only one tuple needs to be added. Experiments we conducted with some real-world datasets show that this expectation is justified. For example, for the Danish Jersey cattle dataset [Rasmussen 1992], which we used for the experiments reported below, only eight tuples have to be added.

3.4 Computation via the Support

We now introduce the technical notions needed to prove, in a final theorem, that a computation of a maximum projection via the closure under tuple intersection is always equal to a computation via the support of a possibility distribution (which, by definition, yields the correct value—see above). We start by making formally precise the notions of the support of a relation and the support of a database: Let R be a relation over a set U of attributes. The *support of* R, written $\mathrm{support}(R)$, is the set of all precise tuples that are at least as specific as a tuple in R, i.e.

$$\mathrm{support}(R) = \left\{ t \in T_U^{(\mathrm{precise})} \;\middle|\; \exists r \in R : t \sqsubseteq r \right\}.$$

Obviously, $\mathrm{support}(R)$ is also a relation over U.

Using this definition we can define the support of a database: Let $D = (R, w_R)$ be a database over a set U of attributes. The *support of* D is the pair $\mathrm{support}(D) = (\mathrm{support}(R), w_{\mathrm{support}(R)})$, where $\mathrm{support}(R)$ is the support of the relation R and

$$w_{\mathrm{support}(R)} : \mathrm{support}(R) \to \mathbb{N}, \qquad w_{\mathrm{support}(R)}(t) \mapsto \sum_{s \in R, t \sqsubseteq s} w_R(s).$$

Obviously, $\mathrm{support}(D)$ is also a database over U. Comparing this definition to the definition of a database-induced possibility distribution, we see that

$$\pi_U^{(D)}(t) = \begin{cases} \frac{1}{w_0} w_{\mathrm{support}(R)}(t), & \text{if } t \in \mathrm{support}(R), \\ 0, & \text{otherwise,} \end{cases}$$

where $w_0 = \sum_{s \in R} w_R(s)$. It follows that any maximum projection of a database-induced possibility distribution $\pi_U^{(D)}$ over a set U of attributes to a set $X \subseteq U$ can be computed from $w_{\mathrm{support}(R)}$ as follows (although the two projections are identical, we write $\pi_X^{(\mathrm{support}(D))}$ instead of $\pi_X^{(D)}$ to indicate that the projection is computed via the support of D):

$$\pi_X^{(\mathrm{support}(D))} : T_X^{(\mathrm{precise})} \to [0,1],$$
$$\pi_X^{(\mathrm{support}(D))}(t) \mapsto \begin{cases} \frac{1}{w_0} \max\limits_{s \in S(t)} w_{\mathrm{support}(R)}(s), & \text{if } S(t) \neq \emptyset, \\ 0, & \text{otherwise,} \end{cases}$$

where $S(t) = \{s \in \mathrm{support}(R) \mid t \sqsubseteq s|_X\}$ and $w_0 = \sum_{s \in R} w_R(s)$.

It should be noted that, as already mentioned above, the computation of maximum projections via the support of a database is, in general, very inefficient, because of the generally huge number of tuples in $\mathrm{support}(R)$. For instance, for the Danish Jersey cattle example, which we use for our experiments below, there are 712957 tuples in the support of the database, which contains 283 tuples with a total weight of 500.

3.5 Computation via the Closure

In this section we turn to the computation of a maximum projection via the closure of a database under tuple intersection. Clearly, we must begin by defining the notion of the *intersection* of two tuples: A tuple s over a set U of attributes is called the *intersection* of two tuples $t_1 = \bigwedge_{A_i \in U} A_i \in Q_i^{(1)}$ and $t_2 = \bigwedge_{A_i \in U} A_i \in Q_i^{(2)}$ over U, written $s = t_1 \sqcap t_2$ iff $\forall A \in U : s(A) = Q_i^{(1)} \cap Q_j^{(2)}$.

Note that the intersection of two given tuples need not exist. For example, the two tuples $t_1 = (\{a_1\}, \{b_1, b_2\})$ and $t_2 = (\{a_2\}, \{b_1, b_3\})$ do not have an intersection, since the intersection of the sets of values for attribute A is empty. Note also that the intersection s of two tuples t_1 and t_2 is at least as specific as both of them, i.e., it is $s \sqsubseteq t_1$ and $s \sqsubseteq t_2$. In addition, s is the least specific of all tuples s' for which $s' \sqsubseteq t_1$ and $s' \sqsubseteq t_2$, i.e.

$$\forall s' \in T_U : \quad (s' \sqsubseteq t_1 \wedge s' \sqsubseteq t_2) \;\Rightarrow\; (s' \sqsubseteq s \equiv t_1 \sqcap t_2).$$

This is important, since it also says that any tuple that is at least as specific as each of two given tuples is at least as specific as their intersection. (This property is needed in the proof of our closure theorem.) Furthermore, note that intersection is idempotent, i.e. $t \sqcap t \equiv t$. (This is needed below, where we define the notion of a closure of a relation.) Finally, note that the above definition can easily be extended to the more general definition of an imprecise tuple, in which it is defined as an arbitrary set of instantiations of the attributes. Clearly, in this case tuple intersection reduces to simple set intersection.

From the intersection of two tuples we can proceed directly to the notions of *closed under tuple intersection* and *closure of a relation*: Let R be a relation over a set U of attributes. R is called *closed under tuple intersection* iff

$$\forall t_1, t_2 \in R : \quad (\exists s \in T_U : s \equiv t_1 \sqcap t_2) \;\Rightarrow\; s \in R,$$

i.e., iff for any two tuples in R their intersection is also contained in R (provided it exists). The *closure of* R, written closure(R), is the set

$$\text{closure}(R) = \Big\{ t \in T_U \;\Big|\; \exists S \subseteq R : t \equiv \bigsqcap_{s \in S} s \Big\},$$

i.e. the relation R together with all possible intersections of tuples from R. Note that closure(R) is, obviously, also a relation and that it is closed under tuple intersection: If $t_1, t_2 \in \text{closure}(R)$, then, due to the construction,

$$\exists S_1 \subseteq R : \quad t_1 = \bigsqcap_{s \in S_1} s \qquad \text{and} \qquad \exists S_2 \subseteq R : \quad t_2 = \bigsqcap_{s \in S_2} s.$$

If now $\exists t \in T_U : t = t_1 \sqcap t_2$, then

$$t = t_1 \sqcap t_2 = \bigsqcap_{s \in S_1} s \sqcap \bigsqcap_{s \in S_2} s = \bigsqcap_{s \in S_1 \cup S_2} s \in \text{closure}(R).$$

(The last equality in this sequence holds, because $\sqcap$ is idempotent, see above.)

Note also that a direct implementation of the above definition is not the best way to compute closure(R). A better, because much more efficient way is to start with a relation $R' = R$, to compute only intersections of pairs of tuples taken from R', and to add the results to R' until no new tuples can be added. The final relation R' is the closure of R.

As for the support, the notion of a closure is extended to databases: Let $D = (R, w_R)$ be a database over a set U of attributes. The *closure of* D is the pair $\text{closure}(D) = (\text{closure}(R), w_{\text{closure}(R)})$, where $\text{closure}(R)$ is the closure of the relation R and

$$w_{\text{closure}(R)} : \text{closure}(R) \to \mathbb{N}, \qquad w_{\text{closure}(R)}(t) \mapsto \sum_{s \in R, t \sqsubseteq s} w_R(s).$$

We assert (and prove in the theorem below) that any maximum projection of $\pi_U^{(D)}$ to a set $X \subseteq U$ can be computed from $w_{\text{closure}(R)}$ as follows (we write $\pi_X^{(\text{closure}(D))}$ to indicate that the projection is computed via the closure of D):

$$\pi_X^{(\text{closure}(D))} : T_X^{(\text{precise})} \to [0,1],$$
$$\pi_X^{(\text{closure}(D))}(t) \mapsto \begin{cases} \frac{1}{w_0} \max\limits_{c \in C(t)} w_{\text{closure}(R)}(c), & \text{if } C(t) \neq \emptyset, \\ 0, & \text{otherwise,} \end{cases}$$

where $C(t) = \{c \in \text{closure}(R) \mid t \sqsubseteq c|_X\}$ and $w_0 = \sum_{s \in R} w_R(s)$. Since, as already mentioned, $\text{closure}(R)$ usually contains much fewer tuples than $\text{support}(R)$, a computation based on the above formula is much more efficient. We verify our assertion that any maximum projection can be computed in this way by the following theorem [Borgelt and Kruse 1998, Borgelt 2000, Borgelt and Kruse 2002]:

Closure Theorem: Let $D = (R, w_R)$ be a database over a set U of attributes and let $X \subseteq U$. Furthermore, let $\text{support}(D) = (\text{support}(R), w_{\text{support}(R)})$ and $\text{closure}(D) = (\text{closure}(R), w_{\text{closure}(R)})$ as well as $\pi_X^{(\text{support}(D))}$ and $\pi_X^{(\text{closure}(D))}$ be defined as above. Then

$$\forall t \in T_X^{(\text{precise})} : \qquad \pi_X^{(\text{closure}(D))}(t) = \pi_X^{(\text{support}(D))}(t),$$

i.e., computing the maximum projection of the possibility distribution $\pi_U^{(D)}$ induced by D to the attributes in X via the closure of D is equivalent to computing it via the support of D.

The proof of this theorem can be found in the appendix. As a consequence of this theorem preprocessing the database by computing the closure under tuple intersection makes computing maximum projections very simple and fast.

4 Learning Graphical Models

In this section we turn to algorithms for learning graphical models from a dataset of sample cases. There are three basic approaches to this problem:

Test whether a distribution is decomposable w.r.t. a given graph.
This is the most direct approach. It does not depend on a graphical representation, but can also be applied in connection with other ways of representing the subsets of attributes to be used to compute the (candidate) decomposition of the given distribution.

Find a conditional independence graph by conditional independence tests.
This approach exploits the theorems connecting conditional independence graphs and graphs that describe decompositions, which were mentioned in Section 2. It has the advantage that by a single conditional independence test, if it fails, several candidate graphs can be excluded.

Find a suitable graph by measuring the strength of marginal dependences.
This is a heuristic, but often highly successful approach, which is based on the frequently valid assumption that in a conditional independence graph an attribute is more strongly dependent on adjacent attributes than on attributes that are not directly connected to it.

Note that none of these methods is perfect. The first approach suffers from the usually huge number of candidate graphs. The second often needs the strong assumption that there is a perfect map (a conditional independence graph that captures *all* conditional independences by node separation) w.r.t. the considered distribution. In addition, if it is not restricted to certain types of graphs (for example, polytrees), one has to test conditional independences of high order, i.e. with a large number of conditioning attributes, which tend to be unreliable unless the amount of data is enormous. The heuristic character of the third approach is obvious. Examples in which it fails can easily be found, since under certain conditions attributes that are not adjacent in a conditional independence graph can exhibit a strong dependence [Borgelt 2000, Borgelt and Kruse 2002].

A (computationally feasible) analytical method to construct an optimal graphical model from a dataset of sample cases has not been found yet. Therefore an algorithm for learning a graphical model from data usually consists of

1. an *evaluation measure* (to assess the quality of a given network) and
2. a *search method* (to traverse the space of possible networks).

It should be noted, though, that restrictions of the search space introduced by an algorithm and special properties of the evaluation measure used sometimes disguise the fact that a search through the space of possible network structures is carried out. For example, by conditional independence tests all graphs missing certain edges can be excluded without inspecting these graphs explicitly. Greedy approaches try to find good edges or subnetworks and combine them in order to construct an overall model and thus may not appear to be searching. Nevertheless the above characterization that a learning algorithm for graphical models consists of an evaluation measure and a search method is apt, since an algorithm that does not explicitly search the space of possible networks usually carries out a (heuristic) search on a different level, guided by an evaluation measure. For example, some greedy approaches search for the best set of parents of an attribute by measuring the strength of dependence on candidate parent attributes; conditional independence test approaches search the space of all possible conditional independence statements also measuring the strengths of (conditional) dependences.

4.1 Evaluation Measures

An *evaluation measure* is used to assess the quality of a given candidate graphical model w.r.t. a given dataset of sample cases, so that it can be decided which of a set of candidate graphical models best fits the given data. A desirable property of an evaluation measure is decomposability, i.e., the total network quality should be computable as an aggregate (e.g. sum or product) of local scores, for example scores for the maximal clique of the graph to be assessed or scores for single edges. Here we consider decomposable (or local) evaluation measures as well as a global one.

Most local evaluation measures are based on measures of dependence, since for both the second and the third basic approach listed above it is necessary to measure the strength of dependence of two or more attributes, either in order to test for conditional independence

or in order to find the strongest dependences. In the following we define all local evaluation measures w.r.t. two attributes. The extension to conditional tests is straightforward: The strength of dependence of the two attributes is computed for each instantiation of the conditions. The results are then summed or averaged, to obtain a measure for the strength of the conditional dependence. We may then decide that the two attributes are conditionally independent if the the value of this measure does not exceed a certain threshold.

A generalization to more than two attributes is also easy to achieve: If a directed graphical model is to be evaluated, all conditioning attributes may be combined into one pseudo-attribute. That is, we measure the strength of dependence between the child attribute and an artificial attribute that represents the combination of all parent attributes. This artificial attribute has the Cartesian product of the domains of the parent attributes as its domain. If the measure exhibits a certain symmetry, it is also possible to find direct generalizations to more than two attributes, examples of which are discussed below. Such a generalization is often preferable if undirected graphical models are to be learned.

4.1.1 Specificity Gain

One way to derive evaluation measures for possibilistic networks is to exploit their close connection to relational networks (see above). The idea is to draw on the *α-cut view* of a possibility distribution, a concept that is also well known in fuzzy set theory [Kruse *et al.* 1994]. In this view a possibility distribution is seen as a *set of relations* with one relation for each degree of possibility α. The indicator functions $[\pi]_\alpha$ of these relations are defined by simply assigning a value of 1 to all tuples for which the degree of possibility is at least α and a value of 0 to all other tuples. It is easy to see that a possibility distribution is decomposable if and only if each of the α-cut relations is decomposable. Formally, this corresponds the obvious equivalence of the equations

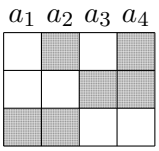
$$\begin{aligned}&\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) : \\ &\quad \pi_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big) = \min_{M \in \mathcal{M}} \pi_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big),\end{aligned}$$

i.e. the decomposition equation for a possibility distribution, and

$$\begin{aligned}&\forall a_1 \in \mathrm{dom}(A_1) : \ldots \forall a_n \in \mathrm{dom}(A_n) : \forall \alpha \in [0,1] : \\ &\quad \left[\pi_U\Big(\bigwedge_{A_i \in U} A_i = a_i\Big)\right]_\alpha = \min_{M \in \mathcal{M}} \left[\pi_M\Big(\bigwedge_{A_i \in M} A_i = a_i\Big)\right]_\alpha,\end{aligned}$$

i.e. the set of decomposition equations for the α-cuts. A pleasant consequence of this equivalence is that we may derive a measure for the strength of possibilistic dependence of two variables by integrating a measure for the strength of relational dependence over all degrees of possibility α.

One measure for the strength of relational dependence we may choose is the so-called *Hartley information gain*. This measure is based on the *Hartley entropy* (or Hartley information) [Hartley 1928] of a set of alternatives, which is defined as the binary logarithm of the number of alternatives in the set. Hartley entropy can be seen as a special case of the better known *Shannon entropy* (or Shannon information) [Shannon 1948], which results if all alternatives have the same probability. Consequently, its interpretation is similar to



Hartley information needed to determine		
coords.:	$\log_2 4 + \log_2 3 = \log_2 12$	≈ 3.58
coord. pair:	$\log_2 6$	≈ 2.58
gain:	$\log_2 12 - \log_2 6 = \log_2 2$	$= 1$

Figure 7: Illustration of the computation of Hartley information gain.

the interpretation of Shannon entropy: It is the average number of yes/no-questions that have to be asked in order to single out the obtaining alternative.

Hartley information gain is derived from Hartley entropy in the same way as the well-known *Shannon information gain* [Shannon 1948] is derived from Shannon entropy. For two attributes A and B Shannon information gain is defined as

$$I_{\text{gain}}^{(\text{Shannon})}(A, B) = H(A) + H(B) - H(A, B),$$

where H is the Shannon entropy, i.e., for instance,

$$H(A) = - \sum_{a \in \text{dom}(A)} p_A(a) \log_2 p_A(a).$$

Therefore Hartley information gain is defined as

$$\begin{aligned} I_{\text{gain}}^{(\text{Hartley})}(A, B) &= \log_2 \Big(\sum_{a \in \text{dom}(A)} r_A(a) \Big) \\ &+ \log_2 \Big(\sum_{b \in \text{dom}(B)} r_B(b) \Big) \\ &- \log_2 \Big(\sum_{\substack{a \in \text{dom}(A) \\ b \in \text{dom}(B)}} r_{AB}(a, b) \Big), \end{aligned}$$

where the terms are the Hartley entropies of A, B, and AB, respectively. (Obviously the sums simply count the number of tuples in the relations, because we describe a relation here again by its indicator function).

To illustrate the idea underlying Hartley information gain, we consider the simple two-dimensional relation shown in figure 7: The grey squares indicate the tuples contained in this relation. Suppose that we want to determine the actual values of the two attributes A and B. It is clear that there are two possible ways to do this: In the first place, we could determine the value of each attribute separately, thus trying to find the "coordinates" of the obtaining value combination. Or we may exploit the fact that the possible value combinations are restricted by the relation shown in figure 7 and try to determine the value combination directly. In the former case we need the Hartley information of the set of values of A plus the Hartley information of the set of values of B, i.e., $\log_2 4 + \log_2 3 \approx 3.58$ bits. In the latter case we need the Hartley information of the possible value pairs, i.e. only $\log_2 6 \approx 2.58$ bit, and thus gain one bit. Since it is plausible that we gain the more bits, the more strongly dependent the two attributes are (because in this case fixing a value of one of the attributes leaves fewer choices for the value of the other), we may use the Hartley information gain as a direct indication of the strength of relational dependence of the two attributes.

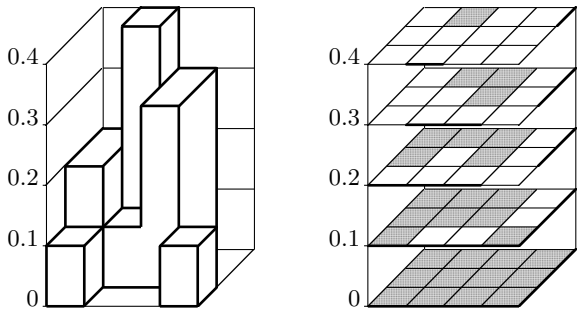


Figure 8: Illustration of the idea of specificity gain.

The Hartley information gain can be generalized to the *specificity gain* [Gebhardt and Kruse 1996b, Borgelt 2000, Borgelt and Kruse 2002] as shown in figure 8: It is simply integrated over all α-cuts of a given possibility distribution, thus exploiting the equivalence of the decomposition formulae pointed out above. Formally, we have

$$\begin{aligned} S_{\text{gain}}(A,B) \;=\; & \int_0^{\sup \pi_{AB}} \log_2 \Big(\sum_{a \in \text{dom}(A)} [\pi_A]_\alpha(a) \Big) \\ & + \log_2 \Big(\sum_{b \in \text{dom}(B)} [\pi_B]_\alpha(b) \Big) \\ & - \log_2 \Big(\sum_{\substack{a \in \text{dom}(A) \\ b \in \text{dom}(B)}} [\pi_{AB}]_\alpha(a,b) \Big) \, \text{d}\alpha. \end{aligned}$$

Another way of deriving this measure is via the notion of the *nonspecificity* of a possibility distribution, which is defined as [Klir and Mariano 1987]

$$\text{nsp}(\pi) = \int_0^{\sup \pi} \log_2 \Big(\sum_{\omega \in \Omega} [\pi]_\alpha(\omega) \Big) \, \text{d}\alpha,$$

where Ω is the domain on which π is defined. Obviously this measure can be seen as a generalization of Hartley entropy to the possibilistic case [Higashi and Klir 1982]. Using nonspecificity in the same way as Hartley entropy and Shannon entropy, we get [Gebhardt and Kruse 1996b, Borgelt *et al.* 1996]:

$$S_{\text{gain}}(A,B) = \text{nsp}(\pi_A) + \text{nsp}(\pi_B) - \text{nsp}(\pi_{AB}).$$

In the probabilistic setting it is well known that Shannon information gain is biased towards many-valued attributes [Kononenko 1995]. Therefore certain normalization of this measure have been introduced, like, for instance, information gain ratio [Quinlan 1993], which is a well-known measure for decision tree induction. Hence, by exploiting the analogy of Shannon information gain and specificity gain, we may define similar normalization for the possibilistic case. This leads to the *specificity gain ratio*

$$S_{\text{gr}}(A,B) \;=\; \frac{S_{\text{gain}}(A,B)}{\text{nsp}(\pi_B)} \;=\; \frac{\text{nsp}(\pi_A) + \text{nsp}(\pi_B) - \text{nsp}(\pi_{AB})}{\text{nsp}(\pi_B)}$$

and two *symmetric specificity gain ratios*, namely

$$S^{(1)}_{\text{sgr}}(A,B) = \frac{S_{\text{gain}}(A,B)}{\text{nsp}(\pi_{AB})} \quad \text{and}$$

$$S^{(2)}_{\text{sgr}}(A,B) = \frac{S_{\text{gain}}(A,B)}{\text{nsp}(\pi_A) + \text{nsp}(\pi_B)}.$$

It should be noted that the specificity gain and the two symmetric gain ratios can easily be generalized to more than two attributes by simply adding terms for additional attributes and extending the term referring to the joint distribution. This is not possible for the (asymmetric) specificity gain ratio due to the special role played by attribute B. Hence this measure can only be used to evaluate directed graphs, where B may be the conditioning attribute.

Another variant of specificity gain can be derived by drawing on a version of the Hartley information gain that is computed from conditional relations (one relation for each instantiation of the parent attribute). The idea of this measure is to compare the Hartley information of the conditional relation (for a given instantiation of the parent attribute) with the Hartley information of the unconditional distribution, because the relative information of the conditional distribution is the lower the more strongly dependent the two attributes are. The comparison results are then summed weighted over all possible instantiation of the parent attribute. The resulting *conditional specificity gain* is defined as

$$S_{\text{cgain}}(A,B) = \sum_{b \in \text{dom}(B)} \int_0^{\pi_B(b)} \frac{[\pi_B]_\alpha(b)}{\sum_{b \in \text{dom}(B)} [\pi_B]_\alpha(b)} \log_2 \frac{\sum_{a \in \text{dom}(A)} [\pi_A]_\alpha(a)}{\sum_{b \in \text{dom}(B)} [\pi_{A|B}]_\alpha(a \mid b)} \, \text{d}\alpha.$$

Obviously, the logarithm describes the relation of the two Hartley information values, as indicated above. These relative values, one for each possible condition, are weighted with the degree of possibility of the conditional distribution. Due to its inherent conditional nature, this measure can only be used for directed graphical models.

4.1.2 Possibilistic Mutual Information

Apart from a transfer from the relational case (as in the preceding section), evaluation measures for possibilistic graphical models may also be derived by forming analogs of well known probabilistic measures. This approach was already implicitly employed in the preceding section, when we derived the specificity gain from nonspecificity by using it in the same way as Shannon entropy is used to derive the Shannon information gain. But Shannon information gain can be written in different ways, one of which was used above. Another way, which is usually called *mutual information* or *cross entropy* (although this is exactly the same measure) is

$$I^{(\text{Shannon})}_{\text{mutual}}(A,B) = \sum_{\substack{a \in \text{dom}(A) \\ b \in \text{dom}(B)}} p_{AB}(a,b) \cdot \log_2 \frac{p_{AB}(a,b)}{p_A(a) \cdot p_B(b)}.$$

A natural interpretation of this measure is that it computes a pointwise comparison of the actual joint distribution p_{AB} with a hypothetical independent distribution $p_A \cdot p_B$. The

pointwise results are weighted with the actual probability p_{AB}. This comparison idea can be transferred by defining

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

as a direct analog of mutual Shannon information [Borgelt and Kruse 1997, Borgelt 2000, Borgelt and Kruse 2002]. (The index "mi" stands for "mutual information".) Note, however, the additional minus sign, which results from the fact that

$$\forall a \in \mathrm{dom}(A) : \forall b \in \mathrm{dom}(B) : \qquad \pi_{AB}(a,b) \leq \min\{\pi_A(a), \pi_B(b)\}.$$

Note also, that this measure differs from the specificity gain, whereas Shannon information gain and mutual information are the same measure, only written in different ways. Finally, note that there is again a straightforward generalization to more than two attributes.

4.1.3 Possibilistic χ^2 measure

Mutual (Shannon) information, as it was studied in the preceding section, is not the only probabilistic measure that is based on the idea to compare the actual joint distribution to a hypothetical independent distribution. Another measure that is directly based on this idea is the well-known χ^2 measure. Instead of a quotient, this measure computes the pointwise squared difference, since it is defined as

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

where N is the number of sample cases in the dataset. To remove the dependence on the number of cases in the dataset this measure is often normalized by dividing it by N.

It is clear that this idea may as well be transferred to the possibilistic case, so that we get [Borgelt and Kruse 1997, Borgelt 2000, Borgelt and Kruse 2002]

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

Alternatively, we may compute the weighted sum of the squared differences of the individual degrees of possibility, i.e., we may compute [Borgelt 2000, Borgelt and Kruse 2002]

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

This measure appears to be slightly more natural than the direct analog of the χ^2 measure. It should also be noted that both of these measures allow again for a straightforward generalization to more than two attributes.

4.1.4 Weighted Sum of Possibility Degrees

All measures studied up to now are local evaluation measures, because they assess the strength of dependence of two attributes and hence the evaluation of a graphical model is composed of several local scores. However, there is also a global evaluation measure, which cannot be decomposed. The idea underlying it is that in order to assess a possibilistic graphical models, we may directly compare the possibility distribution represented by it to the one that is induced by the dataset to learn from. The problem with this approach is, of course, that in order to do so, we would have to compute the degrees of possibility for all points of the multidimensional domain, which is clearly impossible except when there are very few attributes: The size of this joint domain grows exponentially with the number of attributes.

However, we may consider restricting the set of points from which we compute this measure, so that the computation becomes efficient. If we select a proper subset of points of the underlying multidimensional domain, the resulting ranking of different graphical models may coincide with the ranking computed from the full set of points. A natural choice for such a subset is the set of sample cases recorded in the dataset to learn from, because from these the distribution is induced and thus it is most important to approximate their degrees of possibility well. In addition, we may weight the degrees of possibility for these sample cases with their frequency in order to capture their relative importance. That is, we may compute

$$Q(G) = \sum_{t \in D} w(t) \cdot \pi_G(t)$$

to assess the quality of a given graphical model G [Borgelt 2000, Borgelt and Kruse 2002], where D is the dataset to learn from and $w(t)$ is the weight (number of occurrences) of a tuple t. This measure should be minimized by a learning algorithm, since a bad graphical model will, on average, assign higher degrees of possibility than a good one. The reason is that the degrees of possibility of the distribution represented by the graphical model are computed as minima of maximum projections that were derived from the dataset. Therefore these degrees of possibility can only be greater than the degrees of possibility of the database-induced possibility distribution, but never smaller. Only if the graphical model represents the database-induced possibility distribution perfectly, these degrees of possibility are equal. Consequently, we should strive to make the degrees of possibility that are computed from the graphical model as small as possible, in order to approximate the database-induced possibility distribution as closely as possible.

Of course, computing the value of the above measure is simple only if all tuples are precise, because only for a precise tuple a unique degree of possibility can be determined from the graphical model to evaluate. For an imprecise tuple some kind of approximation has to be used. We may, for instance, compute an aggregate, e.g. the average or the maximum, of the degrees of possibility of all precise tuples that are compatible with an imprecise tuple. Since we are trying to minimize the value of the measure, it seems natural to choose pessimistically the maximum as the worst possible case. This choice has the additional advantage that it can be computed efficiently by simply propagating the evidence contained in an imprecise tuple in the given graphical model, whereas other aggregates suffer from the fact that we have to compute explicitly the degree of possibility of every compatible precise tuples. Because the number of these tuples can be very large, such an evaluation can be extremely costly.

Note that the weighted sum of possibility degrees may be penalized—in analogy to the so-called information criteria in the probabilistic case [Lauritzen 1996]—by adding a term that depends on the number of parameters of the model [Borgelt 2000, Borgelt and Kruse 2002]. This penalty term introduces a bias towards simpler models, i.e. simpler graph structures, and thus reduces the danger of overfitting the model to the data.

4.2 Search Methods

A search method determines which graphs are considered in order to find a good graphical model. Since an exhaustive search is impossible due to the huge number of graphs, one has to rely on heuristic search methods. Usually these heuristic methods severely restrict the set of admissible graphs and exploit the value of the chosen evaluation measure to guide the search. In addition they are often greedy w.r.t. the quality of the graphical model.

The simplest instance of such a heuristic search method is, of course, the well-known Kruskal algorithm [Kruskal 1956], which determines an optimum weight spanning tree for given edge weights. This algorithm has been used in the probabilistic setting by [Chow and Liu 1968], who used the mutual Shannon information of the connected attributes as edge weights. In the possibilistic setting, we may simply replace the mutual Shannon information by the specificity gain in order to arrive at an analogous algorithm [Gebhardt and Kruse 1996b, Borgelt 2000, Borgelt and Kruse 2002]. Of course, the other local measures discussed above may also be used, provided they are symmetric w.r.t. the attributes.

A natural extension of the Kruskal algorithm is a greedy parent selection for directed graphs, which is often carried out on a topological order of the attributes that is fixed in advance[2]: At the beginning the value of an evaluation measure is computed for a parentless child attribute. This can be achieved with the measures listed above by simply assuming that the other attribute that enters the computations has only one possible value. Then in turn each of the parent candidates (the attributes preceding the child in the topological order) is temporarily added and the evaluation measure is recomputed. The parent candidate that yields the highest value of the evaluation measure is selected as a first parent and is permanently added. In the third step each remaining parent candidate is added temporarily as a second parent and again the evaluation measure is recomputed. As already pointed out above, the parents may have to be combined into a pseudo-attribute in this case in order to compute the evaluation measure. As before, the parent candidate that yields the highest value is permanently added. The process stops if either no more parent candidates are available, a given maximum number of parents is reached, or none of the parent candidates, if added, results in an improved model quality.

This search method has been used by [Cooper and Herskovits 1992] in the well-known K2 algorithm. As an evaluation measure they used what has become known as the *K2 metric*, which has later been generalized by [Heckerman *et al.* 1995] to the *Bayesian-Dirichlet metric*. Of course, in the possibilistic setting we may also apply this search method, again relying on the specificity gain or any other local possibilistic evaluation measure.

[2] A topological order is an order of the nodes of a directed acyclic graph such that for all nodes their parents precede them in the order. By fixing a topological order, the set of possible graphs is severely restricted and it is ensured that the resulting graph is acyclic.

5 Experimental Results

Evaluating the quality of learning methods is more difficult for possibilistic graphical models than for probabilistic ones. With an algorithm for learning probabilistic graphical models we can always rely on the following simple approach: We select an arbitrary graphical model, either a randomly generated one or any of the several human expert designed example networks that are available on the Internet. From the chosen network we generate a database of appropriate size using simple Monte Carlo simulation, i.e., instantiating the attributes w.r.t. the probability distributions specified by the network. This is especially simple if the chosen network is a Bayesian network, i.e. a directed probabilistic graphical model. Then we try to recover the original network from the database using the learning algorithms we are interested in. The main advantage of this approach is that it provides us with several means to assess the learning algorithm. We may, for instance, compute the probability of a test database, which was not used for learning, w.r.t. the original model and the learned one and then compare the two. We may also compare the edges in the two networks and find out which are missing and which have been added.

Unfortunately, it is not possible to transfer this scheme directly to the possibilistic case: Suppose we choose a possibilistic network, say, a randomly generated one, which we want to recover from a database of sample cases using a learning algorithm. How do we generate the database to learn from in this case? Possibilistic networks do not allow for a Monte Carlo simulation to obtain sample cases as probabilistic networks do. In addition, the sample cases we require in the possibilistic case should contain set-valued information or at least missing values. How do we decide which attributes are specified precisely and which by a set of values in a sample case? The possibilistic graphical model does not provide information about this. Therefore we have to rely on one of the following possibilities: Either we start directly from a set of sample cases, for instance, real world data, or we generate sample cases from a probabilistic network, adding missing values and set valued information either randomly or as specified by a separate "imprecision model". However, both approaches rule out the possibility to compare the learned network to an reference one, in the former approach, because there is no reference network, in the latter, because the structure of a possibilistic network, in general, differs considerably from the structure of a probabilistic network for the same domain due to the different notions of conditional independence employed.

Another problem is the following: While it is possible to evaluate a learned graphical model w.r.t. the training data using, for instance, the global evaluation measure discussed above or, if the distribution is small enough, a direct comparison of the possibility distributions represented by the learned graph and induced by the given data, it is not as easy to evaluate a possibilistic graphical model w.r.t. test data. While a probabilistic network allows us to compute the probability of any database over the same set of attributes, a possibilistic network does not provide us with a similar quality measure. Of course, we can evaluate a test dataset in the same way as the training dataset, i.e., by computing, for example, the global evaluation measure discussed above. However, the result is ambiguous: For the training dataset, a small value surely indicates a good fit to the data. For the test dataset, a small value may as well indicate that the model fits the data well, for the same reasons as a small value indicates that the model fits the training data well. However, it may also indicate that the model fits the data very badly, because the test dataset contains

Baseline

			absolute			relative		
network	edges	parms.	avg.	min.	max.	avg.	min.	max.
indep.	0	80	10.160	10.064	11.390	2.475	2.414	1.572
orig.	22	308	9.917	9.888	11.318	2.232	2.238	1.500

Optimum weight spanning tree

			absolute			relative		
measure	edges	parms.	avg.	min.	max.	avg.	min.	max.
S_{gain}	20	410	8.990	8.878	10.714	1.304	1.228	0.896
S_{sgr1}	20	414	8.916	8.716	10.680	1.231	1.066	0.862
d_{χ^2}	20	444	8.820	8.662	10.334	1.135	1.012	0.516
d_{mi}	20	362	8.596	8.466	10.386	0.911	0.816	0.568

Greedy parent selection

			absolute			relative		
measure	edges	parms.	avg.	min.	max.	avg.	min.	max.
S_{gain}	31	1630	8.621	8.524	10.292	0.936	0.874	0.474
S_{gr}	18	196	9.553	9.390	11.100	1.867	1.740	1.282
S_{sgr1}	28	496	8.057	9.946	10.740	1.372	1.296	0.922
d_{χ^2}	36	1486	8.329	8.154	10.200	0.644	0.504	0.382
d_{mi}	33	774	8.344	8.206	10.416	0.659	0.556	0.598

Table 3: Experimental results on real world data

several tuples to which the graphical model assigns a low degree of possibility, although they are frequent and thus highly possible. Which of the two effects causes the measure to be small cannot be determined and thus the result is inconclusive.

A possible solution of this problem is to compare degrees of possibility as they can be computed from the graphical model with degrees of possibility as they result from the (preprocessed) test database by summing the absolute values of their differences over the sample cases—in analogy to the weighted sum of possibility degrees discussed above. The problem that results from sample cases with set-valued information, which we already mentioned in the previous section, can be solved in the same way as above: We may compute an aggregate—preferably the maximum—of the degrees of possibility over all precise tuples compatible with an imprecise tuple. Although this approach clearly lacks the persuasiveness of the computation of the probability of the test database in the probabilistic setting, it provides some indication of the quality of the learning methods.

Our experiments were conducted with a prototype implementation of the evaluation measures and search methods described above, which we called INES (Induction of NEtwork Structures). We applied this program to the well known Danish Jersey cattle blood type determination example [Rasmussen 1992] in two different ways. In a first set of experiments we applied the program to a real world dataset for this domain that consists of 500 sample cases, a considerable number of which contain missing values. Hence this

Baseline

			train			test		
network	edges	parms.	avg.	min.	max.	avg.	min.	max.
indep.	0	80	1.429	1.198	1.367	1.415	1.192	1.419
orig.	22	308	0.849	0.549	1.064	0.830	0.533	1.135

Optimum weight spanning tree

			train			test		
measure	edges	parms.	avg.	min.	max.	avg.	min.	max.
S_{gain}	20	484	0.775	0.547	0.747	0.753	0.525	0.876
S_{sgr1}	20	328	0.868	0.630	0.891	0.845	0.607	0.992
d_{χ^2}	20	567	0.751	0.502	0.725	0.731	0.482	0.866
d_{mi}	20	577	0.783	0.539	0.732	0.766	0.522	0.871

Greedy parent selection

			train			test		
measure	edges	parms.	avg.	min.	max.	avg.	min.	max.
S_{gain}	38	1808	0.440	0.219	0.517	0.417	0.213	0.706
S_{gr}	22	223	0.911	0.612	1.068	0.895	0.599	1.146
S_{sgr1}	34	1129	0.508	0.284	0.582	0.479	0.267	0.744
d_{χ^2}	38	1896	0.451	0.238	0.516	0.426	0.225	0.712
d_{mi}	38	1974	0.457	0.243	0.505	0.435	0.232	0.703

Table 4: Experimental results on artificial data

dataset is well suited for a possibilistic approach. In a second set of experiments we generated 20 random datasets (with 1000 samples cases each) from a human expert designed Bayesian network for this domain, in which we replaced randomly 15% of all entries by missing values. These datasets were grouped in ten pairs. One dataset of each pair was used for learning, the other for evaluating the learned network with the possibility degree comparison approach described above. The results were then averaged over the ten pairs of datasets.

As a baseline for comparisons we chose a graph without any edges and the human expert designed Bayesian reference network. However, as already pointed out several times, the results obtained with the latter do not provide much insight, because a probabilistic network captures a different kind of (in)dependence, since it is based on a different uncertainty calculus.

In the first set of experiments (with the real world dataset) all possibilistic networks were assessed by computing the weighted sum of the degrees of possibility for the tuples in the dataset, which should be as small as possible (cf. the global evaluation measure discussed in the preceding section). However, since the dataset contains a lot of tuples with missing values, a precise degree of possibility cannot always be computed (see above). To cope with this problem, we computed for a tuple with missing values the minimum, the maximum, and the average degree of possibility of all precise tuples compatible with

it. This is possible here, because the underlying joint domain is still small enough to carry out this costly computation, even though it takes some time. The results are then summed separately for all tuples in the dataset. In addition, we compared the degrees of possibility as they can be computed from the graphical model with the degrees of possibility as they can be computed from the (preprocessed) database relying on the same aggregation idea for tuples with missing values.

The results of these experiments are shown in table 3. Clearly, as already mentioned above, the original Bayesian network is not well suited as a baseline for comparisons. The optimum weight spanning tree construction yields very good results with the possibilistic analogs of the χ^2 measure (d_{χ^2}) and mutual information (d_{mi}). For greedy parent selection specificity gain S_{gain} and d_{χ^2} lead to graphical models that are too complex as can be seen from the high number of parameters. The specificity gain ratio seems to be too reluctant to select parents, and thus leads to a model that is simple, but does not fit the data well. The possibilistic analog of mutual information d_{mi} clearly yields the best results, since it achieves a good fit to the data with a not too complex model.

The results of the experiments on the artificially generated datasets are shown on table 4. The results are very similar to the results on the real world dataset, although the possibilistic analog of mutual information d_{mi} fares worse in this case as it also leads to a fairly complex model. According to these results the symmetric specificity gain ratio is the preferable evaluation measure. In order to check our hypothesis that a comparison to the original Bayesian network is not meaningful, we also determined added and missing edges, only to find out the the networks differed considerably (and therefore we do not report these numbers here).

The INES program, which we made available under the GNU Lesser General Public License, as well as the datasets and shell scripts we used for the experiments can be retrieved free of charge at `http://fuzzy.cs.uni-magdeburg.de/books/gm/software.html`. At this URL you can also find other software we developed in connection with our research on learning graphical models from data.

6 Summary

In this paper we reviewed the theory of possibilistic graphical models and how to learn possibilistic graphical models from a dataset of sample cases. We discussed preprocessing operations for the dataset to learn from, which makes it possible to compute maximum projections of database-induced possibility distributions efficiently. In addition, we studied several evaluation measures, based on different ideas, that may be used to assess the quality of a possibilistic graphical model in a learning algorithm, and compared them in a set of experiments.

Acknowledgments

We are grateful to two anonymous reviewers for their helpful comments, which contributed considerably to improving this paper.

Appendix

Proof of closure theorem [Borgelt and Kruse 1998, Borgelt 2000, Borgelt and Kruse 2002]: The assertion of the theorem is proven in two steps. In the first, it is shown that, for an arbitrary tuple $t \in T_X^{(\text{precise})}$,

$$\pi_X^{(\text{closure}(D))}(t) \geq \pi_X^{(\text{support}(D))}(t),$$

and in the second that

$$\pi_X^{(\text{closure}(D))}(t) \leq \pi_X^{(\text{support}(D))}(t).$$

Both parts together obviously prove the theorem. So let $t \in T_X^{(\text{precise})}$ be an arbitrary precise tuple and let $w_0 = \sum_{u \in R} w_R(u)$. Furthermore, let

$$S = \{s \in \text{support}(R) \mid t \sqsubseteq s|_X\} \quad \text{and} \quad C = \{c \in \text{closure}(R) \mid t \sqsubseteq c|_X\}.$$

(1) $\pi_X^{(\text{closure}(D))}(t) \geq \pi_X^{(\text{support}(D))}(t)$:

We have to distinguish two cases, namely $S = \emptyset$ and $S \neq \emptyset$, the first of which is obviously trivial.

(a) $S = \emptyset$: $\pi_X^{(\text{support}(D))}(t) = 0 \leq \pi_X^{(\text{closure}(D))}(t) \in [0, 1]$.

(b) $S \neq \emptyset$: Due to the definitions of $\pi_X^{(\text{support}(D))}$ and $w_{\text{support}(R)}$

$$\begin{aligned} \pi_X^{(\text{support}(D))}(t) &= \frac{1}{w_0} \max_{s \in S} w_{\text{support}(R)}(s) \\ &= \frac{1}{w_0} \max_{s \in S} \sum_{u \in R, s \sqsubseteq u} w_R(u). \end{aligned}$$

Let $\hat{s} \in S$ be (one of) the tuple(s) $s \in S$ for which $w_{\text{support}(R)}(s)$ is maximal. Let $V = \{v \in R \mid \hat{s} \sqsubseteq v\}$, i.e., let V be the set of tuples from which the weight of $\hat{s}$ is computed. Then

$$\pi_X^{(\text{support}(D))}(t) = \frac{1}{w_0} w_{\text{support}(R)}(\hat{s}) = \frac{1}{w_0} \sum_{v \in V} w_R(v).$$

Since $V \subseteq R$, we have $v^* = \bigsqcap_{v \in V} v \in \text{closure}(R)$, because of the definition of the closure of a relation. Since $\hat{s} \in S$, we have $t \sqsubseteq \hat{s}|_X$ (because of the definition of S), and since $\forall v \in V : \hat{s} \sqsubseteq v$, we have $\hat{s} \sqsubseteq v^*$ (because the intersection of a set of tuples is the least specific tuple that is at least as specific as all tuples in the set), hence $t \sqsubseteq v^*|_X$. It follows that $v^* \in C$.

Let $W = \{w \in R \mid v^* \sqsubseteq w\}$, i.e. let W be the set of tuples from which the weight of v^* is computed. Since $v^* = \bigsqcap_{v \in V} v$ (due to the definition of v^*), we have $\forall v \in V : v^* \sqsubseteq v$ (due to the fact that the intersection of a set of tuples is at least as specific as all tuples in the set), and hence $V \subseteq W$. Putting everything together we arrive at

$$\begin{aligned} \pi_X^{(\text{closure}(D))}(t) &= \frac{1}{w_0} \max_{c \in C} w_{\text{closure}(R)}(c) \\ &\geq \frac{1}{w_0} w_{\text{closure}(R)}(v^*) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{w_0} \sum_{w \in W} w_R(w) \\
&\geq \frac{1}{w_0} \sum_{v \in V} w_R(v) \\
&= \pi_X^{(\text{support}(D))}(t).
\end{aligned}
$$

From what we have considered, the first inequality need not be an equality, since there may be another tuple in closure(R) to which a higher weight was assigned. The second inequality need not be an equality, because W may contain more tuples than V.

(2) $\pi_X^{(\text{closure}(D))}(t) \leq \pi_X^{(\text{support}(D))}(t)$:

Again we have to distinguish two cases, namely $C = \emptyset$ and $C \neq \emptyset$, the first of which is obviously trivial.

(a) $C = \emptyset$: $\pi_X^{(\text{closure}(D))}(t) = 0 \leq \pi_X^{(\text{support}(D))}(t) \in [0, 1]$.

(b) $C \neq \emptyset$: Due to the definitions of $\pi_X^{(\text{closure}(D))}$ and $w_{\text{closure}(R)}$

$$
\begin{aligned}
\pi_X^{(\text{closure}(D))}(t) &= \frac{1}{w_0} \max_{c \in C} w_{\text{closure}(R)}(c) \\
&= \frac{1}{w_0} \max_{c \in C} \sum_{u \in R, c \sqsubseteq u} w_R(u).
\end{aligned}
$$

Let $\hat{c} \in C$ be (one of) the tuple(s) $c \in C$ for which $w_{\text{closure}(R)}(c)$ is maximal. Let $W = \{w \in R \mid \hat{c} \sqsubseteq w\}$, i.e. let W be the set of tuples from which the weight of $\hat{c}$ is computed. Then

$$
\pi_X^{(\text{closure}(D))}(t) = \frac{1}{w_0} w_{\text{closure}(R)}(\hat{c}) = \frac{1}{w_0} \sum_{w \in W} w_R(w).
$$

Let $Q = \{q \in T_X^{(\text{precise})} \mid q \sqsubseteq \hat{c}\}$, i.e. let Q be the set of tuples "supporting" $\hat{c}$. Since $t \in T_X^{(\text{precise})}$ and $t \sqsubseteq \hat{c}|_X$ (due to $\hat{c} \in C$), there must be a tuple $s^* \in Q$, for which $t \sqsubseteq s^*|_X$. Since $s^* \in Q$, we have $s^* \sqsubseteq \hat{c} \in \text{closure}(R)$ (due to the definition of Q), and since $\forall c \in \text{closure}(R) : \exists u \in R : c \sqsubseteq u$ (due to the definition of the closure of a relation), it follows that $\exists u \in R : s^* \sqsubseteq u$ and hence we have $s^* \in \text{support}(R)$.

Let $V = \{v \in R \mid s^* \sqsubseteq v\}$, i.e. let V be the set of tuples from which the weight of s^* is computed. Since $s^* \sqsubseteq \hat{c}$ (see above), we have $\forall w \in W : s^* \sqsubseteq w$ and hence $W \subseteq V$. Thus we arrive at

$$
\begin{aligned}
\pi_X^{(\text{support}(D))}(r) &= \frac{1}{w_0} \max_{s \in S} w_{\text{support}(R)}(s) \\
&\geq \frac{1}{w_0} w_{\text{support}(R)}(s^*) \\
&= \frac{1}{w_0} \sum_{v \in V} w_R(v) \\
&\geq \frac{1}{w_0} \sum_{w \in W} w_R(w) \\
&= \pi_X^{(\text{closure}(D))}(t).
\end{aligned}
$$

The reasons underlying the inequalities are similar to those in (1).
From (1) and (2) it follows that, since t is arbitrary,

$$\forall t \in T_X^{(\text{precise})} : \quad \pi_X^{(\text{closure}(D))}(t) = \pi_X^{(\text{support}(D))}(t).$$

This completes the proof.

References

[Andersen *et al.* 1989] 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 (IJCAI'89, Detroit, MI, USA)*, 1080–1085. Morgan Kaufmann, San Mateo, CA, USA 1989

[Bauer *et al.* 1997] E. Bauer, D. Koller, and Y. Singer. Update Rules for Parameter Estimation in Bayesian Networks. *Proc. 13th Conf. on Uncertainty in Artificial Intelligence (UAI'97, Providence, RI, USA)*, 3–13. Morgan Kaufmann, San Mateo, CA, USA 1997

[Borgelt *et al.* 1996] C. Borgelt, J. Gebhardt, and R. Kruse. Concepts for Probabilistic and Possibilistic Induction of Decision Trees on Real World Data. *Proc. 4th European Congress on Intelligent Techniques and Soft Computing (EUFIT'96, Aachen, Germany)*, Vol. 3:1556–1560. Verlag Mainz, Aachen, Germany 1996

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

[Borgelt and Kruse 1998] C. Borgelt and R. Kruse. Efficient Maximum Projection of Database-Induced Multivariate Possibility Distributions. *Proc. 7th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'98, Anchorage, Alaska, USA)*, CD-ROM. IEEE Press, Piscataway, NJ, USA 1998

[Borgelt 2000] C. Borgelt. *Data Mining with Graphical Models*. Ph.D. Thesis, Otto-von-Guericke-University of Magdeburg, Germany 2000

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

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

[Chickering *et al.* 1994] D.M. Chickering, D. Geiger, and D. Heckerman. *Learning Bayesian Networks is NP-Hard (Technical Report MSR-TR-94-17)*. Microsoft Research, Advanced Technology Division, Redmond, WA, USA 1994

[Chow and Liu 1968] 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

[Cooper and Herskovits 1992] 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

[Dechter and Pearl 1992] R. Dechter and J. Pearl. Structure Identification in Relational Data. *Artificial Intelligence* 58:237–270. North-Holland, Amsterdam, Netherlands 1992

[Dempster *et al.* 1977] A.P. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society (Series B)* 39:1–38. Blackwell, Oxford, United Kingdom 1977

[Dubois and Prade 1988] D. Dubois and H. Prade. *Possibility Theory.* Plenum Press, New York, NY, USA 1988

[Dubois *et al.* 1996] D. Dubois, H. Prade, and R. Yager, eds. *Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications.* J. Wiley & Sons, New York, NY, USA 1996

[Friedman 1998] N. Friedman. The Bayesian Structural EM Algorithm. *Proc. 14th Conf. on Uncertainty in Artificial Intelligence (UAI'98, Madison, WI, USA)*, 80–89. Morgan Kaufmann, San Mateo, CA, USA 1997

[Gebhardt and Kruse 1993] J. Gebhardt and R. Kruse. The Context Model — An Integrating View of Vagueness and Uncertainty. *Int. Journal of Approximate Reasoning* 9:283–314. North-Holland, Amsterdam, Netherlands 1993

[Gebhardt and Kruse 1995] 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

[Gebhardt and Kruse 1996a] J. Gebhardt and R. Kruse. POSSINFER — A Software Tool for Possibilistic Inference. In: [Dubois *et al.* 1996], 407–418

[Gebhardt and Kruse 1996b] J. Gebhardt and R. Kruse. Tightest Hypertree Decompositions of Multivariate Possibility Distributions. *Proc. 7th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU'96, Granada, Spain)*, 923–927. Universidad de Granada, Spain 1996

[Gebhardt 1997] J. Gebhardt. *Learning from Data: Possibilistic Graphical Models.* Habilitation Thesis, University of Braunschweig, Germany 1997

[Hammersley and Clifford 1971] J.M. Hammersley and P.E. Clifford. *Markov Fields on Finite Graphs and Lattices.* Unpublished manuscript, 1971. Cited in: [Isham 1981]

[Hartley 1928] R.V.L. Hartley. Transmission of Information. *The Bell Systems Technical Journal* 7:535–563. Bell Laboratories, USA 1928

[Heckerman 1991] D. Heckerman. *Probabilistic Similarity Networks.* MIT Press, Cambridge, MA, USA 1991

[Heckerman *et al.* 1995] 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

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

[Isham 1981] V. Isham. An Introduction to Spatial Point Processes and Markov Random Fields. *Int. Statistical Review* 49:21–43. Int. Statistical Institute, Voorburg, Netherlands 1981

[Jamshidian and Jennrich 1993] M. Jamshidian and R.I. Jennrich. Conjugate Gradient Acceleration of the EM Algorithm. *Journal of the American Statistical Society* 88(412):221–228. American Statistical Society, Providence, RI, USA 1993

[Jordan 1998] M.I. Jordan, ed. *Learning in Graphical Models.* MIT Press, Cambridge, MA, USA 1998

[Klir and Mariano 1987] 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

[Kononenko 1995] I. Kononenko. On Biases in Estimating Multi-Valued Attributes. *Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining (KDD'95, Montreal, Canada)*, 1034–1040. AAAI Press, Menlo Park, CA, USA 1995

[Kruse and Schwecke 1990] R. Kruse and E. Schwecke. Fuzzy Reasoning in a Multidimensional Space of Hypotheses. *Int. Journal of Approximate Reasoning* 4:47–68. North-Holland, Amsterdam, Netherlands 1990

[Kruse *et al.* 1991] R. Kruse, E. Schwecke, and J. Heinsohn. *Uncertainty and Vagueness in Knowledge-based Systems: Numerical Methods (Series: Artificial Intelligence).* Springer, Berlin, Germany 1991

[Kruse *et al.* 1994] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*, J. Wiley & Sons, Chichester, United Kingdom 1994.

[Kruskal 1956] 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

[Lauritzen and Spiegelhalter 1988] 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

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

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

[Quinlan 1993] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, USA 1993

[Rasmussen 1992] 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

[Saffiotti and Umkehrer 1991] A. Saffiotti and E. Umkehrer. PULCINELLA: A General Tool for Propagating Uncertainty in Valuation Networks. *Proc. 7th Conf. on Uncertainty in Artificial Intelligence (UAI'91, Los Angeles, CA, USA)*, 323–331. Morgan Kaufmann, San Mateo, CA, USA 1991

[Shachter *et al.* 1990] R.D. Shachter, T.S. Levitt, L.N. Kanal, and J.F. Lemmer, eds. *Uncertainty in Artificial Intelligence 4.* North Holland, Amsterdam, Netherlands 1990

[Shannon 1948] C.E. Shannon. The Mathematical Theory of Communication. *The Bell Systems Technical Journal* 27:379–423. Bell Laboratories, USA 1948

[Shenoy 1992] P.P. Shenoy. Valuation-based Systems: A Framework for Managing Uncertainty in Expert Systems. In: [Zadeh and Kacprzyk 1992], 83–104

[Ullman 1988] J.D. Ullman. *Principles of Database and Knowledge-Base Systems, Vol. 1 & 2.* Computer Science Press, Rockville, MD, USA 1988

[Verma and Pearl 1990] T.S. Verma and J. Pearl. Causal Networks: Semantics and Expressiveness. In: [Shachter *et al.* 1990], 69–76

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

[Zadeh and Kacprzyk 1992] L.A. Zadeh and J. Kacprzyk. *Fuzzy Logic for the Management of Uncertainty.* J. Wiley & Sons, New York, NY, USA 1992