Data Mining with Possibilistic Graphical Models

Christian Borgelt and Rudolf Kruse

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

Abstract: *Data Mining*, also called *Knowledge Discovery in Databases*, is a young area of research, which has emerged in response to the flood of data we are faced with nowadays. It has taken up the challenge to develop techniques that can help humans discover useful patterns in their data. One such technique—which certainly is among the most important, as it can be used for frequent data mining tasks like classifier construction and dependence analysis—are *graphical models* and especially learning such models from a dataset of sample cases. In this paper we review the basic ideas of graphical modeling, with a focus on possibilistic networks, and study the principles of learning such graphical models from a dataset of sample cases.

1 Introduction

Today electronic information processing systems are used by almost every company, in departments like production, marketing, stockkeeping, or personnel. These systems were developed, because it turned out to be very important to be able to find certain information, for example the address of a customer, in a fast and reliable way. Today, however, due to increasingly powerful computers and advances in database and software technology, we may also think about using such collections of data not only for retrieving specific information that is needed at a given moment, but also to search for more general knowledge that is hidden in them. If, for instance, a supermarket analyzes the receipts of its customers (which are easily collectible with scanner cashiers) and thus discovers that certain products are frequently bought together, turnover of these products may be increased by properly arranging them on the shelves of the market.

Unfortunately, in order to find such hidden knowledge, the retrieval capacities of standard database systems and the methods of classical data analysis are rarely sufficient. These only allow us to retrieve individual data items as well as to compute simple aggregations like average regional sales. We may also test hypotheses like whether the day of the week has any influence on the production quality. More general patterns, structures, or regularities, however, go undetected. But often knowing these patterns would make it possible, for example, to increase turnover or product quality. Consequently, in recent years we have seen the emergence of a new research area—often called "Knowledge Discovery in Databases" (KDD) or "Data Mining" (DM)—which focuses on automatically generating and testing hypotheses and models that describe the regularities in a given (large) dataset. Hypotheses and models found in this way are then used to make predictions and to justify decisions.

In this paper we concentrate on a single data mining method, namely the automatic construction of graphical models from a dataset of sample cases. This method is very important, because it can be used to tackle such frequent data mining tasks as classifier construction and dependence analysis. Our exposition focuses on possibilistic graphical models, which are introduced as fuzzyfications of relational graphical models.

2 Graphical Models: A Simple Example

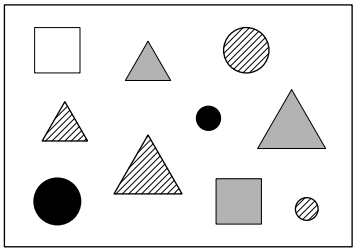
The idea underlying graphical modeling is most easily explained with a simple example, which we study first in the relational and then in the possibilistic setting. The relational case has the advantage that we can neglect degrees of possibility, which may obscure the very simple structure. Possibilistic graphical models are then introduced as straightforward generalizations of relational models and are thus somewhat easier to understand than their probabilistic counterparts, although the basic structure is identical.

Our example domain consists of a set of geometrical objects, as shown in Figure 1. These objects are characterized by three attributes: color (or hatching), shape, and size. As already indicated, we neglect degrees of possibility for the time being and consider only whether a certain state, i.e., a certain combination of attribute values, is possible or not. This enables us to represent the objects as a simple relation, which is shown in the table in Figure 1.

Suppose that an object of the set is chosen at random, but let us assume that not all attributes of the object can be observed. We may imagine, for example, that the object is drawn from a box at some distance, so that we can see the color, but cannot discern the shape or the size. We know, however, that there are only ten objects with certain values of the three attributes. How can we use this information to draw inferences about the unobserved properties?

Problems of this kind frequently occur in applications, for instance, in medical diagnosis: From textbooks and experience a physician knows about the dependences between diseases and symptoms, perhaps in the context of other properties of the patient, like age or sex. But he can only observe or ask for symptoms as well as age, sex, the patient's history etc. Which disease or diseases are present he has to infer with the help of his medical knowledge.

In our illustrative example the solution is, of course, trivial: Simply traverse the table and discard from it all objects with a different color than the one observed, then collect the possible shapes and sizes from the rest. However, this is possible only,



color	shape	size
■	○	small
■	○	medium
▨	○	small
▨	○	medium
▨	△	medium
▨	△	large
□	□	medium
▩	□	medium
▩	△	medium
▩	△	small

Figure 1: A set of simple geometrical objects and the corresponding relation.

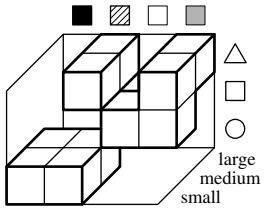


Figure 2: The reasoning space of the simple geometrical objects example shown in Figure 1. Each cube represents one tuple of the relation.

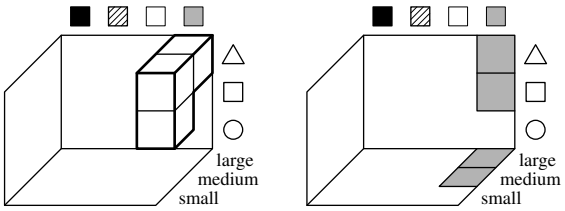


Figure 3: Reasoning in the space as a whole consists in restricting the relation to the "slice" that corresponds to the observation made.

because we have merely ten objects and three attributes. In medical diagnosis this procedure is inapplicable, because the table we had to construct would be much too large to process. Therefore we have to structure the medical knowledge of the physician appropriately, for example, by decomposing it into dependences between few attributes.

Although our example is considerably simpler than the complex domains we have to handle in practice, it can be used to demonstrate how voluminous (tabular) knowledge can be decomposed, so that it becomes manageable. The table describing the geometrical objects can be decomposed, without loss, into two smaller tables, from which it can be reconstructed. We illustrate this by representing the domain as a three-dimensional space, each dimension of which we associate with an attribute. In this way each possible combination of attribute values can be represented by a cube in this space, see Figure 2.

Let us assume that the randomly chosen object is grey. In the representation just described, the naive way of reasoning consists in cutting out the "slice" that is associated with the color grey, as shown in Figure 3. In this way we infer that the object cannot be a circle, but must be square or a triangle, and that it cannot be small, but must be medium or large. However, this inference can also be drawn in a different way, since the knowledge about the objects can be decomposed into so-called projections to two-dimensional subspaces. All possible such projections are shown in Figure 4. They result as shadows thrown by the cubes if light sources are imagined (in sufficient distance) in front, to the right, and above the reasoning space shown in Figure 1.

The relation can be decomposed into the projections to the back plane and to the left plane of the reasoning space, because it can be reconstructed from them. This is demonstrated in Figure 5: First we form the so-called cylindrical extensions of the two projections. that is, we add all values of the missing dimensions. (The name "cylindrical extension" for this operation is derived from the common practice to sketch sets as circles: Adding a dimension to a circle yields a cylinder.) The resulting three-dimensional relations are intersected, i.e., only cubes contained in both are kept. The result is shown in Figure 5. Obviously it coincides with the original relation (cf. Figure 2).

The advantage of relational decomposition is that it can be exploited to draw inferences, without having to reconstruct the three-dimensional representation first. This is demonstrated in Figure 6. First the observation that the object is grey is extended cylindrically to the subspace color×shape (hatched column) and intersected with the projection of the relation to this subspace (grey fields). The result is projected to the shape dimension. From this projection we read, just as we found out above, that the object cannot be a circle, but must be a square or a triangle. Analogously this result is

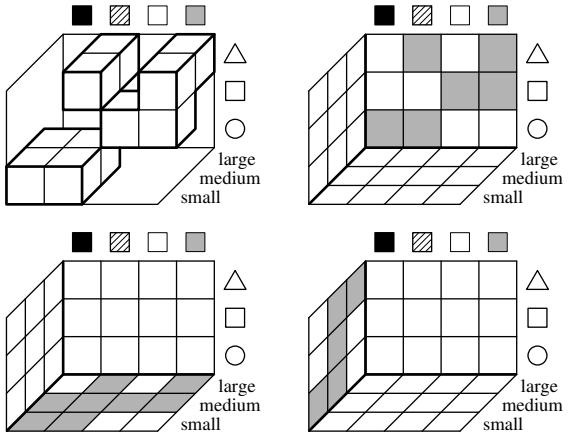


Figure 4: Projections of the relation shown in Figure 1 to the three possible two-dimensional subspaces. They are the shadows thrown by the cubes if light sources are imagined in front, to the left, and above the reasoning space.

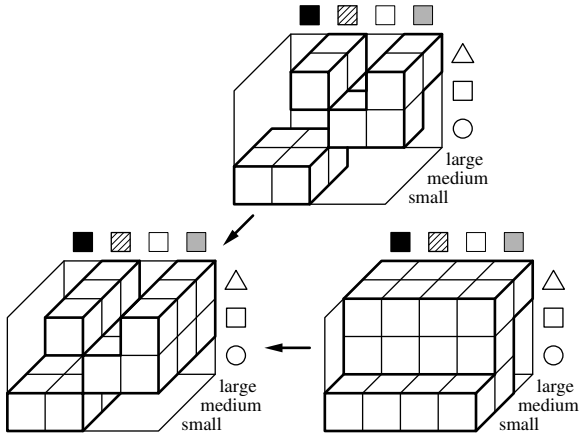


Figure 5: Cylindrical extensions of two projections of the relation depicted in Figure 1 and their intersection. This demonstrates that the relation can be decomposed into two two-dimensional projections.

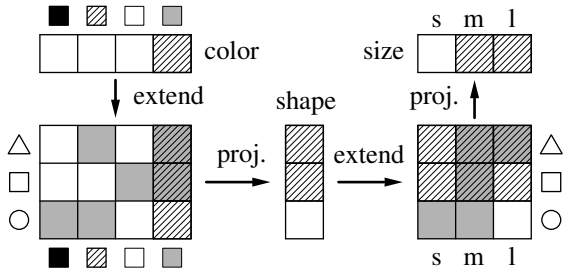


Figure 6: Propagating the evidence that the object is grey. It is not necessary to reconstruct the original relation: We can work with the projections directly.

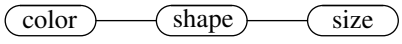


Figure 7: Network representation

extended cylindrically to the subspace shape×size (hatched row), intersected with the projection of the relation to this subspace (grey fields), and finally projected to the size dimension. This yields that the object cannot be small, but must be medium or large.

This reasoning procedure suggests to represent the reasoning space as a graph or network, as shown in Figure 7. Each node of this network stands for an attribute and the edges indicate which projections are needed. It should be noted, though, that the subspaces are not always two-dimensional as in this very simple example. In applications the subspaces may have three, four, or more dimensions. Accordingly, the edges in the corresponding network then connect more than two nodes (thus forming the *hyperedges* of so-called *hypergraphs*).

Furthermore it should be noted that the projections have to be chosen carefully:

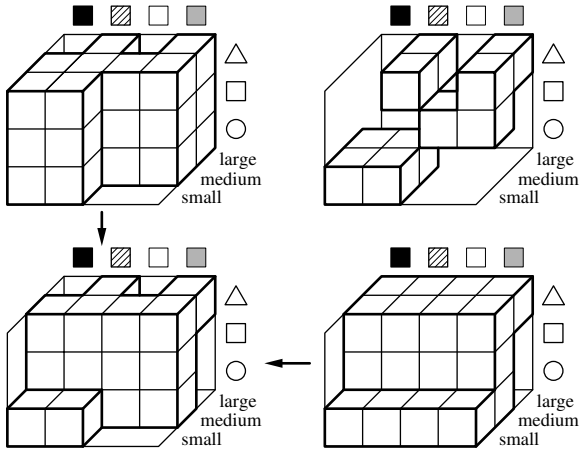


Figure 8: Not all choices of two projections are decompositions. If the wrong projections are selected, the intersection of their cylindrical extensions can contain many additional tuples.

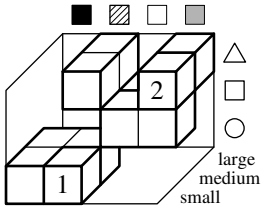


Figure 9: It is not always possible to decompose a relation. In this case approximations may have to be accepted.

Not just any two projections will do. This is demonstrated in Figure 8, where instead of the projection to the back plane we use the projection to the bottom plane. The intersection of the cylindrical extensions of these two projections, which is shown on the bottom left in Figure 8 differs considerably from the original relation, which is shown again on the top right.

But not only have the projections to be chosen with care, sometimes it is not even possible to find a decomposition. To see this consider Figure 9, in which two cubes are marked. Suppose first that the cube marked 1 is removed. It is easily verified that the resulting relation can no longer be decomposed into two projections to two subspaces. However, it is still possible to reconstruct the original relation be using all three possible two-dimensional projections: The intersection with the third projection (to the bottom plane) removes the superfluous cube 1. If, however, the cube marked 2 is removed, the relation cannot be decomposed any more. Removing this cube does not change any of the projections: In all three directions there is still another cube throwing the shadow. Hence the cube marked 2 is contained in all intersections of projections to subspaces.

Such situations are common in practice. But since in applications it is usually impossible to manage the domains under consideration without decomposition, approximations have to be accepted. That is, if no (exact) decomposition is possible, it is tried to find a set of subspaces of limited size so that the intersection of the cylindrical extensions of the projections to these subspaces contains as few additional tuples as possible (obviously, an approximate decomposition can contain only additional tuples).

The idea underlying relational graphical models is easily generalized to probabilistic and possibilistic graphical models. Here we confine to the possibilistic case, though. (Details about probabilistic graphical models can be found, for instance, in [Pearl 1988, Jensen 1996, Lauritzen 1996, Castillo *et al.* 1997].) In the possibilistic setting the (binary) information whether a combination of attribute values is possible or not is replaced by a *degree of possibility* [Dubois and Prade 1988], the semantics of which we consider in more detail below.

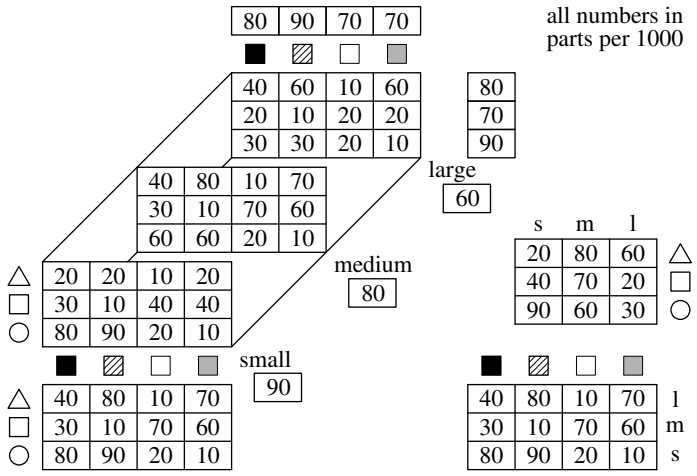


Figure 10: A simple possibility distribution that can be decomposed, just like the relation studied above, into the marginal distributions on two subspaces.

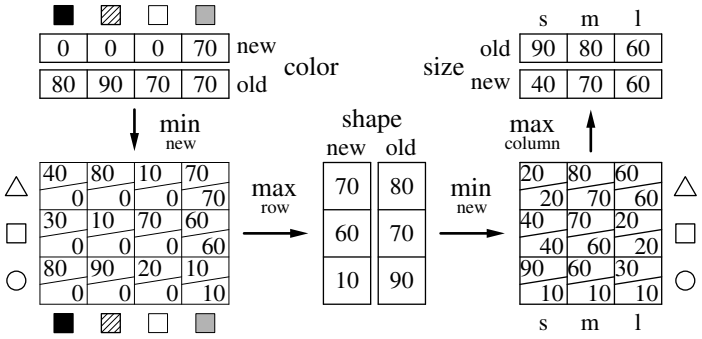


Figure 11: Propagation of the evidence that the object is grey, using only the two marginal distributions.

As an example consider the three-dimensional possibility distribution shown in Figure 10, which is defined on the same space as the relation considered above. The only difference is that tuples that were contained in the relation now have a high degree of possibility, while tuples that were missing have a low degree of possibility. The marginal distributions, which take the place of the shadow projections, are computed by taking the maximum over the dimension along which the projection is carried out.

Like the relation studied above, this possibility distribution can be decomposed into the marginal distributions on the two subspaces color×shape and shape×size. From these marginal distributions the original three-dimensional distribution can be reconstructed by computing the minimum of corresponding marginals. For instance, the value 20 for small black triangles is computed as the minimum of the value 40 for black triangles and the value 20 for small triangles.

As in the relational case the possibility to decompose the distribution enables us to draw inferences using only the marginal distributions that form the decomposition without having to reconstruct the original three-dimensional distribution. This is demonstrated in Figure 11, assuming again that the randomly chosen object is observed to be green. The reasoning procedure is exactly the same as in the relational case (cf. Figure 6): The evidence that the object is green is extended cylindrically to the subspace color×shape (setting all values in the same colum to the marginal value) and intersected with the marginal distribution on this space (upper numbers) by taking the minimum. This yields the new distribution (lower numbers), which is projected to the shape dimension to obtain the degrees of possibility of the different shapes by taking the maximum over rows. The second step is analogous. The shape information is extended cylindrically to the subspace shape×size and intersected with the marginal

distribution on this space (upper numbers) by taking the minimum. The resulting distribution (lower numbers) is projected to the size dimension by taking the maximum, thus yielding degrees of possibility for the different sizes.

3 Graphical Models: General Characterization

Based on the intuition conveyed with the simple examples of the preceding section, we now turn to a more formal characterization of graphical models.

3.1 Decomposition

The decomposition underlying relational graphical models is, of course, well-known from the theory of relational databases [Ullman 1988] and actually relational database theory is strongly connected to the theory of graphical models. The connection is brought about by the notion of the *join-decomposability* of a relation, which in relational databases is exploited to store a high-dimensional relation with less redundancy and, obviously, using less storage space.

The idea underlying join-decomposability is that often a relation can be reconstructed from certain *projections* of it by forming their so-called *natural join*. Formally, this can be described as follows: Let $U = \{A_1, \ldots, A_n\}$ be a set of attributes and let $\mathrm{dom}(A_i)$ be their respective domains. Furthermore, let r_U be a relation over U. We represent this relation by its *indicator function*, which assigns a value of 1 to all tuples contained in the relation and a value of 0 to all tuples not contained in it. The tuples themselves are represented as conjunctions $\bigwedge_{A_i \in U} A_i = a_i$, which state a value for each of the attributes. Using an indicator function a projection r_M of the relation r_U to a subset M of the attributes in U can easily be defined by

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

where the somewhat sloppy notation w.r.t. the maximum is meant to indicate 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 operation used 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 we can always reconstruct the original relation.

The decomposition scheme we just outlined for the relational case is easily transferred to the possibilistic case: We only have to extend the range of values of the indicator function to the real interval $[0, 1]$, i.e., we use *possibility distributions* instead, thus "fuzzifying" relational graphical models. In this way a gradual possibility of a tuple is modeled. The decomposition formula is identical:

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

To define semantics of *degrees of possibility* we rely on the *context model* [Gebhardt and Kruse 1993]: Suppose that for a description of the modeled domain 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 describe 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—the modeled domain may be in under the physical or observation-related frame 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 domain, which we can give for the context c. By "most specific set-valued specification" we mean that we can guarantee that $\Gamma(c)$ contains t_0, but that we cannot guarantee that a proper subset of $\Gamma(c)$ contains t_0. 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.

Formally, the above description results in a *random set* [Nguyen 1978, Hestir *et al.* 1991] (i.e., a set-valued random variable) $\Gamma : C \to 2^T$ as an imperfect (i.e., imprecise and uncertain) specification of the actual state t_0 of the modeled section of the world. From it we derive a possibility distribution by simply computing its *one-point coverage*

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

With this interpretation a possibility distribution represents uncertain *and* imprecise knowledge as can be seen by comparing it to a probability distribution and to a relation. A probability distribution covers *uncertain*, but *precise* knowledge. This becomes obvious if one notices that a possibility distribution in the interpretation described above reduces to a probability distribution if $\forall c \in C : |\Gamma(c)| = 1$, 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.

3.2 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. 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, 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 specify which is the conditioned attribute and which are the conditions. Note, however, that this does not make much of a difference in the relational and the possibilistic case, since here conditional distributions are simple identified with the corresponding marginal distributions, i.e.,

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

Secondly, graphs can be used to describe (conditional) dependence and independence relations between attributes via the concept of *separation* of nodes. What is to be understood by "separation" depends again on whether the graph is directed or undirected. If it is undirected, separation 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. Or, 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 graphs, which have to be acyclic, 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, 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 with converging edges either is in Z or has a descendant in Z,
2. every other node is not in Z.

These separation criteria are 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 C given B) that

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

This formula indicates the close connection of conditional independence and decomposability. Formally, this connection between conditional independence graphs and graphs that describe decompositions is established by theorems 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 certain restrictions have to be introduced [Gebhardt 1997, Borgelt and Kruse 2002].

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 can be found, for instance, in [Jensen 1996, Castillo *et al.* 1997].

4 Learning From Data: A Simple Example

Having reviewed the ideas underlying graphical models, we now turn to the problem how we can find a decomposition if there is one and how we can find a good approximation otherwise. If there is a human expert of the modeled application domain, we may ask him to specify an appropriate conditional independence graph together with the necessary distributions. However, we may also try to find a decomposition automatically by analyzing a dataset of example cases. In the following we study the basic ideas underlying such learning from data using again our simple geometrical objects example.

Suppose that we are given the table shown in Figure 1 and that we desire to find a(n approximate) relational decomposition, maybe satisfying certain complexity con-

subspace	relative number of possible comb.	Hartley info. gain
color×shape	$\frac{6}{12} = \frac{1}{2} = 50\%$	$\log_2 \frac{12}{6} = 1$
color×size	$\frac{8}{12} = \frac{2}{3} \approx 67\%$	$\log_2 \frac{12}{8} \approx 0.58$
shape×size	$\frac{5}{9} = \frac{5}{9} \approx 56\%$	$\log_2 \frac{9}{5} \approx 0.85$

Table 1: Selection criteria for projections. Choosing the subspaces with the smallest (second column) or highest (third column) values yields the decomposition.

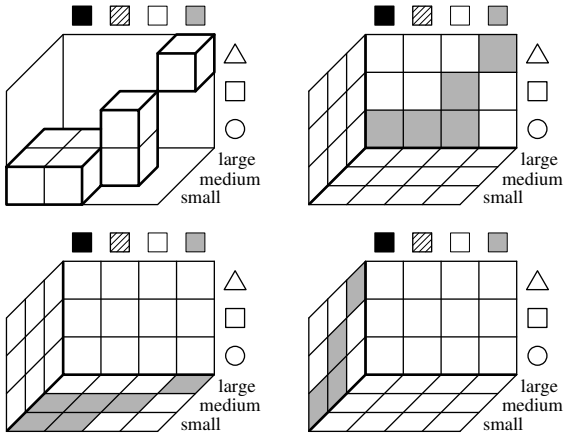


Figure 12: The selection criteria are only heuristics that not always find a possible decomposition: The relation in this figure is decomposable into the projection to the back plane and the projection to the bottom plane. However, the selection criteria do not yield this decomposition.

straints. Of course, we could check all possible graphical models and count the additional tuples. However, such an approach, though feasible for our simple example, is prohibitive in practice due to the very high number of possible graph structures. It would be convenient if we could read from a projection (marginal distribution) whether we need it in a decomposition or not. Fortunately, there is indeed a simple heuristic criterion, which can be used for such local assessments and provides good chances to find an appropriate decomposition.

The basic idea is very simple: The intersection of the cylindrical extensions of the projections should contain as few additional tuples as possible, in order to be as close as possible to the relation to decompose. It is surely plausible that the intersection contains few combinations of attribute values if this holds already for the cylindrical extensions. However, the number of combinations in the cylindrical extensions depends directly on the number of possible combinations of attribute values in the projections (forming the cylindrical extension only adds all values of the missing dimensions). Therefore we should select such projections, which contain as few combinations of attribute values as possible.

In doing so we should take into account the size of the subspace projected to. The larger this subspace is, the larger the number of combinations will be, although this is not relevant for finding a good decomposition. Therefore we should consider not the absolute, but the relative number of value combinations. For our simple example the values of this criterion are shown in Table 1. It is easy to verify that choosing the subspaces with the smallest number of possible value combinations leads to the correct decomposition. The third column lists the binary logarithm of the reciprocal value of the relative numbers, which is also known as Hartley information gain. It is discussed in more detail below.

Although this simple selection criterion works quite nicely in our simple example, it should be noted that it is not guaranteed to yield the best choice of projections. To see this consider Figure 12, which shows another three-dimensional relation together with

the three possible projections to two-dimensional subspaces. Although this relation can be decomposed into the projections to the back plane and to the bottom plane, the selection criterion just studied does not find this decomposition, but selects the back plane and the left plane.

5 Learning From Data: General Characterization

In general, there are three main approaches to learn a graphical model:

- Test whether a distribution is decomposable w.r.t. a given graph.
  This is the most direct approach. It is not bound to a graphical representation, but can also be carried out w.r.t. other representations of the subsets of attributes to be used to compute the (candidate) decomposition of the given distribution.
- Find an conditional independence graph by conditional independence tests.
  This approach exploits the theorems mentioned in the preceding section, which connect conditional independence graphs and graphs that describe decompositions. 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 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). 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. A relational example in which it fails has been studied in the previous section (cf. Figure 12) and similar ones can be found for the possibilistic setting.

A (computationally feasible) analytical method to construct optimal graphical models from a database 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 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 is apt, since an algorithm that does not explicitly search the space of possible networks

usually searches (heuristically) 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 parents; conditional independence test approaches search the space of all possible conditional independence statements.

5.1 Computing Projections

A basic operation needed to learn a graphical model from a dataset of sample cases is a method to determine the marginal or conditional distributions of a candidate decomposition. Such an operation is necessary, because these distributions are needed to assess the quality of a given candidate graphical model.

If the dataset is precise, i.e., if in all tuples there is exactly one value for each attribute, then computing a projection is trivial, since it consists in counting tuples and computing relative frequencies. However, if the data is imprecise, i.e., contains missing values or set-valued information, things are slightly more complicated. Fortunately, with the context model interpretation of a degree of possibility, we have direct means to handle imprecise values: We simply interpret each imprecise tuple as a description of the set $\Gamma(c)$ of states of the world that are possible in some context c. We can do so, because an imprecise tuple can be rewritten as a set of tuples, namely the set of all precise tuples compatible with it.

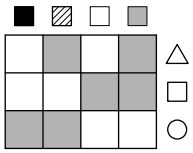
Nevertheless, we face some problems, because we can no longer apply naive methods to determine the marginal distributions (a detailed explanation can be found in [Borgelt and Kruse 2002]). However, there is a simple preprocessing operation by which the database to learn from can be transformed, so that computing maximum projections becomes trivial. This operation is based on the notion of *closure under tuple intersection*. That is, we add (possibly imprecise) tuples to the database in order to achieve a situation, in which for any two tuples from the database their *intersection* (i.e., the intersection of the represented sets of precise tuples) is also contained in the database. Details can be found in [Borgelt and Kruse 1998, Borgelt and Kruse 2002].

5.2 Evaluation Measures

An *evaluation measure* serves to assess the quality of a given candidate graphical model w.r.t. a given database of sample cases, so that it can be determined 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 a score for a maximal clique of the graph to be assessed or a score for a single edge.

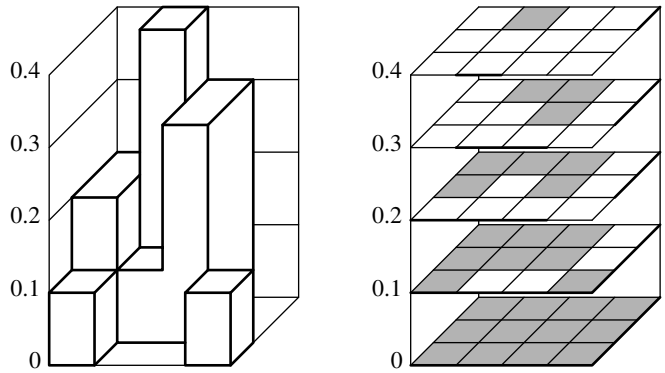
Most such evaluation measures are based on measures of dependence, since for both the second and the third basic approach listed above it is necessary to measure the strength of dependence of two or more variables, either in order to test for conditional independence or in order to find the strongest dependences. Here we confine ourselves to measures that assess the strength of dependence of two variables in the possibilistic case. The transfer to conditional tests (by computing a weighted sum of the results for the different instantiations of the conditions) and to more than two variables is straightforward.

Possibilistic evaluation measures can easily be derived by exploiting the close connection of possibilistic networks to relational networks (see above). The idea is to draw on the α-*cut* view of a possibility distribution. This concept is transferred from the



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

Figure 13: Computation of Hartley information gain.



$$\log_2 1 + \log_2 1 - \log_2 1 = 0$$

$$\log_2 2 + \log_2 2 - \log_2 3 \approx 0.42$$

$$\log_2 3 + \log_2 2 - \log_2 5 \approx 0.26$$

$$\log_2 4 + \log_2 3 - \log_2 8 \approx 0.58$$

$$\log_2 4 + \log_2 3 - \log_2 12 = 0$$

Figure 14: Illustration of the idea of specificity gain.

theory of fuzzy sets [Kruse *et al.* 1994]. In the α-cut view a possibility distribution is seen as a *set of relations* with one relation for each degree of possibility α. The indicator function of such a relation is defined by simply assigning a value of 1 to all tuples for which the degree of possibility is no less than 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. Thus 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.

To make this clearer, we reconsider the simple example studied above. Figure 13 shows the projection to the back plane of our example reasoning space, i.e., to the subspace color×shape. We can measure the strength of dependence of color and shape by computing the *Hartley information gain* [Hartley 1928]

$$\begin{aligned} I_{\text{gain}}^{(\text{Hartley})}(C,S) &= \log_2\Big(\sum_{c\in\text{dom}(C)} r_C(C=c)\Big) + \log_2\Big(\sum_{s\in\text{dom}(S)} r_S(S=s)\Big) \\ &- \log_2\Big(\sum_{c\in\text{dom}(C)}\sum_{s\in\text{dom}(S)} r_{CS}(C=c,S=s)\Big) \\ &= \log_2 \frac{\Big(\sum_{c\in\text{dom}(C)} r_C(C=c)\Big)\Big(\sum_{s\in\text{dom}(S)} r_S(S=s)\Big)}{\sum_{c\in\text{dom}(C)}\sum_{s\in\text{dom}(S)} r_{CS}(C=c,S=s)}, \end{aligned}$$

where C stands for the color and S for the size of an object. The idea underlying this measure is as follows: Suppose we want to determine the actual values of the two attributes C and S. Obviously, 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 value combination. Or we may exploit the fact that the value combination is restricted by the relation shown in Figure 13 and try to determine the value combination directly. In the former case we need the Hartley information of the set of values of C plus the Hartley information of the set of values of S, i.e.

$\log_2 4 + \log_2 3 \approx 3.58$ bits. In the latter case we need the Hartley information of the possible tuples, i.e. $\log_2 6 \approx 2.58$ 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 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.

The Hartley information gain is generalized to the *specificity gain* [Gebhardt and Kruse 1996, Borgelt and Kruse 1997, Borgelt and Kruse 2002] as shown in Figure 14: It is integrated over all α-cuts of a given (two-dimensional) possibility distribution and thus measures the average strength of relational dependence on the different α-levels.

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

Surveys of other evaluation measures—which include probabilistic measures—can be found in [Borgelt and Kruse 1997, Borgelt and Kruse 2002].

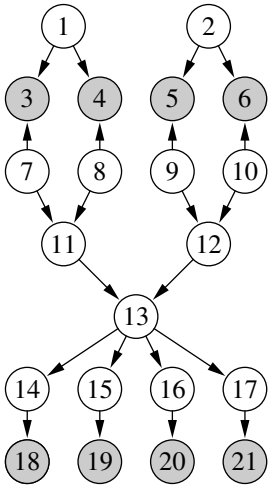
5.3 Search Methods

As already indicated above, 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 (there are $2^{\binom{n}{2}}$ possible undirected graphs over n attributes), heuristic search methods have to be used. Usually these heuristic methods introduce strong restrictions w.r.t. the graphs considered and exploit the value of the chosen evaluation measure to guide the search. In addition they are often greedy w.r.t. the model quality.

The simplest instance of such a search method is, of course, the Kruskal algorithm [Kruskal 1956], which determines an optimum weight spanning tree for given edge weights. This algorithm has been used very early in the probabilistic setting by [Chow and Liu 1968], who used the *Shannon information gain* (also called *mutual information* or *cross entropy*) of the connected attributes as edge weights. In the possibilistic setting, we may simply replace the Shannon information gain by the *specificity gain* in order to arrive at an analogous algorithm [Gebhardt and Kruse 1996, Borgelt and Kruse 2002].

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[1]: At the beginning the value of an evaluation measure is computed for a parentless child attribute. Then in turn each of the parent candidates (i.e. the attributes preceding the child in the topological order) is temporarily added and the evaluation measure is recomputed. The parent candidate yielding the highest value of the evaluation measure is selected as a first parent and permanently added. In the third step each remaining parent candidate is added temporarily as a second parent

[1] A topological order is an order of the nodes of a graph such that all parent nodes of a given node precede it in the order. That is, there cannot be an edge from a node to a node, which precedes it in the topological order. By fixing a topological order in advance, the set of possible graphs is severely restricted and it is ensured that the resulting graph is acyclic.



21 attributes:

1 – dam correct?	11 – offspring phenogroup 1
2 – sire correct?	12 – offspring phenogroup 2
3 – stated dam phenogroup 1	13 – offspring genotype
4 – stated dam phenogroup 2	14 – factor 40
5 – stated sire phenogroup 1	15 – factor 41
6 – stated sire phenogroup 2	16 – factor 42
7 – true dam phenogroup 1	17 – factor 43
8 – true dam phenogroup 2	18 – lysis 40
9 – true sire phenogroup 1	19 – lysis 41
10 – true sire phenogroup 2	20 – lysis 42
	21 – lysis 43

The grey nodes correspond to observable attributes.

Figure 15: Domain expert designed network for the Danish Jersey cattle blood type determination example.

and again the evaluation measure is recomputed. 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 maximal number of parents is reached, or none of the parent candidates, if added, yields a value of the evaluation measure exceeding the best value of the preceding step.

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*. This measure 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 as the evaluation measure. In order to handle multiple parent attributes with it, we simply combine all parents into one pseudo-attribute and compute the specificity gain for this pseudo-attribute and the child attribute.

A more extensive discussion of search methods for learning graphical models from data, which includes a simulated annealing approach, can be found, for example, in [Borgelt and Kruse 2002].

6 An Example Application

As an example of an application we consider the problem of blood group determination of Danish Jersey cattle in the F-blood group system [Rasmussen 1992]. For this problem there is a Bayesian network (a probabilistic graphical model based on a directed acyclic graph) designed by human domain experts, which serves the purpose to verify parentage for pedigree registration.

The world section modeled in this example comprises 21 attributes, eight of which are observable. The size of the domains of these attributes ranges from two to eight values. The total reasoning space has $2^6 \cdot 3^{10} \cdot 6 \cdot 8^4 = 92\,876\,046\,336$ possible states. This number makes it obvious that the knowledge about this world section must be decomposed in order to make reasoning feasible, since it is clearly impossible to store a probability or a degree of possibility for each state. Figure 15 lists the attributes and shows the conditional independence graph of the Bayesian network.

sire correct	true sire ph.gr. 1	stated sire ph.gr. 1 F1	V1	V2
yes	F1	1	0	0
yes	V1	0	1	0
yes	V2	0	0	1
no	F1	0.58	0.10	0.32
no	V1	0.58	0.10	0.32
no	V2	0.58	0.10	0.32

Table 2: An example of a conditional probability distribution that is associated with the conditional independence graph shown in Figure 15.

n	y	y	f1	v2	f1	v2	f1	v2	f1	v2	v2	v2	v2v2	n	y	n	y	0	6	0	6
n	y	y	f1	v2	**	**	f1	v2	**	**	**	**	f1v2	y	y	n	y	7	6	0	7
n	y	y	f1	v2	f1	f1	f1	v2	f1	f1	f1	f1	f1f1	y	y	n	n	7	7	0	0
n	y	y	f1	v2	f1	f1	f1	v2	f1	f1	f1	f1	f1f1	y	y	n	n	7	7	0	0
n	y	y	f1	v2	f1	v1	f1	v2	f1	v1	v2	f1	f1v2	y	y	n	y	7	7	0	7
n	y	y	f1	f1	**	**	f1	f1	**	**	f1	f1	f1f1	y	y	n	n	6	6	0	0
n	y	y	f1	v1	**	**	f1	v1	**	**	v1	v2	v1v2	n	y	y	y	0	5	4	5
n	y	y	f1	v2	f1	v1	f1	v2	f1	v1	f1	v1	f1v1	y	y	y	y	7	7	6	7

Table 3: An extract from the Danish Jersey cattle database.

As described above, a conditional independence graph enables us to decompose the joint distribution into a set of marginal or conditional distributions. In the Danish Jersey cattle example, this decomposition leads to a considerable simplification: Only 308 conditional probabilities have to be specified. An example of a conditional probability table, which is part of the decomposition, is shown in Table 2. It states the conditional probabilities of the phenogroup 1 of the stated sire of a given calf conditioned on the phenogroup 1 of the true sire and whether the sire was correctly identified. The numbers in this table are derived from statistical data and the experience of human domain experts.

Besides the domain expert designed reference structure there is a database of 500 real world sample cases (an extract of this database is shown in Table 3). This database can be used to test learning algorithms for graphical models, because the quality of the learning result can be determined by comparing the constructed graph to the reference structure. However, there is a problem connected with this database, namely that it contains a fairly large number of unknown values—only a little over half of the tuples are complete. (This can already be guessed from the extract shown in Table 3: the stars denote missing values.)

Missing values and set-valued information make it difficult to learn a Bayesian network, because an unknown value can be seen as representing imprecise information: It states that all values contained in the domain of the corresponding attribute are possible, without any known preferences between them. Nevertheless it is still feasible to learn a Bayesian network from the database in this case, since the dependencies are rather strong and thus the small number of complete tuples is still sufficient to recover the underlying structure. However, learning a possibilistic network from the same dataset is much easier, since possibility theory was especially designed to handle imprecise information (see above). Hence no discarding or special treatment of tuples with missing values or set-valued information is necessary.

In order to check this conjecture, we implemented the learning methods discussed above (together with their probabilistic counterparts) in a prototype program called INES (Induction of NEtwork Structures) [Borgelt and Kruse 2002]. The networks induced with different evaluation measures are very similar to the domain expert designed reference structure, even though the reference structure is a Bayesian network, which may differ from the corresponding possibilistic network, since it employs a different notion of conditional independence. Evaluations of the learned networks show that their quality is comparable to that of learned probabilistic networks and the reference structure w.r.t. reasoning.

7 Conclusion

In this paper we reviewed possibilistic graphical models and discussed approaches to learn them from a database of sample cases. Based on the context model interpretation of a degree of possibility imprecise data are easily handled in such a possibilistic approach. W.r.t. learning algorithms a lot of work done in the probabilistic counterpart of this research area can be transferred: All search methods are directly usable, only the evaluation measures have to be adapted. Experiments carried out with an example application show that learning possibilistic networks from data is a noteworthy alternative to the established probabilistic methods if the data to learn from is imprecise.

References

[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 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

[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

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

[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 1996] 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

[Goodman *et al.* 1991] I.R. Goodman, M.M. Gupta, H.T. Nguyen, and G.S. Rogers, eds. *Conditional Logic in Expert Systems*. North-Holland, Amsterdam, Netherlands 1991

[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 *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

[Hestir *et al.* 1991] K. Hestir, H.T. Nguyen, and G.S. Rogers. A Random Set Formalism for Evidential Reasoning. In: [Goodman *et al.* 1991], 209–344

[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

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

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

[Nguyen 1978] H.T. Nguyen. On Random Sets and Belief Functions. *Journal of Mathematical Analysis and Applications* 65:531–542. Academic Press, San Diego, CA, USA 1978

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

[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

[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

[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