Probabilistic and Possibilistic Networks and How To Learn Them from Data

Christian Borgelt and Rudolf Kruse

Dept. of Information and Communication Systems,
Otto-von-Guericke-University of Magdeburg, 39106 Magdeburg, Germany
borgelt@iik.cs.uni-magdeburg.de

Abstract: In this paper we explain in a tutorial manner the technique of reasoning in probabilistic and possibilistic network structures, which is based on the idea to decompose a multi-dimensional probability or possibility distribution and to draw inferences using only the parts of the decomposition. Since constructing probabilistic and possibilistic networks by hand can be tedious and time-consuming, we also discuss how to learn probabilistic and possibilistic networks from a data, i.e. how to determine from a database of sample cases an appropriate decomposition of the underlying probability or possibility distribution.

Keywords: Decomposition, uncertain reasoning, probabilistic networks, possibilistic networks, learning from data.

1. Introduction

Since reasoning in multi-dimensional domains tends to be infeasible in the domains as a whole—and the more so, if uncertainty and/or imprecision are involved—decomposition techniques, that reduce the reasoning process to computations in lower-dimensional subspaces, have become very popular. For example, decomposition based on dependence and independence relations between variables has been studied extensively in the field of graphical modeling [17]. Some of the best-known approaches are Bayesian networks [23], Markov networks [20], and the more general valuation-based networks [27]. They all led to the development of efficient implementations, for example HUGIN [1], PULCINELLA [26], PATHFINDER [12] and POSSINFER [8].

A large part of recent research has been devoted to learning probabilistic and possibilistic networks from data [4, 13, 9], i.e. to determine from a database of sample cases an appropriate decomposition of the probability or possibility distribution on the domain under consideration. Such automatic learning is important, since constructing a network by hand can be tedious and time-consuming. If a database of sample cases is available, as it often is, learning algorithms can take over at least part of the construction task.

In this tutorial paper we survey the basic idea of probabilistic and possibilistic networks and the basic method for learning them from data. In Section 2 we introduce the idea of decomposing multi-dimensional distributions and demonstrate how a decomposition can be used to reason in the underlying multi-dimensional domain. We do so by inspecting decompositions of relations first and then, in Section 3, proceed to decompositions of probability distributions. Section 4 considers the graphical representation of decompositions. Section 5 discusses the general scheme for inducing decompositions from data, which is applied to probability distributions in Section 6.

With Section 7 we start to transfer the ideas of the preceding sections, where they were presented in the relational and the probabilistic setting, to the possibilistic setting. To do so, we first clarify what we understand by a degree of possibility in Section 7. In Section 8 we look at decomposition of and reasoning in possibility distributions, emphasizing the differences to the probabilistic case. Finally, Section 9 discusses how to induce possibilistic networks from data.

2. Decomposition and Reasoning

The basic idea underlying probabilistic as well as possibilistic networks is that a probability or possibility distribution D on a multi-dimensional domain can under certain conditions be decomposed into a set $\{D_1, \ldots, D_n\}$ of (overlapping) distributions on lower-dimensional subspaces. By *multi-dimensional domain* we mean that a state of the universe of discourse can be described by stating the values of a set of attributes. Each attribute—or, more precisely, the set of its possible values—forms a dimension of the domain. Of course, to form a dimension the possible values have to be exhaustive and mutually exclusive. Thus each state corresponds to a single point of the multi-dimensional domain. A *distribution* D assigns to each point of the domain a number in the interval $[0, 1]$, which represents the (prior) probability or the (prior) degree of possibility of the corresponding state. By *decomposition* we mean that the distribution D on the domain as a whole can be reconstructed (at least approximately) from the distributions $\{D_1, \ldots, D_n\}$ on the subspaces.

Such a decomposition has several advantages, the most important being that a decomposition can usually be stored much more efficiently and with less redundancy than the whole distribution. These advantages are the main motive for studying decompositions of relations (which can be seen as special possibility distributions) in database theory [5, 30]. Not surprisingly, database theory is closely connected to our subject. The only difference is that we focus on reasoning, while database theory focuses on storing, maintaining, and retrieving data.

But just being able to store a distribution more efficiently would not be of much use for reasoning tasks, were it not for the possibility to draw inferences in the underlying multi-dimensional domain using only the distributions

Table 1. The relation R_{ABC} stating prior knowledge about the possible combinations of attribute values

A	a_1	a_1	a_2	a_2	a_2	a_2	a_3	a_4	a_4	a_4
B	b_1	b_1	b_1	b_1	b_3	b_3	b_2	b_2	b_3	b_3
C	c_1	c_2	c_1	c_2	c_2	c_3	c_2	c_2	c_2	c_3

$\{D_1, \ldots, D_n\}$ on the subspaces without having to reconstruct the whole distribution D. How this works is perhaps best explained by a simple example, which we present in the relational setting first [6, 16, 18]. We consider only whether a combination of attribute values is possible or not, thus neglecting its probability or degree of possibility. In other words, we restrict ourselves to a distribution that assigns to each point of the underlying domain either a 1 (if the corresponding state is possible) or a 0 (if the corresponding state is impossible). With this restriction the ideas underlying decomposition and reasoning in decompositions can be demonstrated to the novice reader much clearer than in the probabilistic setting, where the probabilities can disguise the very simple structure. Later on we will study the probabilistic and finally the possibilistic case.

Consider three attributes, A, B, and C, with corresponding domains $\text{dom}(A) = \{a_1, a_2, a_3, a_4\}$, $\text{dom}(B) = \{b_1, b_2, b_3\}$, and $\text{dom}(C) = \{c_1, c_2, c_3\}$. Thus the underlying domain of our example is the Cartesian product $\text{dom}(A) \times \text{dom}(B) \times \text{dom}(C)$ or, as we will write as an abbreviation, the three-dimensional space $\{A, B, C\}$.

Table 1 states prior knowledge about the possible combinations of attribute values in the form of a relation R_{ABC}: only the value combinations contained in R_{ABC} are possible. (This relation is to be interpreted under the closed world assumption, i.e. all value combinations not contained in R_{ABC} are impossible.) A graphical representation of R_{ABC} is shown in the top left of Fig. 1: each cube indicates a possible value combination.

The relation R_{ABC} can be decomposed into two two-dimensional relations, namely the two *projections* to the subspaces $\{A, B\}$ and $\{B, C\}$, both shown in the right half of Fig. 1. These projections as well as the projection to the subspace $\{A, C\}$ (shown in the bottom left of Fig. 1) are the shadows thrown by the cubes in the top left of Fig. 1 on the surrounding planes, if light sources are imagined in front, to the right, and above the relation.

Mathematically, a projection of a relation can be defined in the following way. Let $X = \{A_1, \ldots, A_m\}$ be a set of attributes. A *tuple* t over X is a mapping that assigns to each attribute A_i a value $a_{j_i}^{(i)} \in \text{dom}(A_i)$. Assuming an implicit order of the attributes, a tuple t over X can be written $\left(a_{j_1}^{(1)}, \ldots, a_{j_m}^{(m)}\right)$, where each vector element states the value the correspond-

Fig. 1. Graphical representation of the relation R_{ABC} and of all three possible projections to two-dimensional subspaces. Since in this relation the equation $R_{ABC} = \Pi^{\{A,B,C\}}_{\{A,B\}}(R_{ABC}) \bowtie \Pi^{\{A,B,C\}}_{\{B,C\}}(R_{ABC})$ holds, it can be decomposed into two relations on the subspaces $\{A,B\}$ and $\{B,C\}$. This is demonstrated in Fig. 2.

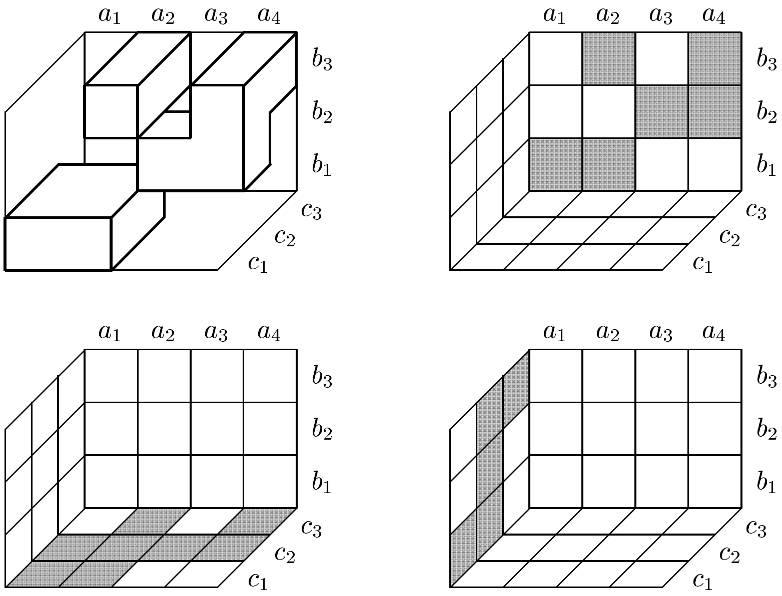


Fig. 2. Cylindrical extensions of two projections of the relation R_{ABC} shown in Fig. 1. On the left is the cylindrical extension of the projection to $\{A,B\}$, on the right the cylindrical extension of the projection to $\{B,C\}$. Their intersection yields the original relation R_{ABC}.

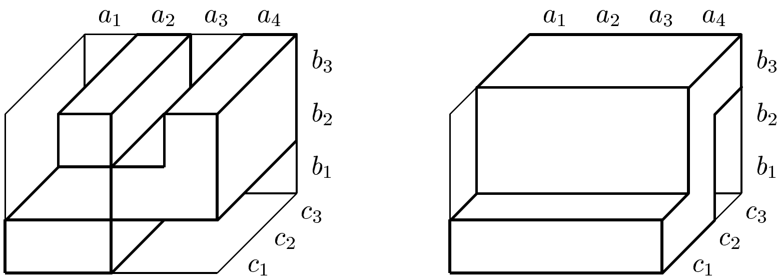
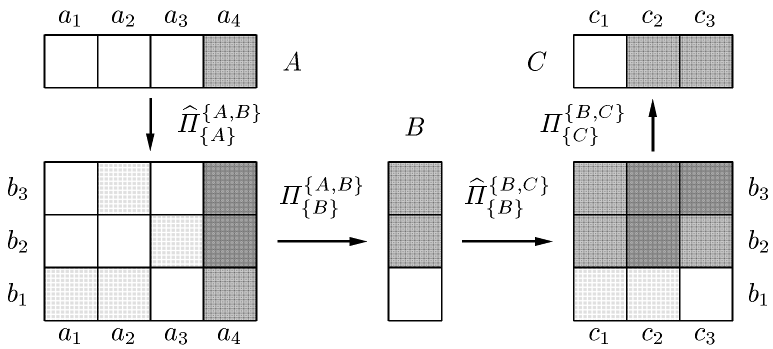


Fig. 3. Propagation of the evidence that attribute A has value a_4 in the three-dimensional relation shown in Fig. 1 using the relations on the subspaces $\{A, B\}$ and $\{B, C\}$



ing attribute is mapped to. To indicate that X is the domain of definition of t, i.e. that t is a tuple over X, we write $\mathrm{dom}(t) = X$. If t is a tuple over X and $Y \subseteq X$, then $t|_Y$ denotes the *restriction* or *projection* of the tuple t to Y, i.e. the mapping $t|_Y$ assigns values only to the attributes in Y. Hence $t|_Y$ is a tuple over Y, i.e. $\mathrm{dom}(t|_Y) = Y$.

A *relation* R over an attribute set X is a set of tuples over X. If R is a relation over X and $Y \subseteq X$, then the *projection* $\Pi^X_Y(R)$ of R from X to Y is defined as

$$\Pi^X_Y(R) \stackrel{\mathrm{def}}{=} \{s \mid \mathrm{dom}(s) = Y \wedge \exists t \in R : s = t|_Y\}.$$

The two relations $R_{AB} = \Pi^{\{A,B,C\}}_{\{A,B\}}(R_{ABC})$ and $R_{BC} = \Pi^{\{A,B,C\}}_{\{B,C\}}(R_{ABC})$ are a decomposition of the relation R_{ABC}, because it can be reconstructed by forming the natural join $R_{AB} \bowtie R_{BC}$. In database theory R_{ABC} would be called *join-decomposable*.

Forming the natural join of two relations is the same as intersecting their *cylindrical extensions* to the union of their attribute sets. The cylindrical extensions of R_{AB} and R_{BC} to $\{A, B, C\}$ are shown in Fig. 2. They result from R_{AB} and R_{BC} by simply adding all possible values of the missing dimension. Thus the name "cylindrical extension" is very expressive: since in sketches a set is usually depicted as a circle, adding all values of a perpendicular dimension yields a cylinder. Mathematically, a cylindrical extension can be defined in the following way. Let R be a relation over an attribute set X and $Y \supseteq X$. Then the *cylindrical extension* $\widehat{\Pi}^Y_X(R)$ of R from X to Y is defined as

$$\widehat{\Pi}^Y_X(R) \stackrel{\mathrm{def}}{=} \{s \mid \mathrm{dom}(s) = Y \wedge \exists t \in R : t = s|_Y\}.$$

It is easy to see that intersecting the cylindrical extensions of R_{AB} and R_{BC} shown in Fig. 2 yields the original relation R_{ABC}. Intuitively this is possible, because fixing the value of attribute B, on which the two relations R_{AB} and R_{BC} overlap, renders the possible values of the remaining attributes A and C freely combinable (see Fig. 1). Hence we can say that given B, the attributes A and C are independent of each other.

To illustrate the reasoning process we assume that from an observation we know that in the current state of the universe of discourse attribute A has value a_4. From this information we can draw inferences about the possible values of the other two attributes B and C. This can be done easily, if we are given the whole distribution as shown in the top left of Fig. 1. We then simply cut out the "slice" corresponding to $A = a_4$ and project the set of possible value combinations in this "slice" to the domains of the attributes B and C. Obviously, we find that according to our prior knowledge neither $B = b_1$ nor $C = c_1$ are possible for the current state of the universe of discourse.

But the same result can also be derived using only the relations R_{AB} and R_{BC}. This is demonstrated in Fig. 3. Starting from the evidence that attribute A has value a_4, we first form the cylindrical extension of the relation $\{(a_4)\}$ to $\{A, B\}$ (medium grey) and intersect it with R_{AB} (light grey). This intersection (R_{AB}^{post}, dark grey) is then projected to $\{B\}$ yielding b_2 and b_3 as possible values for B. In the same way, the relation $\{(b_2), (b_3)\}$ is then extended cylindrically to $\{B, C\}$ (medium grey) and intersected with R_{BC} (light grey). The result (R_{BC}^{post}, dark grey) is projected to $\{C\}$ yielding c_2 and c_3 as possible values for C.

Of course, the reasoning process can also take two observations, for example $A = a_4$ *and* $C = c_3$, as input. To obtain the possible values for B we only have to intersect the projections of $R_{AB}^{\text{post}} = \widehat{\Pi}_{\{A\}}^{\{A,B\}}(\{(a_4)\})$ and $R_{BC}^{\text{post}} = \widehat{\Pi}_{\{C\}}^{\{B,C\}}(\{(c_3)\})$ to $\{B\}$. The result is $\{(b_3)\}$.

It is easy to show that the result of such a reasoning process is always the same as the result obtained directly from the original relation, if intersecting the cylindrical extensions of the projections yields the original relation.

3. Decomposition of Probability Distributions

The method of decomposing a relation can easily be transfered to probability distributions. Only the definitions of projection, cylindrical extension and intersection have to be modified. Projection now consists in calculating the marginal distribution on the subspace. Extension and intersection are combined and consist in multiplying the prior distribution with the quotient of posterior and prior marginal probability. Again the idea is best explained by a simple example.

Fig. 4 shows a probability distribution on the joint domain of the three attributes A, B, and C together with its marginal distributions (sums over

Fig. 4. A three-dimensional probability distribution with its marginal distributions (sums over lines/columns). Since in this distribution the equations $\forall i,j,k : P(a_i,b_j,c_k) = \frac{P(a_i,b_j)P(b_j,c_k)}{P(b_j)}$ hold, it can be decomposed into the marginal distributions on the subspaces $\{A,B\}$ and $\{B,C\}$.

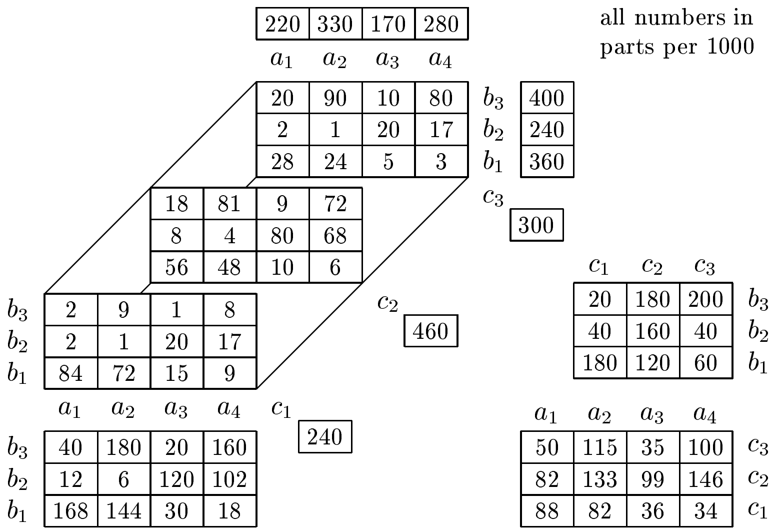
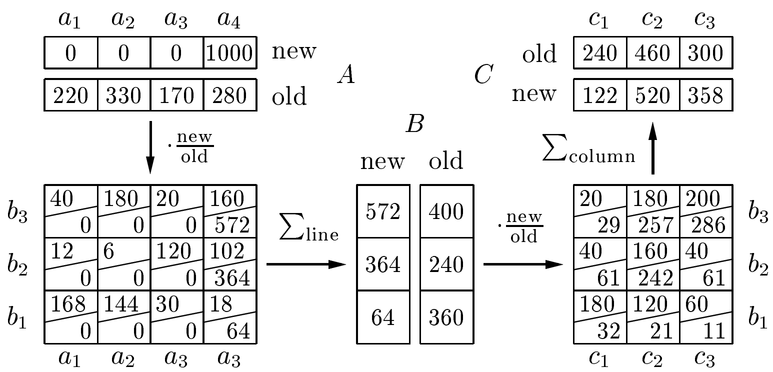


Fig. 5. Propagation of the evidence that attribute A has value a_4 in the three-dimensional probability distribution shown in Fig. 4 using the marginal probability distributions on the subspaces $\{A,B\}$ and $\{B,C\}$



lines/columns). It is closely related to the example of the preceding section, since in this distribution those value combinations that were contained in the relation R_{ABC} (were possible) have a high probability, while those that were missing (were impossible) have a low probability.

Just as the relation R_{ABC} can be decomposed into R_{AB} and R_{BC}, the probability distribution in Fig. 4 can be decomposed into the two marginal distributions on the subspaces $\{A, B\}$ and $\{B, C\}$. This is possible, because the three-dimensional distribution can be reconstructed using the formulae

$$\forall i, j, k : P(a_i, b_j, c_k) = \frac{P(a_i, b_j)P(b_j, c_k)}{P(b_j)},$$

where $P(a_i, b_j, c_k)$ is short for $P(A = a_i, B = b_j, C = c_k)$ etc. These formulae can be derived from the (generally true) formulae

$$\forall i, j, k : P(a_i, b_j, c_k) = P(a_i|b_j, c_k)P(b_j, c_k)$$

by noting that in this probability distribution A is *conditionally independent* of C given B, usually written $A \perp\!\!\!\perp C \mid B$. That is

$$\forall i, j, k : P(a_i|b_j, c_k) = P(a_i|b_j) = \frac{P(a_i, b_j)}{P(b_j)},$$

i.e. if the value of B is known, the value of A does not depend on the value of C. Note, that conditional independence is symmetric, i.e. if $A \perp\!\!\!\perp C \mid B$, then

$$\forall i, j, k : P(c_k|b_j, a_i) = P(c_k|b_j) = \frac{P(c_k, b_j)}{P(b_j)}$$

also holds. In other words, $A \perp\!\!\!\perp C \mid B$ entails $C \perp\!\!\!\perp A \mid B$.

To illustrate the reasoning process, let us assume again that of the current state of the universe of discourse we know that $A = a_4$. Obviously the corresponding probability distributions of B and C can be determined from the three-dimensional distribution by restricting it to the "slice" that corresponds to $A = a_4$ and computing the marginal distributions of that "slice." But the distributions on the two-dimensional subspaces are also sufficient to draw this inference as is demonstrated in Fig. 5. The information that $A = a_4$ is extended to the subspace $\{A, B\}$ by multiplying the joint probabilities by the quotient of posterior and prior probability of $A = a_i$, $i = 1, 2, 3, 4$. Then the marginal distribution on $\{B\}$ is determined by summing over the lines, which correspond to the different values of B. In the same way the information of the new probability distribution on B is propagated to C: the joint distribution on $\{B, C\}$ is multiplied with the quotient of prior and posterior probability of $B = b_j$, $j = 1, 2, 3$, and then the marginal distribution on C is computed by summing over the columns, which correspond to the different values of C. This scheme can be derived directly from the decomposition formulae. It is easy to check that the results obtained are the same as those that follow from the computations on the three-dimensional domain.

Of course, this scheme is a simplification that does not lend itself to direct implementation, as can be seen when assuming that of the current state $A = a_4$ *and* $C = c_3$ are known. In this case additional computations are necessary to join the information from A and C arriving at B. We omit these computations for reasons of simplicity, since our aim is only to illustrate the basic idea.[1]

4. Graphical Representation

The reasoning scheme suggests the idea to use a graphical structure, i.e. an *acyclic hypergraph*, to represent the decomposition: the attributes are represented as nodes, the distributions D_i of the decomposition as hyperedges connecting the attributes of their underlying domains. For the examples of the two preceding sections the hypergraph is simply

$$A \text{ — } B \text{ — } C$$

and hence a normal graph. Of course, in real world applications the resulting hypergraphs can be much more complex, especially the edges can connect more than two nodes, thus forming real *hyper*edges.

This representation uses undirected edges, since in our example the decomposition consists of *joint* marginal distributions. If undirected graphs are used in the probabilistic setting, the network is usually called a *Markov network* [23]. But it is also possible to use *conditional* distributions and directed edges (often accompanied by a refined hypergraph structure), thus arriving at so called *Bayesian networks* [23]. This can be justified by the possibility to write the decomposition formulae in terms of conditional probabilities. E.g. for the above example we can write

$$\begin{aligned} \forall i,j,k: P(a_i,b_j,c_k) &= \frac{P(a_i,b_j)P(b_j,c_k)}{P(b_j)} \\ &= P(a_i)P(b_j|a_i)P(c_k|b_j). \end{aligned}$$

These formulae can also be derived from the so called *chain rule of probability*

$$\forall i,j,k: P(a_i,b_j,c_k) = P(c_k|a_i,b_j)P(b_j|a_i)P(a_i)$$

with the help of the conditional independence $A \perp\!\!\!\perp C \mid B$.

[1] In short: the marginal distributions for B obtained from the two two-dimensional subspaces have to be multiplied with each other, divided by the prior distribution of B and normalized to 1. This can easily be derived from the decomposition formulae.

When using conditional probabilities it seems natural to direct the edges according to their inherent direction, i.e. from the conditioning attributes to the conditioned. In the above example the decomposition would be represented by the directed hypergraph

$$A \to B \to C$$

Usually only conditional dependences of one attribute (called the *child*) given a set of other attributes (called the *parents*) are used, although in principle it is possible to have joint conditional probabilities of two or more attributes given a set of other attributes.

The name *Bayesian* network stems from the fact that, if we want to propagate evidence *against* the direction of a hyperedge, we now have to use *Bayes' formula*

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

to reverse the conditional probability associated with the hyperedge. (Of course, reasoning *in* the direction of a hyperedge is simpler now, since we no longer need to form the quotient of the posterior and prior probability, but can multiply directly with the posterior probability.)

With respect to the class of probability distributions they can represent, the expressive power of Markov networks and Bayesian networks is equivalent, since it is always possible to go from joint probabilities to conditional probabilities and vice versa (provided the necessary marginal probabilities are available). Nevertheless, in some applications Bayesian networks may be preferable, because the additional degree of freedom consisting in the direction of the hyperedges can be used e.g. to express assumed *causal* or *functional* dependences. Indeed, when constructing a Bayesian network by hand, one often starts from a supposed causal model and encodes the causal dependences as probabilistic conditions. This is the reason why Bayesian networks are sometimes also called *probabilistic causal networks*.

However, in this paper we use mostly undirected graphs and joint distributions and we do so for two reasons. In the first place, the direction of an edge can not be justified from the probabilistic model alone. E.g. in the above example it is possible to write the decomposition formulae in several different ways, i.e.

$$\begin{aligned}
\forall i,j,k: P(a_i, b_j, c_k) &= \frac{P(a_i, b_j)P(b_j, c_k)}{P(b_j)} \\
&= P(c_k|b_j)P(b_j|a_i)P(a_i) && A \to B \to C \\
&= P(a_i|b_j)P(b_j|c_k)P(c_k) && A \leftarrow B \leftarrow C \\
&= P(a_i|b_j)P(c_k|b_j)P(b_j) && A \leftarrow B \to C \\
&= P(b_j|a_i)P(b_j|c_k)\frac{P(a_i)P(c_k)}{P(b_j)} && A \to B \leftarrow C
\end{aligned}$$

and represent the decomposition by the corresponding graphs. The direction of an edge always comes from an external source, e.g. from assumptions about causal or functional dependences, and even then, this source may justify directing only some of the edges. Secondly, restricting to joint distributions facilitates the transfer to the possibilistic setting, since it frees us from the need to define what a *conditional* possibility distribution is.

Note, that the interpretation given here to the structure $A \rightarrow B \leftarrow C$ differs from the interpretation that is usually adopted, which is the generally true $\forall i, j, k : P(a_i, b_j, c_k) = P(b_j|a_i, c_k)P(a_i, c_k)$. That is, the conditional independence $A \perp\!\!\!\perp C \mid B$ *need not* hold in this structure. In our interpretation a situation in which A and C are dependent given B would be represented by *one* directed hyperedge connecting A *and* C to B, while *two separate* directed edges indicate conditional independence.

The problem is that the usual interpretation of edge directions owes a lot to causal modeling. As we mentioned above, a Bayesian network is often constructed from a causal model of the universe of discourse. The two separate edges then indicate only that both causes have an influence on the effect: even if one of the causes (e.g. A) is fixed, a change in the other cause (here C) can still change the common effect (here B). That is, the causes independently influence their common effect.

Which interpretation to choose may be a matter of taste, but one should be aware of the fact that the interpretation based on independence of causal influence often contains an implicit assumption. This assumption—which is contained in the *stability assumption* [22]—states that given the value of their common effect causes must be dependent. We will not discuss here whether this assumption is reasonable, but only mention that it is easy to come up with a (mathematical) counterexample[2], and that this assumption is a basic presupposition of the theory of inferred causation [22].

5. Learning Networks from Data

To understand the problem of learning networks from data, i.e. of finding an appropriate decomposition of a multi-dimensional distribution, consider again the relational example of Section 2. We demonstrated that the relation R_{ABC} shown in Fig. 1 can be decomposed into the relations R_{AB} and R_{BC}. It goes without saying that we could not have chosen any pair of two-dimensional subspaces as a decomposition. Intersecting the projections R_{AB} and R_{AC} leads to two, intersecting R_{AC} and R_{BC} to six additional tuples (compared to R_{ABC}). Hence only the pair R_{AB} and R_{BC} forms an exact decomposition.

It is also obvious that there need not be an exact decomposition. Imagine, for example, that the tuple (a_4, b_3, c_2) is not possible. Removing the corre-

[2] Let A, B, and C be variables with $\mathrm{dom}(A) = \mathrm{dom}(B) = \mathrm{dom}(C) = \{0, 1, 2, 3\}$. If $C = (A \bmod 2) + 2(B \bmod 2)$, then A and B independently influence C *and* are independent given C.

sponding cube from Fig. 1 does not change any of the projections, therefore this cube is present in all possible intersections of cylindrical extensions of projections to two-dimensional subspaces. Hence the new relation can not be reconstructed from any projections and thus there is no exact decomposition. In such a situation one either has to work with the relation as a whole or be contented with an approximation that contains some additional tuples. Since the former is often impossible in real world applications because of the high number of dimensions of the underlying domain, a certain loss of information is accepted to make reasoning feasible. Often an approximation has to be accepted even if there is an exact decomposition, because it contains one or more very large hyperedges (connecting a lot of attributes), that can not be dealt with.

Thus the problem of decomposing a relation can be stated in the following way. Given a relation and a maximal size for hyperedges, find an exact decomposition, or, if there is none, the best approximate decomposition of the relation. Unfortunately no direct way to construct such a decomposition has been found yet. Therefore one has to search the space of all possible candidates.

It follows that an algorithm for inducing a decomposition consists always of two parts: an *evaluation measure* and a *search method*. The evaluation measure estimates the quality of a given candidate decomposition (a given hypergraph) and the search method determines which candidates (which hypergraphs) are inspected. Often the search is guided by the value of the evaluation measure, since it is usually the goal to maximize (or to minimize) its value.

A desirable property of an evaluation measure is a certain locality or decomposability, i.e. the possibility to evaluate subgraphs, at best single hyperedges, separately. This is desirable, not only because it facilitates computation, but also because some search methods can make use of such locality. In this paper we only consider local evaluation measures, but global evaluation measures are also available. For example, a simple global evaluation measure for the relational decomposition problem would be the number of additional tuples in the intersection of the cylindrical extensions of the projections [6].

We illustrate the general learning scheme by applying one of the oldest algorithms for decomposing a multi-dimensional probability distribution, that was suggested by Chow and Liu in 1968 [3], to our three-dimensional example. This algorithm can learn only decompositions representable by normal graphs (in which edges connect exactly two nodes) and not hypergraphs (in which edges can connect more than two nodes). But since our example domain is so small, this restriction does not matter.

The idea of Chow and Liu's algorithm is to compute the value of an evaluation measure on all possible edges (two-dimensional subspaces) and use the Kruskal algorithm to determine a maximum or minimum weight spanning tree. For our relational example we could use as an evaluation measure the number of possible value combinations in a subspace relative to the size of this

Table 2. The number of possible combinations relative to the size of the subspace and the gain in Hartley information for three subspaces

subspace	relative number of possible value combinations	gain in Hartley information
$\{A, B\}$	$\frac{6}{3 \cdot 4} = \frac{1}{2} = 50\%$	$\log_2 3 + \log_2 4 - \log_2 6 = 1$
$\{A, C\}$	$\frac{8}{3 \cdot 4} = \frac{2}{3} \approx 67\%$	$\log_2 3 + \log_2 4 - \log_2 8 \approx 0.58$
$\{B, C\}$	$\frac{5}{3 \cdot 3} = \frac{5}{9} \approx 56\%$	$\log_2 3 + \log_2 3 - \log_2 5 \approx 0.85$

subspace (see table 2). Since the overall quality of a decomposition depends on the number of additional tuples in the intersection of the cylindrical extensions of its projections, it is plausible to keep the number of possible value combinations in the cylindrical extensions as small as possible. Obviously, this number depends directly on the number of possible value combinations in the projections. Therefore it seems to be a good heuristic method to select projections in which the ratio of the number of possible value combinations to the size of the subspace is small.

This measure is closely connected to the gain in Hartley information [11] (see table 2), which we will need again in the possibilistic setting. Intuitively, Hartley information measures the average number of questions necessary to determine an element within a given set. It does not take into account the probabilities of the elements and therefore is defined as the binary logarithm of the number of elements in the set. Now consider the task of determining a tuple within a two-dimensional relation. Obviously there are two ways to do this: we can determine the values in the two dimensions (the coordinates) separately, or we can determine the tuple directly. For example, to determine a tuple in the relation R_{AB} (shown in the top right of Fig. 1), we can first determine the value of A ($\log_2 4$ bits) and then the value of B ($\log_2 3$ bits), or we can determine directly the tuple ($\log_2 6$ bits, since there are only six possible tuples). When doing the latter instead of the first, we gain

$$\log_2 4 + \log_2 3 - \log_2 6 = \log_2 \tfrac{3 \cdot 4}{6} = \log_2 2 = 1 \text{ bit}.$$

The above calculation shows that the gain in Hartley information is the binary logarithm of the reciprocal value of the relative number of possible combinations.

If we interpret the values of table 2 as edge weights, we can apply the Kruskal algorithm—to determine a minimum spanning tree for the relative number of possible combinations or a maximum spanning tree for the gain in Hartley information—and thus obtain the graph

$$A — B — C$$

Hence for our example this algorithm finds the exact decomposition.

6. Learning Probabilistic Networks

To apply the ideas of the preceding section to probability distributions we only have to change the evaluation measure. Chow and Liu [3] originally used *mutual information* or *cross entropy* [19] as edge weight. For two variables A and B with domains $\mathrm{dom}(A) = \{a_1, \ldots, a_{r_A}\}$ and $\mathrm{dom}(B) = \{b_1, \ldots, b_{r_B}\}$ it is defined as

$$I_{\mathrm{mut}}(A, B) = \sum_{i=1}^{r_A} \sum_{j=1}^{r_B} P(a_i, b_j) \log_2 \frac{P(a_i, b_j)}{P(a_i)P(b_j)}$$

and can be interpreted in several different ways. One of the simplest interpretations is to see mutual information as a measure of the difference between the joint probability distribution $P(A, B)$ and the distribution $\hat{P}(A, B)$ that can be computed from the marginal distributions $P(A)$ and $P(B)$ under the assumption that A and B are independent, i.e. $\forall i, j : \hat{P}(a_i, b_j) = P(a_i)P(b_j)$.[3] Obviously, the higher the mutual information of two variables, i.e. the more their joint distribution deviates from an independent distribution, the more likely it is that we need their joint distribution to appropriately describe the distribution on the whole domain.

A different interpretation of this measure is connected with the name *information gain* under which it was used in decision tree induction [25]. It is then written differently,

$$\begin{aligned} I_{\mathrm{gain}}(A, B) &= -\sum_{j=1}^{r_B} P(b_j) \log_2 P(b_j) \\ &\quad + \sum_{i=1}^{r_A} P(a_i) \sum_{j=1}^{r_B} P(b_j|a_i) \log_2 P(b_j|a_i) \\ &= H_B - H_{B|A} \\ &= H_A + H_B - H_{AB}, \end{aligned}$$

where H is the Shannon entropy, and thus denotes the *expected reduction in entropy* or, equivalently, the *expected gain in information* about the value of B, if the value of A becomes known. Since mutual information is symmetric, this is also the expected gain in information about the value of A, if the value of B becomes known. Because of its apparent similarity to the gain in Hartley information, it is plausible that the higher the information gain, the more important it is to have the corresponding edge in the network.

Although we do not need this for our example, it should be noted that information gain can easily be extended to more than two attributes:

$$I_{\mathrm{gain}}(A_1, \ldots, A_m) = \sum_{i=1}^{m} H_{A_i} - H_{A_1 \ldots A_m}.$$

[3] It can be shown that I_{mut} is always greater or equal to zero, and equal to zero, if and only if $P(a_i, b_j) = P(a_i)P(b_j)$.

Fig. 6. The mutual information of the three attribute pairs of the probability distribution shown in Fig. 4. On the left is the marginal distribution as calculated from the whole distribution, on the right the independent distribution, i.e. the distribution calculated by multiplying the marginal distributions on the single attribute domains. Mutual information measures the difference of the two.

	a_1	a_2	a_3	a_4
b_3	40	180	20	160
b_2	12	6	120	102
b_1	168	144	30	18

$I_{\text{mut}}(A, B) = 0.43$

a_1	a_2	a_3	a_4	
88	132	68	112	b_3
53	79	41	67	b_2
79	119	61	101	b_1

	a_1	a_2	a_3	a_4
c_3	50	115	35	100
c_2	82	133	99	146
c_1	88	82	36	34

$I_{\text{mut}}(A, C) = 0.05$

a_1	a_2	a_3	a_4	
66	99	51	84	c_3
101	152	78	129	c_2
53	79	41	67	c_1

	b_1	b_2	b_3
c_3	20	180	200
c_2	40	160	40
c_1	180	120	60

$I_{\text{mut}}(B, C) = 0.21$

b_1	b_2	b_3	
96	184	120	c_3
58	110	72	c_2
86	166	108	c_1

Note also, that mutual information or information gain is defined in terms of probabilities. This is no problem, if we are given the probability distribution on the domain of interest, but in practice this distribution is not directly accessible. We are given only a database of sample cases, of which we assume that it was derived from the underlying distribution by a random experiment. In practice we therefore estimate the true probabilities by the empirical probabilities (relative frequencies) found in the database. That is, if n is the total number of tuples in the database and n_i the number of tuples in which attribute A has value a_i, then it is assumed that $P(a_i) = \frac{n_i}{n}$, and the evaluation measure is calculated using this value.

For the example presented in Section 3 we are given the joint probability distribution in Fig. 4. In Fig. 6 mutual information is used to compute the difference of the three possible two-dimensional marginal distributions of this example to the independent distributions calculated from the marginal distributions on the single attribute domains. If we interpret these differences as edge weights, we can again apply the Kruskal algorithm to determine the maximum weight spanning tree. This leads to

$$A - B - C$$

i.e. the graph already used above to represent the possible decomposition of the three-dimensional probability distribution.

A more sophisticated, Bayesian method, the K2 algorithm, was suggested by Cooper and Herskovits in 1992 [4]. It is an algorithm for learning directed graphs and does so by selecting the parents of an attribute.

As an evaluation measure Cooper and Herskovits use the g-function, which is defined as

$$g(A, \mathrm{par}(A)) = c \cdot \prod_{j=1}^{r_{\mathrm{par}(A)}} \frac{(r_A - 1)!}{(n_j + r_A - 1)!} \prod_{i=1}^{r_A} n_{ij}!,$$

where A is an attribute and $\mathrm{par}(A)$ the set of its parents (this is a measure for directed hyperedges). $r_{\mathrm{par}(A)}$ is the number of distinct instantiations (value vectors) of the parent attributes that occur in the database to learn from and r_A the number of values of attribute A. n_{ij} is the number of cases (tuples) in the database in which attribute A has the ith value *and* the parent attributes are instantiated with the jth value vector, n_j the number of cases in which the parent attributes are instantiated with the jth value vector, that is $n_j = \sum_{i=1}^{r_A} n_{ij}$. c is a constant prior probability. If it is assumed that all sets of parents have the same prior probability, it can be neglected, since then only the relation between the values of the evaluation measure for different sets of parent attributes matters.

The g-function estimates (for a certain value of c) the probability of finding the joint distribution of the variable and its parents that is present in the database. That is, assuming that all network structures are equally likely, and that, given a certain structure, all conditional probability distributions compatible with this structure are equally likely, K2 uses Bayesian reasoning to compute the probability of the network structure given the database from the probability of the database given the network structure.

The search method of the K2 algorithm is the following: To narrow the search space and to avoid loops in the resulting hypergraph a topological order of the attributes is defined. A topological order is a concept from graph theory. It describes an order of the nodes of a directed graph, such that: if there is a (hyper)edge from an attribute A (and maybe some others) to attribute B, then A precedes B in the order. Fixing a topological order restricts the permissible graph structures, since the parents of an attribute can only be selected from the attributes preceding it in the order. A topological order can either be stated by a domain expert or derived automatically [29].

The parent attributes are selected using a greedy search. At first the evaluation measure is calculated for the child attribute alone, or—more precisely—for the hyperedge consisting only of the child attribute. Then in turn each of the parent candidates (the attributes preceding the child in the topological order) is temporarily added to the hyperedge and the evaluation measure is computed. The parent candidate yielding the highest value of the evaluation measure is selected as a first parent and permanently added to the hyperedge. In the third step all remaining candidates are added temporarily as a second parent and again the evaluation measure is computed for each of the resulting hyperedges. As before, the parent candidate yielding the highest value is permanently added to the hyperedge. 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 to the hyperedge, yields a value of the evaluation measure exceeding the best value of the preceding step. The resulting hypergraph contains for each attribute a (directed) hyperedge connecting it to its parents (provided parents where added).

Of course, the two algorithms examined are only examples. There are several other search methods (in principle any general heuristic search method is applicable, like hill climbing, simulated annealing, genetic algorithms etc.) and even more evaluation measures (χ^2, information gain ratio, measures based on the minimum description length principle etc.—see [2] for a survey), which we can not consider here.

7. Degrees of Possibility

We now start to transfer the ideas, that up to now were presented in the relational and the probabilistic setting, to the possibilistic setting. Our discussion rests on a specific interpretation of a degree of possibility that is based on the context model [7, 18]. In this model possibility distributions are interpreted as information-compressed representations of (not necessarily nested) random sets, a degree of possibility as the one-point coverage of a random set [21].

More intuitively, a degree of possibility is the least upper bound on the probability of the possibility of a value. We explain this interpretation in three steps. In the first place, the possibility of a value is just what we understand by this term in daily life: whether a value is possible or not. At this point we do not assume intermediate degrees, i.e. if a value is possible, we can not say more than that. We can not give a probability for that value. All we know is that if a value is not possible, its probability must be zero.

Secondly, imagine that we can distinguish between certain disjoint contexts or scenarios, to each of which we can assign a probability and for each of which we can state whether in it the value under consideration is possible or not. Then we can assign to the value as a degree of possibility the sum of the probabilities of the contexts in which it is possible. Thus we arrive at a degree of possibility as the probability of the possibility of a value.

Thirdly, we drop the requirement that the contexts or scenarios must be disjoint. They can overlap, but we assume that we do not know how. This seems to be a sensible assumption, since we should be able to split contexts, if we knew how they overlap. If we now assign to a value as the degree of possibility the sum of the probabilities of the contexts in which it is possible, this value may exceed the actual probability, because of the possible overlap. But since we do not know which contexts overlap and how they overlap, this is the least upper bound consistent with the available information.

Note, that in this interpretation probability distributions are just special possibility distributions. If we have disjoint contexts and if in all contexts in which a value is possible it has the probability 1, the degree of possibility is identical to the probability. Note also, that in this interpretation the degree of possibility can not be less than the probability.

8. Decomposition of Possibility Distributions

The method of decomposing a relation can be transfered to possibility distributions as easily as it could be transfered to probability distributions in Section 3. Again only the definitions of projection, cylindrical extension and intersection have to be modified. Projection now consists in computing the maximal degrees of possibility over the dimensions removed by it. Extension and intersection are combined and consist in calculating the minimum of the prior joint and the posterior marginal possibility degrees.

Determining the maximum or minimum of a number of possibility degrees are the usual methods for reasoning in possibility distributions, but it should be noted that projecting multi-dimensional possibility distributions by determining the maximum over the dimensions removed changes the interpretation of the resulting marginal distribution. Unlike marginal probabilities, which refer only to value vectors over the attributes of the subspace, maximum-projected possibilities still refer to value vectors over all attributes of the universe of discourse. The values of attributes removed by the projection are implicitly fixed but left unknown.

For example, a marginal probability distribution may state: "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 single element vectors. A maximum projection states instead: "The degree of possibility of a value vector with the highest degree of possibility of those value vectors in which attribute A has value a is p." That is, it always refers to a specific value vector over all attributes of the universe of discourse (a specific point), although only the value of the attribute A is known for this vector. The reason is that computing the maximum focuses on a specific vector and on the contexts in which it is possible. But these contexts need not be all in which a value vector with value a for attribute A is possible. Hence the maximum projection of a possibility distribution will in general be less than the actual marginal possibility distribution.

Why all these complications? Basically they are due to the course of history. There are good reasons for using maximum and minimum when working with one-dimensional possibility distributions, especially if the underlying random sets are nested. So it seemed plausible to extend this scheme to multi-dimensional possibility distributions. In addition possibility distributions need not be normalized like probability distributions are. The sum over the degrees of possibility of all elements of the universe of discourse can

Fig. 7. A three-dimensional possibility distribution with maximum projections (maximums over lines/columns). Since in this distribution the equations $\forall i,j,k : \pi(a_i, b_j, c_k) = \min_j(\max_k \pi(a_i, b_j, c_k), \max_i \pi(a_i, b_j, c_k))$ hold, it can be decomposed into the two projections to the subspaces $\{A, B\}$ and $\{B, C\}$.

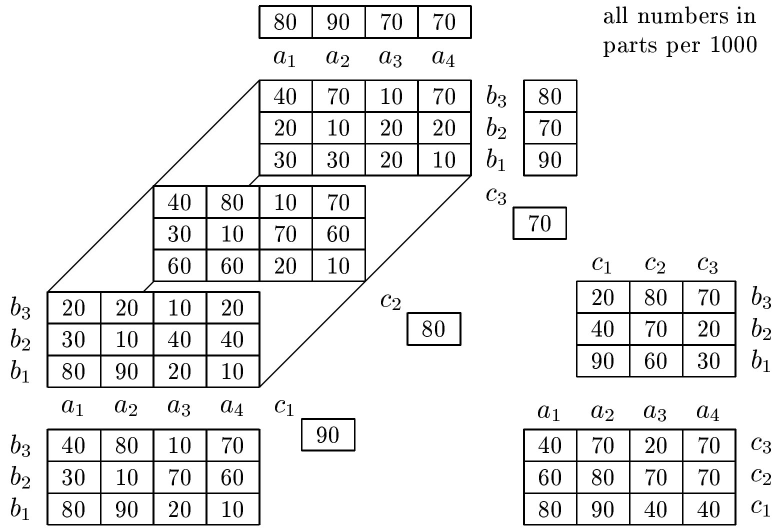
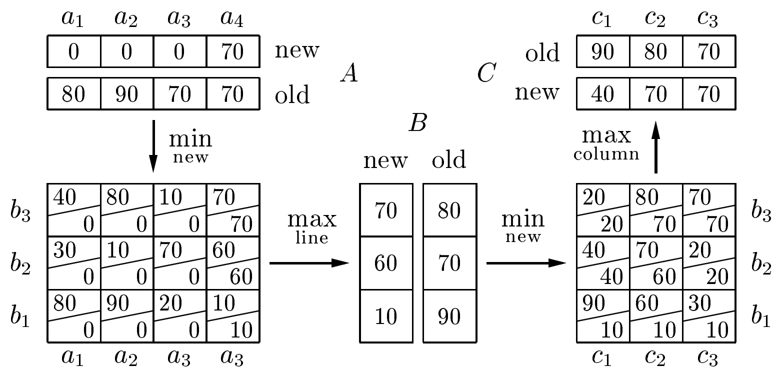


Fig. 8. Propagation of the evidence that attribute A has value a_4 in the three-dimensional possibility distribution shown in Fig. 7 using the projections to the subspaces $\{A, B\}$ and $\{B, C\}$



exceed one. Therefore, at first sight, sum projection seems to be inapplicable. But a closer look reveals that it can be used as well (though with some precaution). Nevertheless we will not discuss this possibility here. Rather we proceed by illustrating the decomposition of possibility distributions with a simple example.

Fig. 7 shows a three-dimensional possibility distribution on the joint domain of the attributes A, B, and C and its maximum projections. Since the equations $\forall i,j,k : \pi(a_i,b_j,c_k) = \min_j(\max_k \pi(a_i,b_j,c_k), \max_i \pi(a_i,b_j,c_k))$ hold in this distribution, it can be decomposed into projections to the subspaces $\{A,B\}$ and $\{B,C\}$. Therefore it is possible to propagate the observation that attribute A has value a_4 using the scheme shown in Fig. 8. Again the results obtained are the same as those that can be computed directly from the three-dimensional distribution.

9. Learning Possibilistic Networks

Learning possibilistic networks follows the same scheme as learning probabilistic ones: we need an evaluation measure and a search method. Since the search method is fairly independent of the underlying uncertainty calculus, we can use the same methods as for learning probabilistic networks. Hence we only have to look for appropriate evaluation measures.

One evaluation measure can be derived from the U-uncertainty measure of *nonspecificity* of a possibility distribution [15], which is defined as

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

and can be justified as a generalization of Hartley information [11] to the possibilistic setting [14]. $\mathrm{nsp}(\pi)$ reflects the expected amount of information (measured in bits) that has to be added in order to identify the actual value within the set $[\pi]_\alpha$ of alternatives, assuming a uniform distribution on the set $[0, \sup(\pi)]$ of possibilistic confidence levels α [10].

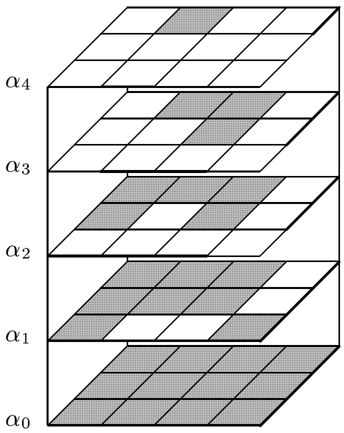
The role nonspecificity plays in possibility theory is similar to that of Shannon entropy in probability theory. Thus the idea suggests itself to construct an evaluation measure from nonspecificity in the same way as mutual information or information gain is constructed from Shannon entropy. By analogy we define *specificity gain* for two variables A and B as

$$S_{\text{gain}} = \mathrm{nsp}\left(\max_B(\pi_{AB})\right) + \mathrm{nsp}\left(\max_A(\pi_{AB})\right) - \mathrm{nsp}(\pi_{AB}),$$

or for more than two variables

$$S_{\text{gain}} = \sum_{k=1}^{m} \mathrm{nsp}\left(\max_X(\pi_{A_1 \ldots A_m})\right) - \mathrm{nsp}(\pi_{A_1 \ldots A_m}),$$

Fig. 9. Illustration of the idea of specificity gain. A two-dimensional possibility distribution is seen as a set of relational cases, one for each α-level. In each relational case, determining the allowed coordinates is compared to determining directly the allowed value pairs. Specificity gain aggregates the gain in Hartley information that can be achieved on each α-level by computing the integral over all α-levels.



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

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

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

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

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

Fig. 10. The specificity gain of the three attribute pairs of the possibility distribution shown in Fig. 7. On the left is the maximum projection as calculated from the whole distribution, on the right the independent distribution, i.e. the distribution calculated as the minimum of the maximum projections to the single variable domains. Specificity gain measures the difference of the two.

	a_1	a_2	a_3	a_4
b_3	40	80	10	70
b_2	30	10	70	60
b_1	80	90	20	10

$S_{\text{gain}}(A, B) = 0.055$

a_1	a_2	a_3	a_4	
80	80	70	70	b_3
70	70	70	70	b_2
80	90	70	70	b_1

	a_1	a_2	a_3	a_4
c_3	40	70	20	70
c_2	60	80	70	70
c_1	80	90	40	40

$S_{\text{gain}}(A, C) = 0.026$

a_1	a_2	a_3	a_4	
70	70	70	70	c_3
80	80	70	70	c_2
80	90	70	70	c_1

	b_1	b_2	b_3
c_3	20	80	70
c_2	40	70	20
c_1	90	60	30

$S_{\text{gain}}(B, C) = 0.048$

b_1	b_2	b_3	
70	70	70	c_3
80	70	80	c_2
90	70	80	c_1

where $X = \{A_i \mid 1 \leq i \leq m, i \neq k\}$. This measure is equivalent to the one suggested in [10].

The idea of specificity gain is illustrated in Fig. 9. A joint possibility distribution is seen as a set of relational cases, one for each α-level. Specificity gain aggregates the gain in Hartley information for these relational cases by computing the integral over all α-levels.

In analogy to information gain it is also possible to interpret specificity gain as a difference measure. For example, for two attributes A and B the possibility distribution π_{AB} is compared to $\hat{\pi}_{AB}$, which is defined as $\forall i, j: \hat{\pi}_{AB}(a_i, b_j) = \min(\max_j \pi_{AB}(a_i, b_j), \max_i \pi_{AB}(a_i, b_j))$. Since it is easy to show that $\mathrm{nsp}(\hat{\pi}_{AB}) = \mathrm{nsp}(\max_B(\pi_{AB})) + \mathrm{nsp}(\max_A(\pi_{AB}))$, it follows that $S_{\text{gain}} = \mathrm{nsp}(\hat{\pi}_{AB}) - \mathrm{nsp}(\pi_{AB})$.

In the same way as information gain is used in Fig. 6, specificity gain is used in Fig. fig.spcgain to compute the difference of the three possible two-dimensional maximum projections of the example shown in Fig. 7 to the distribution calculated from the maximum projections to the single variable domains. If we interpret these differences as edge weights we can apply the Kruskal algorithm to determine the maximum weight spanning tree. This leads to

$$A — B — C$$

Hence the exact decomposition is found.

Just as for learning decompositions of probability distributions this measure is only an example. There are also some other measures available (specificity gain ratio, a variant of the χ^2-measure etc., see [2]), which we can not discuss in detail here.

10. Summary

As we hope to have shown in this paper, the ideas underlying decomposition of probability as well as possibility distributions are very simple. Since decomposition reduces the amount of storage needed, but does not restrict reasoning (at least if the distribution can be reconstructed exactly from the decomposition), it is a valuable technique for expert system development. The methods available for inducing decompositions are also fairly simple, as was demonstrated by the examples of Sections 6 and 9. All of them consist of an evaluation measure and a search method, for both of which there are several alternatives. Although most decomposition methods are heuristic in nature, they lead to good results in practice. Nevertheless there is a large potential for future refinements.

References

1. S.K. Andersen, K.G. Olesen, F.V. Jensen, and F. Jensen. HUGIN — A shell for building Bayesian belief universes for expert systems. *Proc. 11th Int. J. Conf. on Artificial Intelligence*, 1080–1085, 1989
2. C. Borgelt and R. Kruse. Evaluation Measures for Learning Probabilistic and Possibilistic Networks. Proc. 6rh IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97), Vol. 2, 669–676, Barcelona, Spain, 1997.
3. C.K. Chow and C.N. Liu. Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Trans. on Information Theory* 14(3):462–467, IEEE 1968
4. G.F. Cooper and E. Herskovits. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning* 9:309–347, Kluwer 1992
5. C.J. Date. *An Introduction to Database Systems*, Vol. 1. Addison Wesley, Reading, MA, 1986
6. R. Dechter. Decomposing a Relation into a Tree of Binary Relations. *Journal of Computer and System Sciences* 41:2–24, 1990
7. J. Gebhardt and R. Kruse. A Possibilistic Interpretation of Fuzzy Sets in the Context Model. *Proc. IEEE Int. Conf. on Fuzzy Systems*, 1089-1096, San Diego, CA, 1992.
8. J. Gebhardt and R. Kruse. POSSINFER — A Software Tool for Possibilistic Inference. In: D. Dubois, H. Prade, and R. Yager, eds. *Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications*, Wiley 1995
9. J. Gebhardt and R. Kruse. Learning Possibilistic Networks from Data. *Proc. 5th Int. Workshop on Artificial Intelligence and Statistics*, 233–244, Fort Lauderdale, FL, 1995
10. J. Gebhardt and R. Kruse. Tightest Hypertree Decompositions of Multivariate Possibility Distributions. *Proc. Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems*, 1996
11. R.V.L. Hartley. Transmission of Information. *The Bell Systems Technical Journal* 7:535–563, 1928
12. D. Heckerman. *Probabilistic Similarity Networks.* MIT Press, Cambridge, MA, 1991
13. D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning* 20:197–243, Kluwer 1995
14. M. Higashi and G.J. Klir. Measures of Uncertainty and Information based on Possibility Distributions. *Int. Journal of General Systems* 9:43–58, 1982
15. G.J. Klir and M. Mariano. On the Uniqueness of a Possibility Measure of Uncertainty and Information. *Fuzzy Sets and Systems* 24:141–160, 1987
16. R. Kruse and E. Schwecke. Fuzzy Reasoning in a Multidimensional Space of Hypotheses. *Int. Journal of Approximate Reasoning* 4:47–68, 1990

17. R. Kruse, E. Schwecke, and J. Heinsohn. *Uncertainty and Vagueness in Knowledge-based Systems: Numerical Methods.* Series: Artificial Intelligence, Springer, Berlin 1991
18. R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*, John Wiley & Sons, Chichester, England 1994
19. S. Kullback and R.A. Leibler. On Information and Sufficiency. *Ann. Math. Statistics* 22:79–86, 1951
20. S.L. Lauritzen and D.J. Spiegelhalter. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *Journal of the Royal Statistical Society, Series B*, 2(50):157–224, 1988
21. H.T. Nguyen. Using Random Sets. *Information Science* 34:265–274, 1984
22. J. Pearl and T.S. Verma. A Theory of Inferred Causation. In: J.A. Allen, R. Fikes, and E. Sandewall, eds. *Proc. 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning*, Morgan Kaufman, San Mateo, CA, 1991
23. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (2nd edition).* Morgan Kaufman, San Mateo, CA, 1992
24. J.R. Quinlan. Induction of Decision Trees. *Machine Learning* 1:81–106, 1986
25. J.R. Quinlan. *C4.5: Programs for Machine Learning*, Morgan Kaufman, San Mateo, CA, 1993
26. A. Saffiotti and E. Umkehrer. PULCINELLA: A General Tool for Propagating Uncertainty in Valuation Networks. *Proc. 7th Conf. on Uncertainty in AI*, 323–331, San Mateo, CA, 1991
27. G. Shafer and P.P. Shenoy. Local Computations in Hypertrees. Working Paper 201, School of Business, University of Kansas, Lawrence, KS, 1988
28. C.E. Shannon. The Mathematical Theory of Communication. *The Bell Systems Technical Journal* 27:379–423, 1948
29. M. Singh and M. Valtorta. An Algorithm for the Construction of Bayesian Network Structures from Data. *Proc. 9th Conf. on Uncertainty in AI*, 259–265, Morgan Kaufman, San Mateo, CA, 1993
30. J.D. Ullman. *Principles of Database and Knowledge-Base Systems*, Vol. 1 and 2. Computer Sciences Press, Rockville, MD, 1988/1989

This article was processed using the LaTeX macro package with LLNCS style