Reasoning with Uncertainty

7

We have already shown in Chap. 4 with the Tweety problem that two-value logic leads to problems in everyday reasoning. In this example, the statements *Tweety is a penguin*, *Penguins are birds*, and *All birds can fly* lead to the (semantically incorrect) inference *Tweety can fly*. Probability theory provides a language in which we can formalize the statement *Nearly all birds can fly* and carry out inferences on it. Probability theory is a proven method we can use here because the uncertainty about whether birds can fly can be modeled well by a probability value. We will show, that statements such as *99% of all birds can fly*, together with probabilistic logic, lead to correct inferences.

Reasoning under uncertainty with limited resources plays a big role in everyday situations and also in many technical applications of AI. In these areas heuristic processes are very important, as we have already discussed in Chap. 6. For example, we use heuristic techniques when looking for a parking space in city traffic. Heuristics alone are often not enough, especially when a quick decision is needed given incomplete knowledge, as shown in the following example. A pedestrian crosses the street and an auto quickly approaches. To prevent a serious accident, the pedestrian must react quickly. He is not capable of worrying about complete information about the state of the world, which he would need for the search algorithms discussed in Chap. 6. He must therefore come to an optimal decision under the given constraints (little time and little, potentially uncertain knowledge). If he thinks too long, it will be dangerous. In this and many similar situations (see Fig. 7.1 on page 126), a method for reasoning with uncertain and incomplete knowledge is needed.

We want to investigate the various possibilities of reasoning under uncertainty in a simple medical diagnosis example. If a patient experiences pain in the right lower abdomen and a raised white blood cell (leukocyte) count, this raises the suspicion that it might be appendicitis. We model this relationship using propositional logic with the formula

$$\textit{Stomach pain right lower} \wedge \textit{Leukocytes} > 10000 \rightarrow \textit{Appendicitis}$$

W. Ertel, *Introduction to Artificial Intelligence*, Undergraduate Topics in Computer Science, DOI 10.1007/978-3-319-58487-4_7

Fig. 7.1 "Let's just sit back and think about what to do!"

If we then know that

$$Stomach\ pain\ right\ lower \wedge Leukocytes > 10000$$

is true, then we can use modus ponens to derive *Appendicitis*. This model is clearly too coarse. In 1976, Shortliffe and Buchanan recognized this when building their medical expert system MYCIN [Sho76]. They developed a calculus using so-called certainty factors, which allowed the certainty of facts and rules to be represented. A rule $A \rightarrow B$ is assigned a certainty factor β. The semantic of a rule $A \rightarrow_{\beta} B$ is defined via the conditional probability $P(B \mid A) = \beta$. In the above example, the rule could then read

$$Stomach\ pain\ right\ lower \wedge Leukocytes > 10000 \rightarrow_{0.6} Appendicitis.$$

For reasoning with this kind of formulas, they used a calculus for connecting the factors of rules. It turned out, however, that with this calculus inconsistent results could be derived.

As discussed in Chap. 4, there were also attempts to solve this problem by using non-monotonic logic and default logic, which, however, were unsuccessful in the end. The Dempster–Schäfer theory assigns a belief function *Bel*(*A*) to a logical proposition *A*, whose value gives the degree of evidence for the truth of *A*. But even this formalism has weaknesses, which is shown in [Pea88] using a variant of the Tweety example. Even fuzzy logic, which above all is successful in control theory, demonstrates considerable weaknesses when reasoning under uncertainty in more complex applications [Elk93].

Since about the mid-1980s, probability theory has had more and more influence in AI [Pea88, Che85, Whi96, Jen01]. In the field of reasoning with Bayesian networks, or subjective probability, it has secured itself a firm place among successful AI techniques. Rather than implication as it is known in logic (material implication), conditional probability is used here, which models everyday causal reasoning significantly better. Reasoning with probability profits heavily from the fact that probability theory is a hundreds of years old, well-established branch of mathematics.

In this chapter we will select an elegant, but for an instruction book somewhat unusual, entry point into this field. After a short introduction to the most important foundations needed here for reasoning with probability, we will begin with a simple, but important example for reasoning with uncertain and incomplete knowledge. In a quite natural, almost compelling way, we will be led to the method of maximum entropy (MaxEnt). Then we will show the usefulness of this method in practice using the medical expert system LEXMED. Finally we will introduce the now widespread reasoning with Bayesian networks, and show the relationship between the two methods.

7.1 Computing with Probabilities

The reader who is familiar with probability theory can skip this section. For everyone else we will give a quick ramp-up and recommend a few appropriate textbooks such as [Ros09, FPP07].

Probability is especially well-suited for modeling reasoning under uncertainty. One reason for this is that probabilities are intuitively easy to interpret, which can be seen in the following elementary example.

Example 7.1 For a single roll of a gaming die (experiment), the probability of the event "rolling a six" equals 1/6, whereas the probability of the occurrence "rolling an odd number" is equal to 1/2.

Definition 7.1 Let Ω be the finite set of *events* for an experiment. Each event $\omega \in \Omega$ represents a possible outcome of the experiment. If these events $w_i \in \Omega$ mutually exclude each other, but cover all possible outcomes of the attempt, then they are called *elementary events*.

Example 7.2 For a single roll of one gaming die

$$\Omega = \{1, 2, 3, 4, 5, 6\}$$

because no two of these events can happen simultaneously. Rolling an even number ($\{2, 4, 6\}$) is therefore not an elementary event, nor is rolling a number smaller than five ($\{1, 2, 3, 4\}$) because $\{2, 4, 6\} \cap \{1, 2, 3, 4\} = \{2, 4\} \neq \emptyset$.

Given two events A and B, $A \cup B$ is also an event. Ω itself is denoted the *certain event*, and the empty set $\emptyset$ the *impossible event*.

In the following we will use the propositional logic notation for set operations. That is, for the set $A \cap B$ we write $A \wedge B$. This is not only a syntactic transformation, rather it is also semantically correct because the intersection of two sets is defined as

$$x \in A \cap B \Leftrightarrow x \in A \wedge x \in B.$$

Because this is the semantic of $A \wedge B$, we can and will use this notation. This is also true for the other set operations union and complement, and we will, as shown in the following table, use the propositional logic notation for them as well.

Set notation	Propositional logic	Description
$A \cap B$	$A \wedge B$	intersection / and
$A \cup B$	$A \vee B$	union / or
$\bar{A}$	$\neg A$	complement / negation
Ω	t	certain event / true
$\emptyset$	f	impossible event / false

The variables used here (for example A, B, etc.) are called *random variables* in probability theory. We will only use discrete chance variables with finite domains here. The variable *face_number* for a dice roll is discrete with the values 1, 2, 3, 4, 5, 6. The probability of rolling a five or a six is equal to 1/3. This can be described by

$$P(face\_number \in \{5, 6\}) = P(face\_number = 5 \vee face\_number = 6) = 1/3.$$

The concept of probability is supposed to give us a description as objective as possible of our "belief" or "conviction" about the outcome of an experiment. All numbers in the interval [0,1] should be possible, where 0 is the probability of the impossible event and 1 the probability of the certain event. We come to this from the following definition.

Definition 7.2 Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ be finite. There is no preferred elementary event, which means that we assume a symmetry related to the frequency of how often each elementary event appears. The *probability* $P(A)$ of the event A is then

$$P(A) = \frac{|A|}{|\Omega|} = \frac{\text{Number of favorable cases for } A}{\text{Number of possible cases}}.$$

It follows immediately that every elementary event has the probability $1/|\Omega|$. The requirement that elementary events have equal probability is called the *Laplace assumption* and the probabilities calculated thereby are called *Laplace probabilities*. This definition hits its limit when the number of elementary events becomes infinite. Because we are only looking at finite event spaces here, though, this does not present a problem. To describe events we use variables with the appropriate number of values. For example, a variable *eye_color* can take on the values *green*, *blue*, *brown*. $eye\_color = blue$ then describes an event because we are dealing with a proposition with the truth values *t* or *f*. For binary (boolean) variables, the variable itself is already a proposition. Here it is enough, for example, to write $P(JohnCalls)$ instead of $P(JohnCalls = t)$.

Example 7.3 By this definition, the probability of rolling an even number is

$$P(face\_number \in \{2,4,6\}) = \frac{|\{2,4,6\}|}{|\{1,2,3,4,5,6\}|} = \frac{3}{6} = \frac{1}{2}.$$

The following important rules follow directly from the definition.

Theorem 7.1

1. $P(\Omega) = 1$.
2. $P(\emptyset) = 0$, *which means that the impossible event has a probability of* 0.
3. *For pairwise exclusive events A and B it is true that* $P(A \vee B) = P(A) + P(B)$.
4. *For two complementary events A and* $\neg A$ *it is true that* $P(A) + P(\neg A) = 1$.
5. *For arbitrary events A and B it is true that* $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$.
6. *For* $A \subseteq B$ *it is true that* $P(A) \leq P(B)$.
7. *If* $A_1, \ldots, A_n$ *are the elementary events, then* $\sum_{i=1}^{n} P(A_i) = 1$ (*normalization condition*).

The expression $P(A \wedge B)$ or equivalently $P(A, B)$ stands for the probability of the events $A \wedge B$. We are often interested in the probabilities of all elementary events, that is, of all combinations of all values of the variables A and B. For the binary variables A and B these are $P(A, B)$, $P(A,\neg B)$, $P(\neg A, B)$, $P(\neg A, \neg B)$. We call the vector

$$(P(A,\ B), P(A,\ \neg B), P(\neg A,\ B), P(\neg A,\ \neg B))$$

consisting of these four values a *distribution* or *joint probability distribution* of the variables A and B. A shorthand for this is $\boldsymbol{P}(A, B)$. The distribution in the case of two variables can be nicely visualized in the form of a table (matrix), represented as follows:

$\boldsymbol{P}(A, B)$	$B = w$	$B = f$
$A = w$	$P(A, B)$	$P(A, \neg B)$
$A = f$	$P(\neg A, B)$	$P(\neg A, \neg B)$

For the d variables $X_1, \ldots, X_d$ with n values each, the distribution has the values $P(X_1 = x_1, \ldots, X_d = x_d)$ and $x_1, \ldots, x_d$, each of which take on n different values. The distribution can therefore be represented as a d-dimensional matrix with a total of n^d elements. Due to the normalization condition from Theorem 7.1 on page 129, however, one of these n^d values is redundant and the distribution is characterized by $n^d - 1$ unique values.

7.1.1 Conditional Probability

Example 7.4 On Landsdowne street in Boston, the speed of 100 vehicles is measured. For each measurement it is also noted whether the driver is a student. The results are

Event	Frequency	Relative frequency
Vehicle observed	100	1
Driver is a student (S)	30	0.3
Speed too high (G)	10	0.1
Driver is a student and speeding ($S \wedge G$)	5	0.05

We pose the question: *Do students speed more frequently than the average person, or than non-students?*[1]

[1]The computed probabilities can only be used for continued propositions if the measured sample (100 vehicles) is representative. Otherwise only propositions about the observed 100 vehicles can be made.

The answer is given by the probability

$$P(G|S) = \frac{|\text{Driver is a student and speeding}|}{|\text{Driver is a student}|} = \frac{5}{30} = \frac{1}{6} \approx 0.17$$

for speeding under the condition that the driver is a student. This is obviously different from the a priori probability $P(G) = 0.1$ for speeding. For the a priori probability, the event space is not limited by additional conditions.

Definition 7.3 For two events A and B, the probability $P(A|B)$ for A under the condition B (*conditional probability*) is defined by

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}.$$

In Example 7.4 we see that in the case of a finite event space, the conditional probability $P(A|B)$ can be understood as the probability of $A \wedge B$ when we only look at the event B, that is, as

$$P(A|B) = \frac{|A \wedge B|}{|B|}.$$

This formula can be easily derived using Definition 7.2 on page 129

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{\frac{|A \wedge B|}{|\Omega|}}{\frac{|B|}{|\Omega|}} = \frac{|A \wedge B|}{|B|}.$$

Definition 7.4 If, for two events A and B,

$$P(A|B) = P(A),$$

then these events are called independent.

Thus A and B are independent if the probability of the event A is not influenced by the event B.

Theorem 7.2 *For independent events A and B, it follows from the definition that*

$$P(A \wedge B) = P(A) \cdot P(B).$$

Example 7.5 For a roll of two dice, the probability of rolling two sixes is 1/36 if the two dice are independent because

$$P(D_1 = 6 \wedge D_2 = 6) = P(D_1 = 6) \cdot P(D_2 = 6) = \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36},$$

where the first equation is only true when the two dice are independent. If for example by some magic power die 2 is always the same as die 1, then

$$P(D_1 = 6 \wedge D_2 = 6) = \frac{1}{6}.$$

Chain Rule

Solving the definition of conditional probability for $P(A \wedge B)$ results in the so-called product rule

$$P(A \wedge B) = P(A|B)\, P(B),$$

which we immediately generalize for the case of n variables. By repeated application of the above rule we obtain the *chain rule*

$$\begin{aligned}
&\boldsymbol{P}(X_1, \ldots, X_n) \\
&\quad = \boldsymbol{P}(X_n|X_1, \ldots, X_{n-1}) \cdot \boldsymbol{P}(X_1, \ldots, X_{n-1}) \\
&\quad = \boldsymbol{P}(X_n|X_1, \ldots, X_{n-1}) \cdot \boldsymbol{P}(X_{n-1}|X_1, \ldots, X_{n-2}) \cdot \boldsymbol{P}(X_1, \ldots, X_{n-2}) \\
&\quad = \boldsymbol{P}(X_n|X_1, \ldots, X_{n-1}) \cdot \boldsymbol{P}(X_{n-1}|X_1, \ldots, X_{n-2}) \cdot \ldots \cdot \boldsymbol{P}(X_n|X_1) \cdot \boldsymbol{P}(X_1) \\
&\quad = \prod_{i=1}^{n} \boldsymbol{P}(X_n|X_1, \ldots, X_{i-1}),
\end{aligned} \tag{7.1}$$

with which we can represent a distribution as a product of conditional probabilities. Because the chain rule holds for all values of the variables $X_1, \ldots, X_n$, it has been formulated for the distribution using the symbol $\boldsymbol{P}$.

Marginalization

Because $A \Leftrightarrow (A \wedge B) \vee (A \wedge \neg B)$ is true for binary variables A and B

$$P(A) = P((A \wedge B) \vee (A \wedge \neg B)) = P(A \wedge B) + P(A \wedge \neg B).$$

By summation over the two values of B, the variable B is eliminated. Analogously, for arbitrary variables $X_1, \ldots, X_d$, a variable, for example X_d, can be eliminated by summation over all of their variables and we get

$$P(X_1 = x_1, \ldots, X_{d-1} = x_{d-1}) = \sum_{x_d} P(X_1 = x_1, \ldots, X_{d-1} = x_{d-1}, X_d = x_d).$$

The application of this formula is called marginalization. This summation can continue with the variables $X_1, \ldots, X_{d-1}$ until just one variable is left. Marginalization can also be applied to the distribution $\boldsymbol{P}(X_1, \ldots, X_d)$. The resulting distribution $\boldsymbol{P}(X_1, \ldots, X_{d-1})$ is called the *marginal distribution*. It is comparable to the projection of a rectangular cuboid on a flat surface. Here the three-dimensional object is drawn on the edge or "margin" of the cuboid, i.e. on a two-dimensional set. In both cases the dimensionality is reduced by one.

Example 7.6 We observe the set of all patients who come to the doctor with acute stomach pain. For each patient the leukocyte value is measured, which is a metric for the relative abundance of white blood cells in the blood. We define the variable *Leuko*, which is true if and only if the leukocyte value is greater than 10,000. This indicates an infection in the body. Otherwise we define the variable *App*, which tells us whether the patient has appendicitis, that is, an infected appendix. The distribution $\boldsymbol{P}(App, Leuko)$ of these two variables is given in the following table:

$\boldsymbol{P}(App, Leuko)$	App	$\neg App$	Total
$Leuko$	0.23	0.31	0.54
$\neg Leuko$	0.05	0.41	0.46
Total	0.28	0.72	1

In the last row the sum over the rows is given, and in the last column the sum of the columns is given. These sums are arrived at by marginalization. For example, we read off

$$P(Leuko) = P(App, Leuko) + P(\neg App, Leuko) = 0.54.$$

The given distribution $\boldsymbol{P}(App, Leuko)$ could come from a survey of German doctors, for example. From it we can then calculate the conditional probability

$$P(Leuko|App) = \frac{P(Leuko, App)}{P(App)} = 0.82$$

which tells us that about 82% of all appendicitis cases lead to a high leukocyte value. Values like this are published in medical literature. However, the conditional

probability $P(App|Leuko)$, which would actually be much more helpful for diagnosing appendicitis, is not published. To understand this, we will first derive a simple, but very important formula.

Bayes' Theorem
Swapping A and B in Definition 7.3 yields

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} \quad \text{and} \quad P(B|A) = \frac{P(A \wedge B)}{P(A)}.$$

By solving both equations for $P(A \wedge B)$ and equating them we obtain **Bayes' theorem**

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}, \tag{7.2}$$

whose relevance to many applications we will illustrate using three examples. First we apply it to the appendicitis example and obtain

Example 7.7

$$P(App|Leuko) = \frac{P(Leuko|App) \cdot P(App)}{P(Leuko)} = \frac{0.82 \cdot 0.28}{0.54} = 0.43. \tag{7.3}$$

Why then is $P(Leuko|App)$ published, but not $P(App|Leuko)$?

Assuming that appendicitis affects the biology of all humans the same, regardless of ethnicity, $P(Leuko|App)$ is a universal value that is valid worldwide. In Equation 7.3 we see that $P(App|Leuko)$ is not universal, for this value is influenced by the *a priori* probabilities $P(App)$ and $P(Leuko)$. Each of these can vary according to one's life circumstances. For example, $P(Leuko)$ is dependent on whether a population has a high or low rate of exposure to infectious diseases. In the tropics, this value can differ significantly from that of cold regions. Bayes' theorem, however, makes it easy for us to take the universally valid value $P(Leuko|App)$, and compute $P(App|Leuko)$ which is useful for diagnosis.

Before we dive deeper into this example and build a medical expert system for appendicitis in Sect. 7.3 let us first apply Bayes' theorem to another interesting medical example.

Example 7.8 In cancer diagnosis, so-called tumor markers are often measured. One example of this is the use of the tumor marker PSA (prostate specific antigen) for the diagnosis of prostate cancer (PCa = prostate cancer) in men. Assuming that no further tests for PCa have been conducted, the test is considered positive, that is, there is suspected PCa, if the concentration of PSA reaches a level at or above 4 *ng/ml*. If this occurs, the probability $P(C|pos)$ of PCa is of interest to the patient.

The binary variable C is true if the patient has PCa, and *pos* represents a PSA value $\geq 4\ ng/ml$. Let us now compute the $P(C|pos)$. For reasons similar to those mentioned for appendicitis diagnosis, this value is not reported. Instead, researchers publish the sensitivity $P(pos|C)$ and the specificity $P(neg|\neg C)$ of the test.[2] According to [HL04], for a sensitivity of 0.95, the specificity can be at most 0.25, which is why we proceed from $P(pos|C) = 0.95$ and $P(neg|\neg C) = 0.25$ below. We apply Bayes' theorem and obtain

$$\begin{aligned} P(C|pos) &= \frac{P(pos|C) \cdot P(C)}{P(pos)} = \frac{P(pos|C) \cdot P(C)}{P(pos|C) \cdot P(C) + P(pos|\neg C) \cdot P(\neg C)} \\ &= \frac{0.95 \cdot 0.0021}{0.95 \cdot 0.0021 + \mathbf{0.75} \cdot 0.99679} = \frac{0.95 \cdot 0.0021}{0.75} = 0.0027. \end{aligned}$$

Here we use $P(pos|\neg C) = 1 - P(neg|\neg C) = 1 - 0.25 = 0.75$ and $P(C) = 0.0021 = 0.21\%$ as the *a priori* probability of PCa during one year.[3] It makes sense to assume that the PSA test is done once per year. This result is somewhat surprising from the patient's perspective because the probability of PCa after a positive test is, at 0.27%, only marginally higher than the probability of 0.21% for PCa for a 55-year-old man. Thus, a PSA value of just over 4 *ng/ml* is definitively no reason for the patient to panic. At most it is used as a basis for further examinations, such as biopsy or MRI, leading if necessary to radiation and surgery. The situation is similar for many other tumor markers such as those for colorectal cancer or breast cancer diagnosis by mammography.

The cause of this problem is the very low specificity $P(neg|\neg C) = 0.25$, which leads to 75% of healthy patients (without PCa) getting a false-positive test result and consequently undergoing unnecessary examinations. Because of this, PSA testing has been a controversial discussion topic for years.[4]

Assume we had a better test with a specificity of 99%, which would only deliver a false-positive result for one percent of healthy men. Then, in the above calculation, we would assign $P(pos|\neg C)$ the value 0.01 and obtain the result $P(C|pos) = 0,17$. Plainly, this test would be much more specific.

Example 7.9 A sales representative who wants to sell an alarm system could make the following argument:

> *If you buy this very reliable alarm system, it will alert you to any break-in with 99% certainty. Our competitor's system only offers a certainty of 85%.*

Hearing this, if the buyer concludes that from an alert A he can infer a break-in B with high certainty, he is wrong. Bayes' theorem shows the reason. What the

[2] For definitions of sensitivity and specificity see Eqs. 7.16 and 7.17.

[3] See http://www.prostata.de/pca_haeufigkeit.html for a 55-year-old man.

[4] The author is not a medical doctor. Therefore these computations should not be used as a basis for personal medical decisions by potentially afflicted individuals. If necessary, please consult a specialist physician or the relevant specialist literature.

representative told us is that $P(A|B) = 0.99$. What he doesn't say, however, is what it means when we hear the alarm go off. To find out, we use Bayes' theorem to compute $P(B|A)$ and assume that the buyer lives in a relatively safe area in which break-ins are rare, with $P(B) = 0.001$. Additionally, we assume that the alarm system is triggered not only by burglars, but also by animals, such as birds or cats in the yard, which results in $P(A) = 0.1$. Thus we obtain

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{0.99 \cdot 0.001}{0.1} = 0.01,$$

which means that whoever buys this system will not be happy because they will be startled by too many false alarms. When we examine the denominator

$$P(A) = P(A|B)\,P(B) + P(A|\neg B)\,P(\neg B) = 0.00099 + P(A|\neg B) \cdot 0.999 = 0.1$$

of Bayes' theorem more closely, we see that $P(A|\neg B) \approx 0.1$, which means that the alarm will be triggered roughly every tenth day that there is not a break-in.

From this example we learn, among other things, that it is important to consider which probabilities we are really interested in as a buyer, expecially when it comes to security. When the arguments of a conditional probability are interchanged, the value can change dramatically when the *prior* probabilities differ significantly.

7.2 The Principle of Maximum Entropy

We will now show, using an inference example, that a calculus for reasoning under uncertainty can be realized using probability theory. However, we will soon see that the well-worn probabilistic paths quickly come to an end. Specifically, when too little knowledge is available to solve the necessary equations, new ideas are needed. The American physicist E.T. Jaynes did pioneering work in this area in the 1950s. He claimed that given missing knowledge, one can maximize the entropy of the desired probability distribution, and applied this principle to many examples in [Jay57, Jay03]. This principle was then further developed [Che83, Nil86, Kan89, KK92] and is now mature and can be applied technologically, which we will show in the example of the LEXMED project in Sect. 7.3.

7.2.1 An Inference Rule for Probabilities

We want to derive an inference rule for uncertain knowledge that is analogous to the modus ponens. From the knowledge of a proposition A and a rule $A \Rightarrow B$, the conclusion B shall be reached. Formulated succinctly, this reads

$$\frac{A,\ A \to B}{B}.$$

The generalization for probability rules yields

$$\frac{P(A) = \alpha, \quad P(B|A) = \beta}{P(B) = ?}.$$

Let the two probability rules α, β be given and the value $P(B)$ desired. By marginalization we obtain the desired marginal distribution

$$P(B) = P(A, B) + P(\neg A, B) = P(B|A) \cdot P(A) + P(B|\neg A) \cdot P(\neg A).$$

The three values $P(A)$, $P(\neg A)$, $P(B|A)$ on the right side are known, but the value $P(B|\neg A)$ is unknown. We cannot make an exact statement about $P(B)$ with classical probability theory, but at the most we can estimate $P(B) \geq P(B|A) \cdot P(A)$.

We now consider the distribution

$$\boldsymbol{P}(A,\ B) = (P(A,\ B),\ P(A,\ \neg B),\ P(\neg A, B),\ P(\neg A,\ \neg B))$$

and introduce for shorthand the four unknowns

$$\begin{aligned} p_1 &= P(A, B), \\ p_2 &= P(A, \neg B), \\ p_3 &= P(\neg A, B), \\ p_4 &= P(\neg A, \neg B). \end{aligned}$$

These four parameters determine the distribution. If they are all known, then every probability for the two variables A and B can be calculated. To calculate the four unknowns, four equations are needed. One equation is already known in the form of the normalization condition

$$p_1 + p_2 + p_3 + p_4 = 1.$$

Therefore, three more equations are needed. In our example, however, only two equations are known.

From the given values $P(A) = \alpha$ and $P(B|A) = \beta$ we calculate

$$P(A, B) = P(B|A) \cdot P(A) = \alpha\beta$$

and

$$P(A) = P(A, B) + P(A, \neg B).$$

From this we can set up the following system of equations and solve it as far as is possible:

$$p_1 = \alpha\beta, \tag{7.4}$$

$$p_1 + p_2 = \alpha, \tag{7.5}$$

$$p_1 + p_2 + p_3 + p_4 = 1, \tag{7.6}$$

$$(7.4)\text{ in }(7.5): \quad p_2 = \alpha - \alpha\beta = \alpha(1-\beta), \tag{7.7}$$

$$(7.5)\text{ in }(7.6): \quad p_3 + p_4 = 1 - \alpha. \tag{7.8}$$

The probabilities p_1, p_2 for the interpretations (A, B) and $(A, \neg B)$ are thus known, but for the values p_3, p_4 only one equation still remains. To come to a definite solution despite this missing knowledge, we change our point of view. We use the given equation as a constraint for the solution of an optimization problem.

We are looking for a distribution $\boldsymbol{p}$ (for the variables p_3, p_4) which maximizes the entropy

$$H(\boldsymbol{p}) = -\sum_{i=1}^{n} p_i \ln p_i = -p_3 \ln p_3 - p_4 \ln p_4 \tag{7.9}$$

under the constraint $p_3 + p_4 = 1 - \alpha$ (7.8). Why exactly should the entropy function be maximized? Because we are missing information about the distribution, it must somehow be added in. We could fix an ad hoc value, for example $p_3 = 0.1$. Yet it is better to determine the values p_3 and p_4 such that the information added is minimal. We can show (Sect. 8.4.2 and [SW76]) that entropy measures the uncertainty of a distribution up to a constant factor. Negative entropy is then a measure of the amount of information a distribution contains. Maximization of entropy minimizes the information content of the distribution. To visualize this, the entropy function for the two-dimensional case is represented graphically in Fig. 7.2 on page 139.

To determine the maximum of the entropy under the constraint $p_3 + p_4 - 1 + \alpha = 0$ we use the method of Lagrange multipliers [Ste07]. The Lagrange function reads

$$L = -p_3 \ln p_3 - p_4 \ln p_4 + \lambda(p_3 + p_4 - 1 + \alpha).$$

Taking the partial derivatives with respect to p_3 and p_4 we obtain

$$\frac{\partial L}{\partial p_3} = -\ln p_3 - 1 + \lambda = 0,$$

$$\frac{\partial L}{\partial p_4} = -\ln p_4 - 1 + \lambda = 0$$

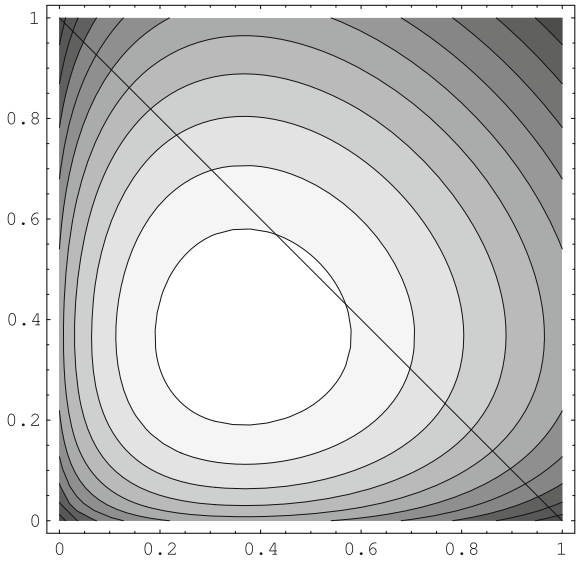


Fig. 7.2 *Contour line diagram* of the two-dimensional entropy function. We see that it is strictly convex in the whole unit square and that it has an isolated global maximum. Also marked is the constraint $p_3 + p_4 = 1$ as a special case of the condition $p_3 + p_4 - 1 + \alpha = 0$ for $\alpha = 0$ which is relevant here

and calculate

$$p_3 = p_4 = \frac{1-\alpha}{2}.$$

Now we can calculate the desired value

$$P(B) = P(A,B) + P(\neg A,B) = p_1 + p_3 = \alpha\beta + \frac{1-\alpha}{2} = \alpha\left(\beta - \frac{1}{2}\right) + \frac{1}{2}.$$

Substituting in α and β yields

$$P(B) = P(A)\left(P(B|A) - \frac{1}{2}\right) + \frac{1}{2}.$$

$P(B)$ is shown in Fig. 7.3 on page 140 for various values of $P(B|A)$. We see that in the two-value edge case, that is, when $P(B)$ and $P(B|A)$ take on the values 0 or 1, probabilistic inference returns the same value for $P(B)$ as the modus ponens. When A and $B|A$ are both true, B is also true. An interesting case is $P(A) = 0$, in which $\neg A$ is true. Modus ponens cannot be applied here, but our formula results in the value 1/2 for $P(B)$ irrespective of $P(B|A)$. When A is false, we know nothing about B, which reflects our intuition exactly. The case where $P(A) = 1$ and $P(B|A) = 0$ is also covered by propositional logic. Here A is true and $A \Rightarrow B$ false, and thus $A \wedge \neg B$ true. Then B is false. The horizontal line in the figure means that we cannot make a prediction about B in the case of $P(B|A) = 1/2$. Between these points, $P(B)$ changes linearly for changes to $P(A)$ or $P(B|A)$.

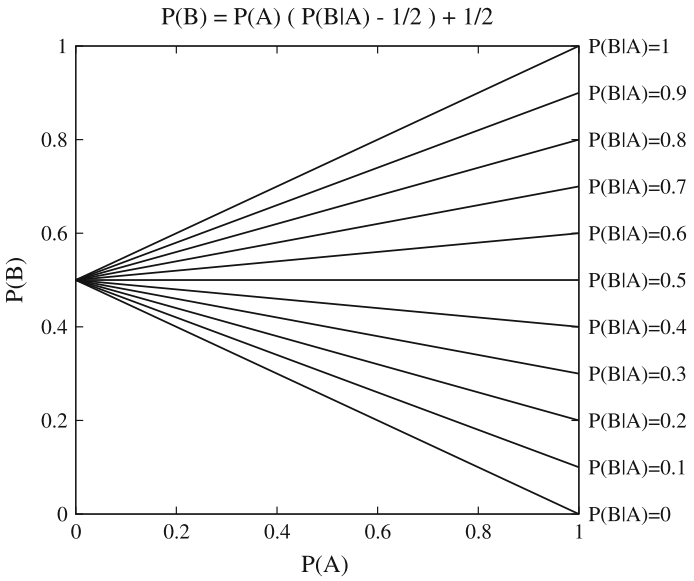


Fig. 7.3 Curve array for $P(B)$ as a function of $P(A)$ for different values of $P(B|A)$

Theorem 7.3 *Let there be a consistent*[5] *set of linear probabilistic equations. Then there exists a unique maximum for the entropy function with the given equations as constraints. The MaxEnt distribution thereby defined has minimum information content under the constraints.*

It follows from this theorem that there is no distribution which satisfies the constraints and has higher entropy than the MaxEnt distribution. A calculus that leads to lower entropy puts in additional ad hoc information, which is not justified.

Looking more closely at the above calculation of $P(B)$, we see that the two values p_3 and p_4 always occur symmetrically. This means that swapping the two variables does not change the result. Thus the end result is $p_3 = p_4$. The so-called indifference of these two variables leads to them being set equal by MaxEnt. This relationship is valid generally:

Definition 7.5 If an arbitrary exchange of two or more variables in the Lagrange equations results in equivalent equations, these variables are called *indifferent*.

[5]A set of probabilistic equations is called consistent if there is at least one solution, that is, one distribution which satisfies all equations.

Theorem 7.4 *If a set of variables* $\{p_{i_1}, \ldots, p_{i_k}\}$ *is indifferent, then the maximum of the entropy under the given constraints is at the point where* $p_{i_1} = p_{i_2} = \cdots = p_{i_k}$.

With this knowledge we could have immediately set the two variables p_3 and p_4 equal (without solving the Lagrange equations).

7.2.2 Maximum Entropy Without Explicit Constraints

We now look at the case in which no knowledge is given. This means that, other than the normalization condition

$$p_1 + p_2 + \cdots + p_n = 1,$$

there are no constraints. All variables are therefore indifferent. Therefore we can set them equal and it follows that $p_1 = p_2 = \cdots = p_n = 1/n$.[6] For reasoning under uncertainty, this means that given a complete lack of knowledge, all worlds are equally probable. That is, the distribution is uniform. For example, in the case of two variables A and B it would be the case that

$$P(A, B) = P(A, \neg B) = P(\neg A, B) = P(\neg A, \neg B) = 1/4,$$

from which $P(A) = P(B) = 1/2$ and $P(B|A) = 1/2$ follow. The result for the two-dimensional case can be seen in Fig. 7.2 on page 139 because the marked condition is exactly the normalization condition. We see that the maximum of the entropy lies on the line at exactly (1/2, 1/2).

As soon as the value of a condition deviates from the one derived from the uniform distribution, the probabilities of the worlds shift. We show this in a further example. With the same descriptions as used above we assume that only

$$P(B|A) = \beta$$

is known. Thus $P(A, B) = P(B|A)P(A) = \beta P(A)$, from which $p_1 = \beta(p_1 + p_2)$ follows and we derive the two constraints

$$\beta p_2 + (\beta - 1)p_1 = 0,$$

$$p_1 + p_2 + p_3 + p_4 - 1 = 0.$$

[6]The reader may calculate this result by maximization of the entropy under the normalization condition (Exercise 7.5 on page 132).

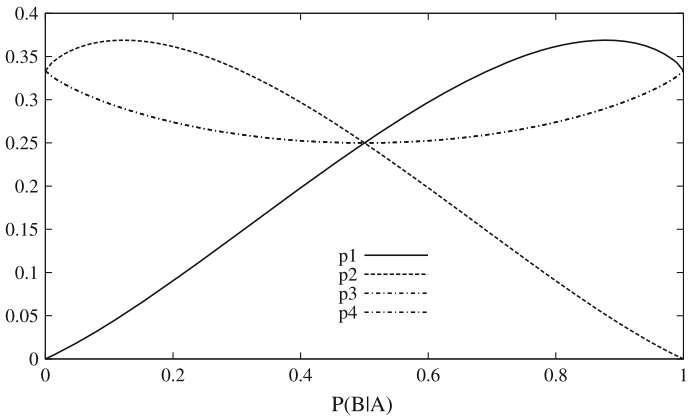


Fig. 7.4 p_1, p_2, p_3, p_4 in dependence on β

Here the Lagrange equations can no longer be solved symbolically so easily. A numeric solution of the Lagrange equations yields the picture represented in Fig. 7.4, which shows that $p_3 = p_4$. We can already see this in the constraints, in which p_3 and p_4 are indifferent. For $P(B|A) = 1/2$ we obtain the uniform distribution, which is no surprise. This means that the constraint for this value does not imply a restriction on the distribution. Furthermore, we can see that for small $P(B|A)$, $P(A, B)$ is also small.

7.2.3 Conditional Probability Versus Material Implication

We will now show that, for modeling reasoning, conditional probability is better than what is known in logic as material implication (to this end, also see [Ada75]). First we observe the truth table shown in Table 7.1, in which the conditional probability and material implication for the extreme cases of probabilities zero and one are compared. In both cases with false premises (which, intuitively, are critical cases), $P(B|A)$ is undefined, which makes sense.

Table 7.1 Truth table for material implication and conditional probability for propositional logic limit

A	B	$A \Rightarrow B$	$P(A)$	$P(B)$	$P(B\|A)$
t	t	t	1	1	1
t	f	f	1	0	0
f	t	t	0	1	*Undefined*
f	f	t	0	0	*Undefined*

Now we ask ourselves which value is taken on by $P(B|A)$ when arbitrary values $P(A) = \alpha$ and $P(B) = \gamma$ are given and no other information is known. Again we maximize entropy under the given constraints. As above we set

$$p_1 = P(A,B), \quad p_2 = P(A,\neg B), \quad p_3 = P(\neg A,B), \quad p_4 = P(\neg A,\neg B)$$

and obtain as constraints

$$p_1 + p_2 = \alpha, \tag{7.10}$$

$$p_1 + p_3 = \gamma, \tag{7.11}$$

$$p_1 + p_2 + p_3 + p_4 = 1. \tag{7.12}$$

With this we calculate using entropy maximization (see Exercise 7.8 on page 173)

$$p_1 = \alpha\gamma, \quad p_2 = \alpha(1-\gamma), \quad p_3 = \gamma(1-\alpha), \quad p_4 = (1-\alpha)(1-\gamma).$$

From $p_1 = \alpha\gamma$ it follows that $P(A, B) = P(A) \cdot P(B)$, which means that A and B are independent. Because there are no constraints connecting A and B, the MaxEnt principle results in the independence of these variables. The right half of Table 7.1 on page 142 makes this easier to understand. From the definition

$$P(B|A) = \frac{P(A,B)}{P(A)}$$

it follows for the case $P(A) \neq 0$, that is, when the premise is not false, because A and B are independent, that $P(B|A) = P(B)$. For the case $P(A) = 0$, $P(B|A)$ remains undefined.

7.2.4 MaxEnt-Systems

As previously mentioned, due to the nonlinearity of the entropy function, MaxEnt optimization usually cannot be carried out symbolically for non-trivial problems. Thus two systems were developed for numerical entropy maximization. The first system, SPIRIT (Symmetrical Probabilistic Intensional Reasoning in Inference Networks in Transition, www.xspirit.de), [RM96] was built at Fernuniversität Hagen. The second, PIT (Probability Induction Tool) was developed at the Munich Technical University [Sch96, ES99, SE00]. We will now briefly introduce PIT.

The PIT system uses the sequential quadratic programming (SQP) method to find an extremum of the entropy function under the given constraints. As input, PIT

expects data containing the constraints. For example, the constraints $P(A) = \alpha$ and $P(B|A) = \beta$ from Sect. 7.2.1 have the form

```
var A{t,f}, B{t,f};

P([A=t]) = 0.6;
P([B=t] | [A=t]) = 0.3;

QP([B=t]);
QP([B=t] | [A=t]);
```

Because PIT performs a numerical calculation, we have to input explicit probability values. The second to last row contains the query `QP([B = t])`. This means that $P(B)$ is the desired value. At www.pit-systems.de under "Examples" we now put this input into a blank input page ("Blank Page") and start PIT. As a result we get

Nr.	Truth value	Probability	Query
1	UNSPECIFIED	3.800e–01	QP([B = t]);
2	UNSPECIFIED	3.000e–01	QP([A = t]-|> [B = t]);

and from there read off $P(B) = 0.38$ and $P(B|A) = 0.3$.

7.2.5 The Tweety Example

We now show, using the Tweety example from Sect. 4.3, that probabilistic reasoning and in particular MaxEnt are non-monotonic and model everyday reasoning very well. We model the relevant rules with probabilities as follows:

$$\begin{array}{lll} P(\mathit{bird}|\mathit{penguin}) = 1 & & \text{"penguins are birds"} \\ P(\mathit{flies}|\mathit{bird}) \in [0.95, 1] & & \text{"(almost all) birds can fly"} \\ P(\mathit{flies}|\mathit{penguin}) = 0 & & \text{"penguins cannot fly"} \end{array}$$

The first and third rules represent firm predictions, which can also be easily formulated in logic. In the second, however, we express our knowledge that almost all birds can fly by means of a probability interval. With the PIT input data

```
var penguin{yes,no}, bird{yes,no}, flies{yes,no};

P([bird=yes] | [penguin=yes]) = 1;
P([flies=yes] | [bird=yes]) IN [0.95,1];
P([flies=yes] | [penguin=yes]) = 0;

QP([flies=yes]| [penguin=yes]);
```

we get back the correct answer

Nr.	Truthvalue	Probability	Query
1	UNSPECIFIED	0.000e+00	QP([penguin = yes]-|> [flies = yes]);

with the proposition that penguins cannot fly.[7] The explanation for this is very simple. With $P(flies|bird) \in [0.95, 1]$ it is possible that there are non-flying birds. If this rule were replaced by $P(flies|bird) = 1$, then PIT would not be able to do anything and would output an error message about inconsistent constraints.

In this example we can easily see that probability intervals are often very helpful for modeling our ignorance about exact probability values. We could have made an even fuzzier formulation of the second rule in the spirit of "normally birds fly" with $P(flies|bird) \in (0.5, 1]$. The use of the half-open interval excludes the value 0.5.

It has already been shown in [Pea88] that this example can be solved using probabilistic logic, even without MaxEnt. In [Sch96] it is shown for a number of demanding benchmarks for non-monotonic reasoning that these can be solved elegantly with MaxEnt. In the following section we introduce a successful practical application of MaxEnt in the form of a medical expert system.

7.3 LEXMED, an Expert System for Diagnosing Appendicitis

The medical expert system LEXMED, which uses the MaxEnt method, was developed at the Ravensburg-Weingarten University of Applied Sciences by Manfred Schramm, Walter Rampf, and the author, in cooperation with the Weingarten 14-Nothelfer Hospital [SE00, Le999].[8] The acronym LEXMED stands for *learning expert system for medical diagnosis*.

7.3.1 Appendicitis Diagnosis with Formal Methods

The most common serious cause of acute abdominal pain [dD91] is appendicitis—an inflammation of the appendix, a blind-ended tube connected to the cecum. Even today, diagnosis can be difficult in many cases [OFY+95]. For example, up to about 20% of the removed appendices are without pathological findings, which means that the operations were unnecessary. Likewise, there are regularly cases in which an inflamed appendix is not recognized as such.

Since as early as the beginning of the 1970s, there have been attempts to automate the diagnosis of appendicitis, with the goal of reducing the rate of false

[7] `QP([penguin=yes]-|> [flies=yes])` is an alternative form of the PIT syntax for `QP([flies=yes] | [penguin=yes])`.

[8] The project was financed by the German state of Baden-Württemberg, the health insurance company AOK Baden-Württemberg, the Ravensburg-Weingarten University of Applied Sciences, and the 14 Nothelfer Hospital in Weingarten.

diagnoses [dDLS+72, OPB94, OFY+95]. Especially noteworthy is the expert system for diagnosis of acute abdominal pain, developed by de Dombal in Great Britain. It was made public in 1972, thus distinctly earlier than the famous system MYCIN.

Nearly all of the formal diagnostic processes used in medicine to date have been based on scores. Score systems are extremely easy to apply: For each value of a symptom (for example *fever* or *lower right stomach pain*) the doctor notes a certain number of points. If the sum of the points is over a certain value (threshold), a certain decision is recommended (for example operation). For n symptoms $S_1, \ldots, S_n$ a score for appendicitis can be described formally as

$$\text{Diagnose} = \begin{cases} \text{Appendicitis} & \text{if } w_1 S_1 + \cdots + w_n S_n > \Theta, \\ \text{negative} & \text{else.} \end{cases}$$

With scores, a linear combination of symptom values is thus compared with a threshold Θ. The weights of the symptoms are extracted from databases using statistical methods. The advantage of scores is their simplicity of application. The weighted sum of the points can be computed by hand easily and a computer is not needed for the diagnosis.

Because of the linearity of this method, scores are too weak to model complex relationships. Since the contribution $w_i S_i$ of a symptom S_i to the score is calculated independently of the other symptoms, it is clear that score systems cannot take any "context" into account. Principally, they cannot distinguish between combinations of symptoms, for example they cannot distinguish between the white blood cell count of an old patient and that of a young patient.

For a fixed given set of symptoms, conditional probability is much more powerful than scores for making predictions because the latter cannot describe the dependencies between different symptoms. We can show that scores implicitly assume that all symptoms are independent.

When using scores, yet another problem comes up. To arrive at a good diagnosis quality, we must put strict requirements on the databases used to statistically determine the weights w_i. In particular they must be representative of the set of patients in the area in which the diagnosis system is used. This is often difficult, if not impossible, to guarantee. In such cases, scores and other statistical methods either cannot be used, or will have a high rate of errors.

7.3.2 Hybrid Probabilistic Knowledge Base

Complex probabilistic relationships appear frequently in medicine. With LEXMED, these relationships can be modeled well and calculated quickly. Here the use of probabilistic propositions, with which uncertain and incomplete information can be expressed and processed in an intuitive and mathematically grounded way, is essential. The following question may serve as a typical query against the expert system: "How high is the probability of an inflamed appendix if the patient is a 23-year-old man with pain in the right lower abdomen and a white blood cell count

Table 7.2 Symptoms used for the query in LEXMED and their values. The number of values for the each symptom is given in the column marked #

Symptom	Values	#	Short
Gender	*Male*, *female*	2	*Sex2*
Age	*0–5*, *6–10*, *11–15*, *16–20*, *21–25*, *26–35*, *36–45*, *46–55*, *56–65*, *65–*	10	*Age10*
Pain 1st Quad.	*Yes*, *no*	2	*P1Q2*
Pain 2nd Quad.	*Yes*, *no*	2	*P2Q2*
Pain 3rd Quad.	*Yes*, *no*	2	*P3Q2*
Pain 4th Quad.	*Yes*, *no*	2	*P4Q2*
Guarding	*Local*, *global*, *none*	3	*Gua3*
Rebound tenderness	*Yes*, *no*	2	*Reb2*
Pain on tapping	*Yes*, *no*	2	*Tapp2*
Rectal pain	*Yes*, *no*	2	*RecP2*
Bowel sounds	*Weak*, *normal*, *increased*, *none*	4	*BowS4*
Abnormal ultrasound	*Yes*, *no*	2	*Sono2*
Abnormal urine sedim.	*Yes*, *no*	2	*Urin2*
Temperature (rectal)	*–37.3*, *37.4–37.6*, *37.7–38.0*, *38.1–38.4*, *38.5–38.9*, *39.0–*	6	*TRec6*
Leukocytes	*0–6k*, *6k–8k*, *8k–10k*, *10k–12k*, *12k–15k*, *15k–20k*, *20k–*	7	*Leuko7*
Diagnosis	*Inflamed*, *perforated*, *negative*, *other*	4	*Diag4*

of 13,000?" Formulated as conditional probability, using the names and value ranges for the symptoms used in Table 7.2, this reads

$$P(Diag4 = inflamed \vee Diag4 = perforated \mid Sex2 = male \wedge Age10 \in 21{-}25 \wedge Leuko7 \in 12k{-}15k).$$

By using probabilistic propositions, LEXMED has the ability to use information from non-representative databases because this information can be complemented appropriately from other sources. Underlying LEXMED is a database which only contains data about patients whose appendixes were surgically removed. With statistical methods, (about 400) rules are generated which compile the knowledge contained in the database into an abstracted form [ES99]. Because there are no patients in this database who were suspected of having appendicitis but had negative diagnoses (that is, not requiring treatment),[9] there is no knowledge about negative patients in the database. Thus knowledge from other sources must be added in. In LEXMED therefore the rules gathered from the database are complemented by (about 100) rules from medical experts and the medical literature. This results in a hybrid probabilistic database, which contains knowledge extracted from data as well as knowledge explicitly formulated by experts. Because both types of rules are formulated as conditional probabilities (see for example (7.14) on page 152), they can be easily combined, as shown in Fig. 7.5 on page 148 and with more details in Fig. 7.7 on page 150.

[9]These negative diagnoses are denoted "non-specific abdominal pain" (*NSAP*).

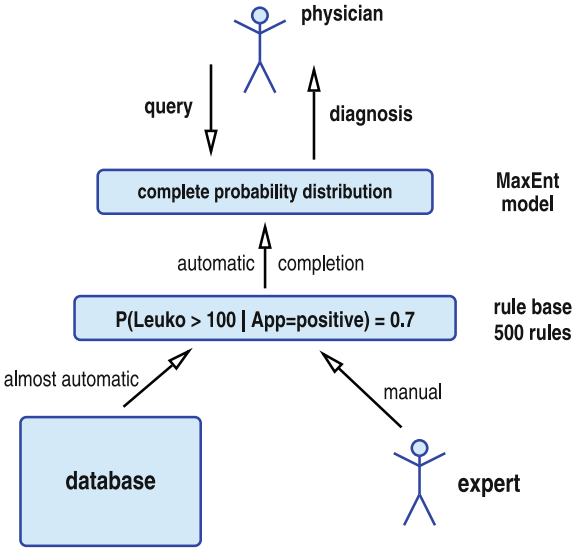


Fig. 7.5 Probabilistic rules are generated from data and expert knowledge, which are integrated in a rule base (knowledge base) and finally made complete using the MaxEnt method

LEXMED calculates the probabilities of various diagnoses using the probability distribution of all relevant variables (see Table 7.2 on page 147). Because all 14 symptoms used in LEXMED and the diagnoses are modeled as discrete variables (even continuous variables like the leukocyte value are divided into ranges), the size of the distribution (that is, the size of the event space) can be determined using Table 7.2 on page 147 as the product of the number of values of all symptoms, or

$$2^{10} \cdot 10 \cdot 3 \cdot 4 \cdot 6 \cdot 7 \cdot 4 = 20643840$$

elements. Due to the normalization condition from Theorem 7.1 on page 129, it thus has 20643839 independent values. Every rule set with fewer than 20643839 probability values potentially does not completely describe this event space. To be able to answer any arbitrary query, the expert system needs a complete distribution. The construction of such an extensive, consistent distribution using statistical methods is very difficult.[10] To require from a human expert all 20643839 values for the distribution (instead of the aforementioned 100 rules) would essentially be impossible.

Here the MaxEnt method comes into play. The generalization of about 500 rules to a complete probability model is done in LEXMED by maximizing the entropy with the 500 rules as constraints. An efficient encoding of the resulting MaxEnt distribution leads to response times for the diagnosis of around one second.

[10]The task of generating a function from a set of data is known as machine learning. We will cover this thoroughly in Chap. 8.

Personal Details	unknown	values	
Gender	◉	○ male ○ female	?
Age-group	○	○ 0-5 ○ 6-10 ○ 11-15 ○ 16-20 ◉ 21-25 ○ 26-35 ○ 36-45 ○ 46-55 ○ 56-65 ○ 65-	?
Results of examination	**not done**	**values**	
1st quadrant	◉	○ yes ○ no	?
2nd quadrant	◉	○ yes ○ no	?
3rd quadrant	○	◉ yes ○ no	?
4th quadrant	◉	○ yes ○ no	?
guarding	○	◉ local ○ global ○ none	?
rebound tenderness	○	◉ yes ○ no	?
pain on tapping	○	◉ yes ○ no	?
rectal pain	◉	○ yes ○ no	?
bowel sounds	○	○ weak ◉ normal ○ increased ○ none	?
abnormal ultrasound	◉	○ yes ○ no	?
abnormal urine sediment	◉	○ yes ○ no	?
temperature range (rectal)	○	○ -37.3 ○ 37.4-37.6 ○ 37.7-38.0 ○ 38.1-38.4 ◉ 38.5-38.9 ○ 39.0-	?
leucocyte count	○	○ 0-6k ○ 6k-8k ○ 8k-10k ○ 10k-12k ◉ 12k-15k ○ 15k-20k ○ 20k-	?

	Result of the PIT diagnosis			
Diagnosis	App. inflamed	App. perforated	Negative	Other
Probability	0.70	0.17	0.06	0.07

Fig. 7.6 The LEXMED input mask for input of the examined symptoms and below it the output of the resulting diagnosis probabilities

7.3.3 Application of LEXMED

The usage of LEXMED is simple and self-explanatory. The doctor visits the LEXMED home page at www.lexmed.de.[11] For an automatic diagnosis, the doctor inputs the results of his examination into the input form in Fig. 7.6. After one or two seconds he receives the probabilities for the four different diagnoses as well as a suggestion for a treatment (Sect. 7.3.5). If certain examination results are missing as input (for example the sonogram results), then the doctor chooses the entry *not examined*. Naturally the certainty of the diagnosis is higher when more symptom values are input.

[11] A version with limited functionality is accessible without a password.

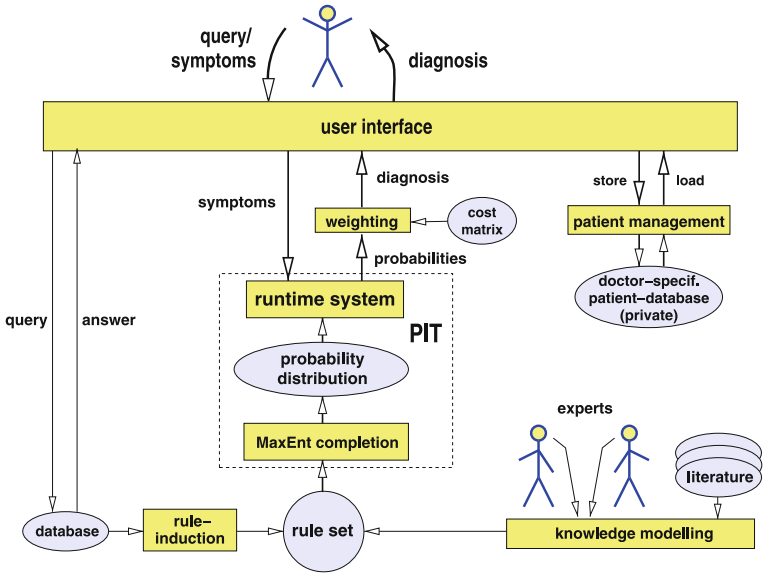


Fig. 7.7 Rules are generated from the database as well as from expert knowledge. From these, MaxEnt creates a complete probability distribution. For a user query, the probability of every possible diagnosis is calculated. Using the cost matrix (see Sect. 7.3.5) a decision is then suggested

Each registered user has access to a private patient database, in which input data can be archived. Thus data and diagnoses from earlier patients can be easily compared with those of a new patient. An overview of the processes in LEXMED is given in Fig. 7.7.

7.3.4 Function of LEXMED

Knowledge is formalized using probabilistic propositions. For example, the proposition

$$P(Leuko7 > 20000|Diag4 = inflamed) = 0.09$$

gives a frequency of 9% for a leukocyte value of more than 20,000 in case of an inflamed appendix.[12]

Learning of Rules by Statistical Induction

The raw data in LEXMED's database contain 54 different (anonymized) values for 14,646 patients. As previously mentioned, only patients whose appendixes were surgically removed are included in this database. Of the 54 attributes used in the

[12]Instead of individual numerical values, intervals can also be used here (for example [0.06, 0.12]).

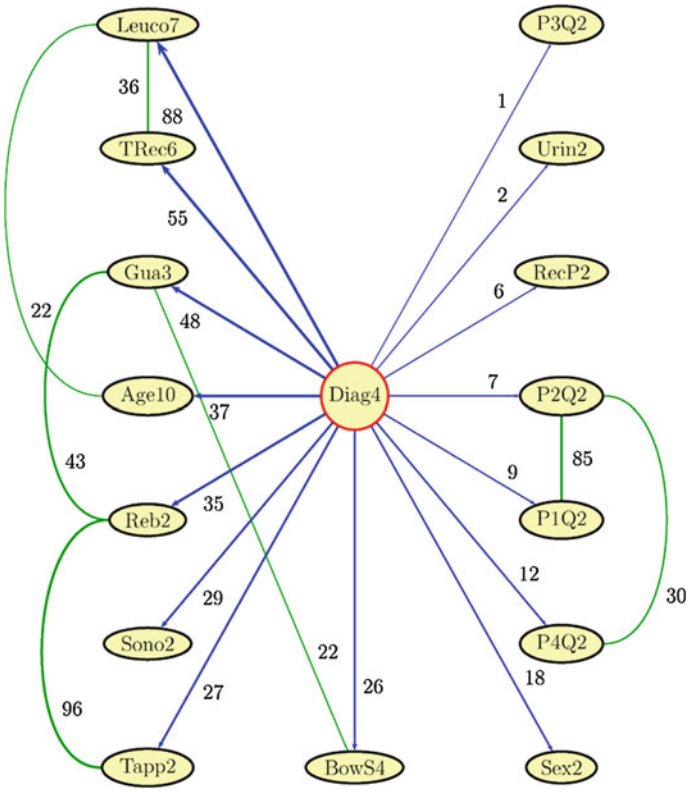


Fig. 7.8 Dependency graph computed from the database

database, after a statistical analysis the 14 symptoms shown in Table 7.2 on page 147 were used. Now the rules are created from this database in two steps. The first step determines the dependency structure of the symptoms. The second step fills this structure with the respective probability rules.[13]

Determining the Dependency Graph The graph in Fig. 7.8 contains for each variable (the symptom and the diagnosis) a node and directed edges which connect various nodes. The thickness of the edges between the variables represents a measure of the statistical dependency or correlation of the variables. The correlation of two independent variables is equal to zero. The pair correlation for each of the 14 symptoms with *Diag4* was computed and listed in the graph. Furthermore, all triple correlations between the diagnosis and two symptoms were calculated. Of these, only the strongest values have been drawn as additional edges between the two participating symptoms.

[13]For a systematic introduction to machine learning we refer the reader to Chap. 8.

```
 1 P([Leuco7=0-6k]   | [Diag4=negativ] * [Age10=16-20]) = [0.132,0.156];
 2 P([Leuco7=6-8k]   | [Diag4=negativ] * [Age10=16-20]) = [0.257,0.281];
 3 P([Leuco7=8-10k]  | [Diag4=negativ] * [Age10=16-20]) = [0.250,0.274];
 4 P([Leuco7=10-12k] | [Diag4=negativ] * [Age10=16-20]) = [0.159,0.183];
 5 P([Leuco7=12-15k] | [Diag4=negativ] * [Age10=16-20]) = [0.087,0.112];
 6 P([Leuco7=15-20k] | [Diag4=negativ] * [Age10=16-20]) = [0.032,0.056];
 7 P([Leuco7=20k-]   | [Diag4=negativ] * [Age10=16-20]) = [0.000,0.023];
 8 P([Leuco7=0-6k]   | [Diag4=negativ] * [Age10=21-25]) = [0.132,0.172];
 9 P([Leuco7=6-8k]   | [Diag4=negativ] * [Age10=21-25]) = [0.227,0.266];
10 P([Leuco7=8-10k]  | [Diag4=negativ] * [Age10=21-25]) = [0.211,0.250];
11 P([Leuco7=10-12k] | [Diag4=negativ] * [Age10=21-25]) = [0.166,0.205];
12 P([Leuco7=12-15k] | [Diag4=negativ] * [Age10=21-25]) = [0.081,0.120];
13 P([Leuco7=15-20k] | [Diag4=negativ] * [Age10=21-25]) = [0.041,0.081];
14 P([Leuco7=20k-]   | [Diag4=negativ] * [Age10=21-25]) = [0.004,0.043];
```

Fig. 7.9 Some of the LEXMED rules with probability intervals. "*" stands for "∧" here

Estimating the Rule Probabilities The structure of the dependency graph describes the structure of the learned rules.[14] The rules here have different complexities: there are rules which only describe the distribution of the possible diagnoses (a priori rules, for example (7.13)), rules which describe the dependency between the diagnosis and a symptom (rules with simple conditions, for example (7.14)), and finally rules which describe the dependency between the diagnosis and two symptoms, as given in Fig. 7.9 in PIT syntax.

$$P(Diag4 = inflamed) = 0.40, \tag{7.13}$$

$$P(Sono2 = yes|Diag4 = inflamed) = 0.43, \tag{7.14}$$

$$P(P4Q2 = yes|Diag4 = inflamed \wedge P2Q2 = yes) = 0.61. \tag{7.15}$$

To keep the context dependency of the saved knowledge as small as possible, all rules contain the diagnosis in their conditions and not as conclusions. This is quite similar to the construction of many medical books with formulations of the kind "With appendicitis we usually see …". As previously shown in Example 7.6 on page 133, however, this does not present a problem because, using the Bayesian formula, LEXMED automatically puts these rules into the right form.

The numerical values for these rules are estimated by counting their frequency in the database. For example, the value in (7.14) is given by counting and calculating

$$\frac{|Diag4 = inflamed \wedge Sono2 = yes|}{|Diag4 = inflamed|} = 0.43.$$

[14]The difference between this and a Bayesian network is, for example, that the rules are equipped with probability intervals and that only after applying the principle of maximum entropy is a unique probability model produced.

Expert Rules

Because the appendicitis database only contains patients who have undergone the operation, rules for non-specific abdominal pain (*NSAP*) receive their values from propositions of medical experts. The experiences in LEXMED confirm that the probabilistic rules are easy to read and can be directly translated into natural language. Statements by medical experts about frequency relationships of specific symptoms and the diagnosis, whether from the literature or as the result of an interview, can therefore be incorporated into the rule base with little expense. To model the uncertainty of expert knowledge, the use of probability intervals has proven effective. The expert knowledge was primarily acquired from the participating surgeons, Dr. Rampf and Dr. Hontschik, and their publications [Hon94].

Once the expert rules have been created, the rule base is finished. Then the complete probability model is calculated with the method of maximum entropy by the PIT-system.

Diagnosis Queries

Using its efficiently stored probability model, LEXMED calculates the probabilities for the four possible diagnoses within a few seconds. For example, we assume the following output:

	Results of the PIT diagnosis			
Diagnosis	Appendix inflamed	Appendix perforated	Negative	Other
Probability	0.24	0.16	0.57	0.03

A decision must be made based on these four probability values to pursue one of the four treatments: operation, emergency operation, stationary observation, or ambulant observation.[15] While the probability for a negative diagnosis in this case outweighs the others, sending the patient home as healthy is not a good decision. We can clearly see that, even when the probabilities of the diagnoses have been calculated, the diagnosis is not yet finished.

Rather, the task is now to derive an optimal decision from these probabilities. To this end, the user can have LEXMED calculate a recommended decision.

7.3.5 Risk Management Using the Cost Matrix

How can the computed probabilities now be translated optimally into decisions? A naive algorithm would assign a decision to each diagnosis and ultimately select the decision that corresponds to the highest probability. Assume that the computed probabilities are 0.40 for the diagnosis *appendicitis* (inflamed or perforated), 0.55 for the diagnosis *negative*, and 0.05 for the diagnosis *other*. A naive algorithm would now choose the (too risky) decision "no operation" because it corresponds to the diagnosis with the higher probability. A better method consists of comparing

[15] Ambulant observation means that the patient is released to stay at home.

Table 7.3 The cost matrix of LEXMED together with a patient's computed diagnosis probabilities

Therapy	Probability of various diagnoses				
	Inflamed	*Perforated*	*Negative*	*Other*	
	0.25	0.15	0.55	0.05	
Operation	0	500	5800	6000	3565
Emergency operation	500	0	6300	6500	3915
Ambulant observ.	12000	**150000**	0	16500	26325
Other	3000	5000	1300	0	2215
Stationary observ.	3500	7000	400	600	**2175**

the costs of the possible errors that can occur for each decision. The error is quantified in the form of "(hypothetical) additional cost of the current decision compared to the optimum". The given values contain the costs to the hospital, to the insurance company, the patient (for example risk of post-operative complications), and to other parties (for example absence from work), taking into account long term consequences. These costs are given in Table 7.3.

The entries are finally averaged for each decision, that is, summed while taking into account their frequencies. These are listed in the last column in Table 7.3. Finally, the decision with the smallest average cost of error is suggested. In Table 7.3 the matrix is given together with the probability vector calculated for a patient (in this case: (0.25, 0.15, 0.55, 0.05)). The last column of the table contains the result of the calculations of the average expected costs of the errors. The value of *Operation* in the first row is thus calculated as $0.25 \cdot 0 + 0.15 \cdot 500 + 0.55 \cdot 5800 + 0.05 \cdot 6000 = 3565$, a weighted average of all costs. The optimal decisions are entered with (additional) costs of 0. The system decides on the treatment with the minimal average cost. It thus is an example of a cost-oriented agent.

Cost Matrix in the Binary Case

To better understand the cost matrix and risk management we will now restrict the LEXMED system to the two-value decision between the diagnosis *appendicitis* with probability

$$p_1 = P(appendicitis) = P(Diag4 = inflamed) + P(Diag4 = perforated)$$

and *NSAP* with the probability

$$p_2 = P(NSAP) = P(Diag4 = negative) + P(Diag4 = other)$$

The only available treatments are *operation* and *ambulant observation*.

The cost matrix is thus a 2×2 matrix of the form

$$\begin{pmatrix} 0 & k_2 \\ k_1 & 0 \end{pmatrix}.$$

The two zeroes in the diagonal stand for the correct decision *operation* in the case of *appendicitis* and *ambulant observation* for *NSAP*. The parameter k_2 stands for the expected costs which occur when a patient without an inflamed appendix is operated on. This error is called a *false positive*. On the other hand, the decision *ambulant observation* in the case of appendicitis is a *false negative*. The probability vector $(p_1, p_2)^T$ is now multiplied by this matrix and we obtain the vector

$$(k_2\, p_2,\ k_1\, p_1)^T$$

with the average additional cost for the two possible treatments. Because the decision only takes into account the relationship of the two components, the vector can be multiplied by any scalar factor. We choose $1/k_1$ and obtain $((k_2/k_1)\ p_2, p_1)$. Thus only the relationship $k = k_2/k_1$ is relevant here. The same result is obtained by the simpler cost matrix

$$\begin{pmatrix} 0 & k \\ 1 & 0 \end{pmatrix},$$

which only contains the variable k. This parameter is very important because it determines risk management. By changing k we can fit the "working point" of the diagnosis system. For $k \rightarrow \infty$ the system is put in an extremely risky setting because no patient will ever be operated on, with the consequence that it gives no false positive classifications, but many False negatives. In the case of $k = 0$ the conditions are in exact reverse and all patients are operated upon.

7.3.6 Performance

LEXMED is intended for use in a medical practice or ambulance. Prerequisites for the use of LEXMED are acute abdominal pain for several hours (but less than five days). Furthermore, LEXMED is (currently) specialized for appendicitis, which means that for other illnesses the system contains very little information.

In the scope of a prospective study, a representative database with 185 cases was created in the 14 Nothelfer Hospital. It contains the hospital's patients who came to the clinic after several hours of acute abdominal pain and suspected appendicitis. From these patients, the symptoms and the diagnosis (verified from a tissue sample in the case of an operation) is noted.

If the patients were released to go home (without operation) after a stay of several hours or 1–2 days with little or no complaint, it was afterwards inquired by telephone whether the patient remained free of symptoms or whether a positive diagnosis was found in subsequent treatment.

To simplify the representation and make for a better comparison to similar studies, LEXMED was restricted to the two-value distinction between *appendicitis* and *NSAP*, as described in Sect. 7.3.5. Now k is varied between zero and infinity

and for every value of k the *sensitivity* and *specificity* are measured against the test data. Sensitivity measures

$$P(\text{classified positive}|\text{positive}) = \frac{|\text{positive and classified positive}|}{|\text{positive}|}, \qquad (7.16)$$

that is, the relative portion of positive cases which are correctly identified. It indicates how sensitive the diagnostic system is. Specificity, on the other hand, measures

$$P(\text{classified negative}|\text{negative}) = \frac{|\text{negative and classified negative}|}{|\text{negative}|}, \qquad (7.17)$$

that is, the relative portion of negative cases which are correctly identified.

We give the results of the sensitivity and specificity in Fig. 7.10 for $0 \leq k < \infty$. This curve is denoted the *ROC curve*, or receiver operating characteristic. Before we come to the analysis of the quality of LEXMED, a few words about the meaning of the ROC curve. The line bisecting the diagram diagonally is drawn in for orientation. All points on this line correspond to a random decision. For example, the point (0.2, 0.2) corresponds to a specificity of 0.8 with a sensitivity of 0.2. We can arrive at this quite easily by classifying a new case, without looking at it, with probabilities 0.2 for positive and 0.8 for negative. Every knowledge-based diagnosis system must therefore generate a ROC which clearly lies above the diagonal.

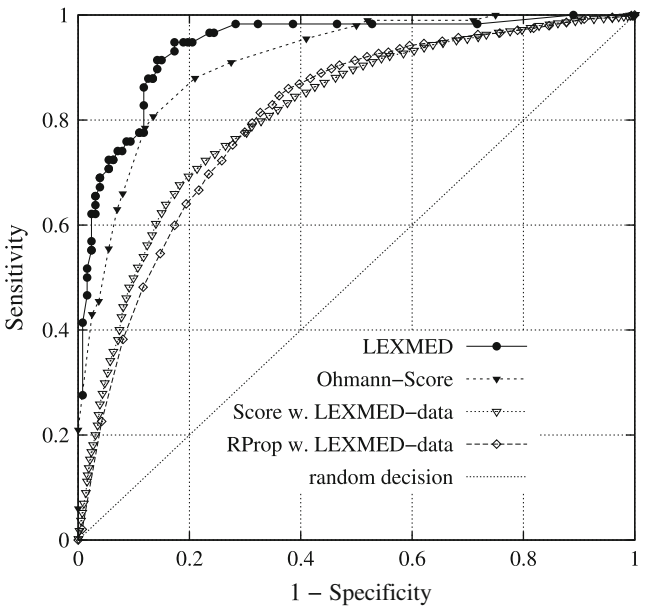


Fig. 7.10 ROC curve from LEXMED compared with the Ohmann score and two additional models

The extreme values in the ROC curve are also interesting. At point (0, 0) all three curves intersect. The corresponding diagnosis system would classify all cases as negative. The other extreme value (1, 1) corresponds to a system which would decide to do the operation for every patient and thus has a sensitivity of 1. We could call the ROC curve the characteristic curve for two-value diagnostic systems. The ideal diagnostic system would have a characteristic curve which consists only of the point (0, 1), and thus has 100% specificity and 100% sensitivity.

Now let us analyse the ROC curve. At a sensitivity of 88%, LEXMED attains a specificity of 87% ($k = 0.6$). For comparison, the Ohmann score, an established, well-known score for appendicitis is given [OMYL96, ZSR+99]. Because LEXMED is above or to the left of the Ohmann score almost everywhere, its average quality of diagnosis is clearly better. This is not surprising because scores are simply too weak to model interesting propositions. In Sect. 8.7 and in Exercise 8.17 on page 242 we will show that scores are equivalent to the special case of naive Bayes, that is, to the assumption that all symptoms are pairwise independent when the diagnosis is known. When comparing LEXMED with scores it should, however, be mentioned that a statistically representative database was used for the Ohmann score, but a non-representative database enhanced with expert knowledge was used for LEXMED. To get an idea of the quality of the LEXMED data in comparison to the Ohmann data, a linear score was calculated using the least squares method (see Sect. 9.4.1), which is also drawn for comparison. Furthermore, a neural network was trained on the LEXMED data with the RProp algorithm (see Sect. 9.5). The strength of combining data and expert knowledge is displayed clearly in the difference between the LEXMED curve and the curves of the score system and the RProp algorithm.

7.3.7 Application Areas and Experiences

LEXMED should not replace the judgment of an experienced surgeon. However, because a specialist is not always available in a clinical setting, a LEXMED query offers a substantive second opinion. Especially interesting and worthwhile is the application of the system in a clinical ambulance and for general practitioners.

The learning capability of LEXMED, which makes it possible to take into account further symptoms, further patient data, and further rules, also presents new possibilities in the clinic. For especially rare groups which are difficult to diagnose, for example children under six years of age, LEXMED can use data from pediatricians or other special databases, to support even experienced surgeons.

Aside from direct use in diagnosis, LEXMED also supports quality assurance measures. For example, insurance companies can compare the quality of diagnosis of hospitals with that of expert systems. By further developing the cost matrix created in LEXMED (with the consent of doctors, insurance, and patients), the quality of physician diagnoses, computer diagnoses, and other medical institutions will become easier to compare.

LEXMED has pointed to a new way of constructing automatic diagnostic systems. Using the language of probability theory and the MaxEnt algorithm, inductively,

statistically derived knowledge is combined with knowledge from experts and from the literature. The approach based on probabilistic models is theoretically elegant, generally applicable, and has given very good results in a small study.

LEXMED has been in practical use in the 14 Nothelfer Hospital in Weingarten since 1999 and has performed there very well. It is also available at www.lexmed.de, without warranty, of course. Its quality of diagnosis is comparable with that of an experienced surgeon and is thus better than that of an average general practitioner, or that of an inexperienced doctor in the clinic.

Despite this success it has become evident that it is very difficult to market such a system commercially in the German medical system. One reason for this is that there is no free market to promote better quality (here better diagnoses) through its selection mechanisms. Furthermore, in medicine the time for broad use of intelligent techniques is not yet at hand—even in 2010. One cause of this could be conservative teachings in this regard in German medical school faculties.

A further issue is the desire of many patients for personal advice and care from the doctor, together with the fear that, with the introduction of expert systems, the patient will only communicate with the machine. This fear, however, is wholly unfounded. Even in the long term, medical expert systems cannot replace the doctor. They can, however, just like laser surgery and magnetic resonance imaging, be used advantageously for all participants. Since the first medical computer diagnostic system of de Dombal in 1972, almost 40 years have passed. It remains to be seen whether medicine will wait another 40 years until computer diagnostics becomes an established medical tool.

7.4 Reasoning with Bayesian Networks

One problem with reasoning using probability in practice was already pointed out in Sect. 7.1. If d variables $X_1, \ldots, X_d$ with n values each are used, then the associated probability distribution has n^d total values. This means that in the worst case the memory use and computation time for determining the specified probabilities grows exponentially with the number of variables.

In practice the applications are usually very structured and the distribution contains many redundancies. This means that it can be heavily reduced with the appropriate methods. The use of Bayesian networks has proved its power here and is one of the AI techniques which have been successfully used in practice. Bayesian networks utilize knowledge about the independence of variables to simplify the model.

7.4.1 Independent Variables

In the simplest case, all variables are pairwise independent and it is the case that

$$\boldsymbol{P}(X_1, \ldots, X_d) = \boldsymbol{P}(X_1) \cdot \boldsymbol{P}(X_2) \cdot \cdots \cdot \boldsymbol{P}(X_d).$$

All entries in the distribution can thus be calculated from the d values $P(X_1), \ldots, P(X_d)$. Interesting applications, however, can usually not be modeled because conditional probabilities become trivial.[16] Because of

$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(A)P(B)}{P(B)} = P(A)$$

all conditional probabilities are reduced to the a priori probabilities. The situation becomes more interesting when only a portion of the variables are independent or independent under certain conditions. For reasoning in AI, the dependencies between variables happen to be important and must be utilized.

We would like to outline reasoning with Bayesian networks through a simple and very illustrative example by J. Pearl [Pea88], which became well known through [RN10] and is now basic AI knowledge.

Example 7.10 (Alarm-Example) Bob, who is single, has had an alarm system installed in his house to protect against burglars. Bob cannot hear the alarm when he is working at the office. Therefore he has asked his two neighbors, John in the house next door to the left, and Mary in the house to the right, to call him at his office if they hear his alarm. After a few years Bob knows how reliable John and Mary are and models their calling behavior using conditional probability as follows.[17]

$$\begin{aligned} P(J|Al) &= 0.90 & P(M|Al) &= 0.70, \\ P(J|\neg Al) &= 0.05 & P(M|\neg Al) &= 0.01. \end{aligned}$$

Because Mary is hard of hearing, she fails to hear the alarm more often than John. However, John sometimes mixes up the alarm at Bob's house with the alarms at other houses. The alarm is triggered by a burglary, but can also be triggered by a (weak) earthquake, which can lead to a false alarm because Bob only wants to know about burglaries while at his office. These relationships are modeled by

$$\begin{aligned} P(Al|Bur, Ear) &= 0.95, \\ P(Al|Bur, \neg Ear) &= 0.94, \\ P(Al|\neg Bur, Ear) &= 0.29, \\ P(Al|\neg Bur, \neg Ear) &= 0.001, \end{aligned}$$

as well as the a priori probabilities $P(Bur) = 0.001$ and $P(Ear) = 0.002$. These two variables are independent because earthquakes do not make plans based on the habits of burglars, and conversely there is no way to predict earthquakes, so burglars do not have the opportunity to set their schedule accordingly.

[16] In the naive Bayes method, the independence of all attributes is assumed, and this method has been successfully applied to text classification (see Sect. 8.7).

[17] The binary variables J and M stand for the two events "John calls", and "Mary calls", respectively, *Al* for "alarm siren sounds", *Bur* for "burglary" and *Ear* for "earthquake".

Queries are now made against this knowledge base. For example, Bob might be interested in $P(Bur|J \vee M)$, $P(J|Bur)$ or $P(M|Bur)$. That is, he wants to know how sensitively the variables J and M react to a burglary report.

7.4.2 Graphical Representation of Knowledge as a Bayesian Network

We can greatly simplify practical work by graphically representing knowledge that is formulated as conditional probability. Figure 7.11 shows the Bayesian network for the alarm example. Each node in the network represents a variable and every directed edge a statement of conditional probability. The edge from Al to J for example represents the two values $P(J|Al)$ and $P(J \mid \neg Al)$, which is given in the form of a table, the so-called CPT (conditional probability table). The CPT of a node lists all the conditional probabilities of the node's variable conditioned on all the nodes connected by incoming edges.

While studying the network, we might ask ourselves why there are no other edges included besides the four that are drawn in. The two nodes *Bur* and *Ear* are not linked since the variables are independent. All other nodes have a parent node, which makes the reasoning a little more complex. We first need the concept of conditional independence.

7.4.3 Conditional Independence

Analogously to independence of random variables, we give

Definition 7.6 Two variables A and B are called *conditionally independent*, given C if

$$\boldsymbol{P}(A, B|C) = \boldsymbol{P}(A|C) \cdot \boldsymbol{P}(B|C).$$

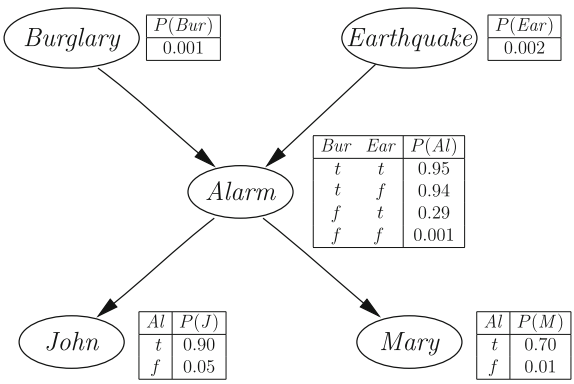


Fig. 7.11 Bayesian network for the alarm example with the associated CPTs

This equation is true for all combinations of values for all three variables (that is, for the distribution), which we see in the notation. We now look at nodes J and M in the alarm example, which have the common parent node Al. If John and Mary independently react to an alarm, then the two variables J and M are independent given Al, that is:

$$\boldsymbol{P}(J, M|Al) = \boldsymbol{P}(J|Al) \cdot \boldsymbol{P}(M|Al).$$

If the value of Al is known, for example because an alarm was triggered, then the variables J and M are independent (under the condition $Al = w$). Because of the conditional independence of the two variables J and M, no edge between these two nodes is added. However, J and M are not independent (see Exercise 7.11 on page 173).

Quite similar is the relationship between the two variables J and Bur, because John does not react to a burglary, rather the alarm. This could be, for example, because of a high wall that blocks his view on Bob's property, but he can still hear the alarm. Thus J and Bur are independent given Al and

$$\boldsymbol{P}(J, Bur|Al) = \boldsymbol{P}(J|Al) \cdot \boldsymbol{P}(Bur|Al).$$

Given an alarm, the variables J and Ear, M and Bur, as well as M and Ear are also independent. For computing with conditional independence, the following characterizations, which are equivalent to the above definition, are helpful:

Theorem 7.5 *The following equations are pairwise equivalent, which means that each individual equation describes the conditional independence for the variables A and B given C.*

$$\boldsymbol{P}(A, B|C) = \boldsymbol{P}(A|C) \cdot \boldsymbol{P}(B|C), \tag{7.18}$$

$$\boldsymbol{P}(A|B, C) = \boldsymbol{P}(A|C), \tag{7.19}$$

$$\boldsymbol{P}(B|A, C) = \boldsymbol{P}(B|C). \tag{7.20}$$

Proof On one hand, using conditional independence (7.18) we can conclude that

$$\boldsymbol{P}(A, B, C) = \boldsymbol{P}(A, B|C)\boldsymbol{P}(C) = \boldsymbol{P}(A|C)\,\boldsymbol{P}(B|C)\,\boldsymbol{P}(C).$$

On the other hand, the product rule gives us

$$\boldsymbol{P}(A, B, C) = \boldsymbol{P}(A|B, C)\boldsymbol{P}(B|C)\boldsymbol{P}(C).$$

Thus $\boldsymbol{P}(A|B, C) = \boldsymbol{P}(A|C)$ is equivalent to (7.18). We obtain (7.20) analogously by swapping A and B in this derivation. □

7.4.4 Practical Application

Now we turn again to the alarm example and show how the Bayesian network in Fig. 7.11 can be used for reasoning. Bob is interested, for example, in the sensitivity of his two alarm reporters John and Mary, that is, in $P(J|Bur)$ and $P(M|Bur)$. However, the values $P(Bur|J)$ and $P(Bur|M)$, as well as $P(Bur|J,M)$ are even more important to him. We begin with $P(J|Bur)$ and calculate

$$P(J|Bur) = \frac{P(J,Bur)}{P(Bur)} = \frac{P(J,Bur,Al) + P(J,Bur,\neg Al)}{P(Bur)} \tag{7.21}$$

and

$$\boldsymbol{P}(J,Bur,Al) = \boldsymbol{P}(J|Bur,Al)\boldsymbol{P}(Al|Bur)\boldsymbol{P}(Bur) = \boldsymbol{P}(J|Al)\boldsymbol{P}(Al|Bur)\boldsymbol{P}(Bur), \tag{7.22}$$

where for the last two equations we have used the product rule and the conditional independence of J and Bur given Al. Inserted in (7.21) we obtain

$$\begin{aligned} P(J|Bur) &= \frac{P(J|Al)P(Al|Bur)P(Bur) + P(J|\neg Al)P(\neg Al|Bur)P(Bur)}{P(Bur)} \\ &= P(J|Al)P(Al|Bur) + P(J|\neg Al)P(\neg Al|Bur). \end{aligned} \tag{7.23}$$

Here $P(Al|Bur)$ and $P(\neg Al|Bur)$ are missing. Therefore we calculate

$$\begin{aligned} P(Al|Bur) &= \frac{P(Al,Bur)}{P(Bur)} = \frac{P(Al,Bur,Ear) + P(Al,Bur,\neg Ear)}{P(Bur)} \\ &= \frac{P(Al|Bur,Ear)P(Bur)P(Ear) + P(Al|Bur,\neg Ear)P(Bur)P(\neg Ear)}{P(Bur)} \\ &= P(Al|Bur,Ear)P(Ear) + P(Al|Bur,\neg Ear)P(\neg Ear) \\ &= 0.95 \cdot 0.002 + 0.94 \cdot 0.998 = 0.94 \end{aligned}$$

as well as $P(\neg Al|Bur) = 0.06$ and insert this into (7.23) which gives the result

$$P(J|Bur) = 0.9 \cdot 0.94 + 0.05 \cdot 0.06 = 0.849.$$

Analogously we calculate $P(M|Bur) = 0.659$. We now know that John calls for about 85% of all break-ins and Mary for about 66% of all break-ins. The probability that both of them call is calculated, due to conditional independence, as

$$\begin{aligned} P(J,M|Bur) &= P(J,M|Al)P(Al|Bur) + P(J,M|\neg Al)P(\neg Al|Bur) \\ &= P(J|Al)P(M|Al)P(Al|Bur) + P(J|\neg Al)P(M|\neg Al)P(\neg Al|Bur) \\ &= 0.9 \cdot 0.7 \cdot 0.94 + 0.05 \cdot 0.01 \cdot 0.06 = 0.5922. \end{aligned}$$

More interesting, however, is the probability of a call from John or Mary

$$\begin{aligned} P(J \vee M|Bur) &= P(\neg(\neg J, \neg M)|Bur) = 1 - P(\neg J, \neg M|Bur) \\ &= 1 - [P(\neg J|Al)P(\neg M|Al)P(Al|Bur) + P(\neg J|\neg Al)P(\neg M|\neg Al)P(\neg Al|Bur)] \\ &= 1 - [0.1 \cdot 0.3 \cdot 0.94 + 0.95 \cdot 0.99 \cdot 0.06] = 1 - 0.085 = 0.915. \end{aligned}$$

Bob thus receives a notification for about 92% of all burglaries. Now to calculate $P(Bur|J)$, we apply Bayes' theorem, which gives us

$$P(Bur|J) = \frac{P(J|Bur)P(Bur)}{P(J)} = \frac{0.849 \cdot 0.001}{0.052} = 0.016.$$

Evidently only about 1.6% of all calls from John are actually due to a break-in. Because the probability of false alarms is five times smaller for Mary, with $P(Bur|M) = 0.056$, we have significantly higher confidence given a call from Mary. Bob should only be seriously concerned about his home if both of them call, because $P(Bur|J, M) = 0.284$ (see Exercise 7.11 on page 173).

In (7.23) on page 162 we showed with

$$P(J|Bur) = P(J|Al)P(Al|Bur) + P(J|\neg Al)P(\neg Al|Bur)$$

how we can "slide in" a new variable. This relationship holds in general for two variables A and B given the introduction of an additional variable C and is called **conditioning:**

$$P(A|B) = \sum_c P(A|B, C = c)P(C = c|B).$$

If furthermore A and B are conditionally independent given C, this formula simplifies to

$$P(A|B) = \sum_c P(A|C = c)P(C = c|B).$$

7.4.5 Software for Bayesian Networks

We will give a brief introduction to two tools using the alarm example. We are already familiar with the system PIT. We input the values from the CPTs in PIT syntax into the online input window at www.pit-systems.de. After the input shown in Fig. 7.12 on page 164 we receive the answer:

```
P([Einbruch=t] | [John=t] AND [Mary=t]) = 0.2841.
```

```
 1 var Alarm{t,f}, Burglary{t,f}, Earthquake{t,f}, John{t,f}, Mary{t,f};
 2
 3 P([Earthquake=t]) = 0.002;
 4 P([Burglary=t]) = 0.001;
 5 P([Alarm=t] | [Burglary=t] AND [Earthquake=t]) = 0.95;
 6 P([Alarm=t] | [Burglary=t] AND [Earthquake=f]) = 0.94;
 7 P([Alarm=t] | [Burglary=f] AND [Earthquake=t]) = 0.29;
 8 P([Alarm=t] | [Burglary=f] AND [Earthquake=f]) = 0.001;
 9 P([John=t] | [Alarm=t]) = 0.90;
10 P([John=t] | [Alarm=f]) = 0.05;
11 P([Mary=t] | [Alarm=t]) = 0.70;
12 P([Mary=t] | [Alarm=f]) = 0.01;
13
14 QP([Burglary=t] | [John=t] AND [Mary=t]);
```

Fig. 7.12 PIT input for the alarm example

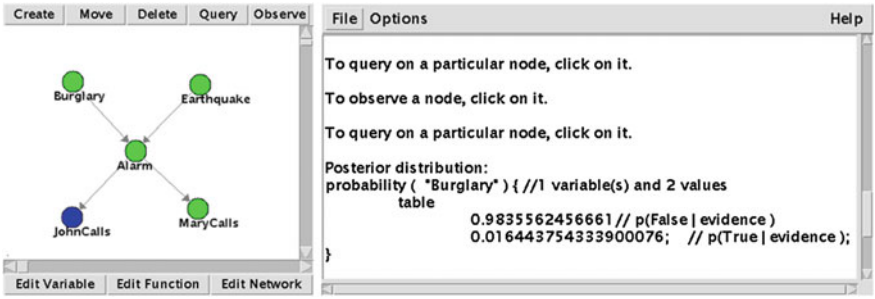


Fig. 7.13 The user interface of JavaBayes: *left* the graphical editor and *right* the console where the answers are given as output

While PIT is not a classical Bayesian network tool, it can take arbitrary conditional probabilities and queries as input and calculate correct results. It can be shown [Sch96], that on input of CPTs or equivalent rules, the MaxEnt principle implies the same conditional independences and thus also the same answers as a Bayesian network. Bayesian networks are thus a special case of MaxEnt.

Next we will look at JavaBayes [Coz98], a classic system also freely available on the Internet with the graphical interface shown in Fig. 7.13. With the graphical network editor, nodes and edges can be manipulated and the values in the CPTs edited. Furthermore, the values of variables can be assigned with "Observe" and the values of other variables called up with "Query". The answers to queries then appear in the console window.

The professional, commercial system Hugin is significantly more powerful and convenient. For example, Hugin can use continuous variables in addition to discrete variables. It can also learn Bayesian networks, that is, generate the network fully automatically from statistical data (see Sect. 8.5).

7.4.6 Development of Bayesian Networks

A compact Bayesian network is very clear and significantly more informative for the reader than a full probability distribution. Furthermore, it requires much less memory. For the variables $v_1, \ldots, v_n$ with $|v_1|, \ldots, |v_n|$ different values each, the distribution has a total of

$$\prod_{i=1}^{n} |v_i| - 1$$

independent entries. In the alarm example the variables are all binary. Thus for all variables $|v_i| = 2$, and the distribution has $2^5 - 1 = 31$ independent entries. To calculate the number of independent entries for the Bayesian network, the total number of all entries of all CPTs must be determined. For a node v_i with k_i parent nodes $e_{i1}, \ldots, e_{ik_i}$, the associated CPT has

$$(|v_i| - 1) \prod_{j=1}^{k_i} |e_{ij}|$$

entries. Then all CPTs in the network together have

$$\sum_{i=1}^{n} (|v_i| - 1) \prod_{j=1}^{k_i} |e_{ij}| \tag{7.24}$$

entries.[18] For the alarm example the result is then

$$2 + 2 + 4 + 1 + 1 = 10$$

independent entries which uniquely describe the network. The comparison in memory complexity between the full distribution and the Bayesian network becomes clearer when we assume that all n variables have the same number b of values and each node has k parent nodes. Then (7.24) can be simplified and all CPTs together have $n(b-1)b^k$ entries. The full distribution contains $b^n - 1$ entries. A significant gain is only made then if the average number of parent nodes is much smaller than the number of variables. This means that the nodes are only locally connected. Because of the local connection, the network becomes modularized, which—as in software engineering—leads to a reduction in complexity. In the alarm example the alarm node separates the nodes *Bur* and *Ear* from the nodes *J* and *M*. We can also see this clearly in the LEXMED example.

[18] For the case of a node without ancestors the product in this sum is empty. For this we substitute the value 1 because the CPT for nodes without ancestors contains, with its a priori probability, exactly one value.

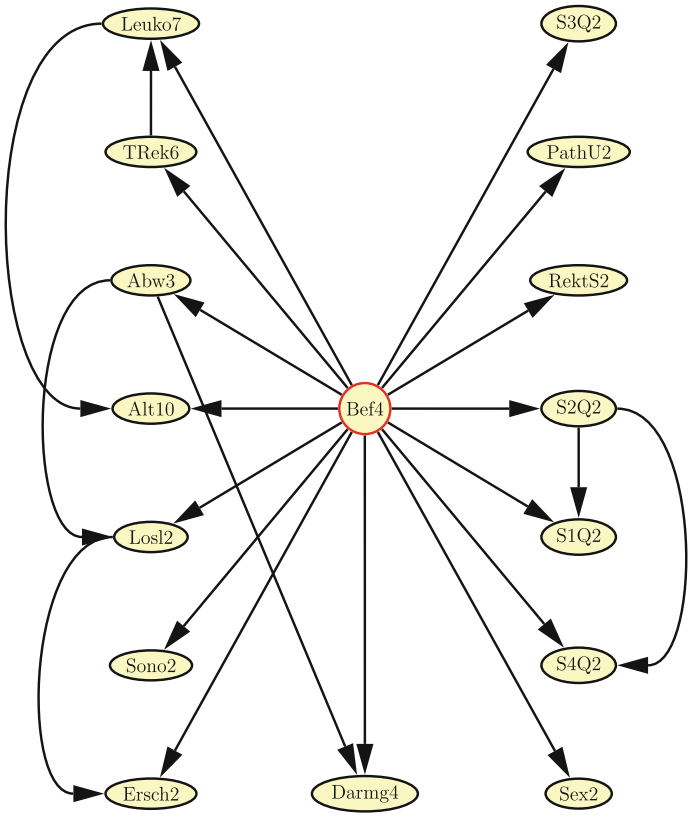


Fig. 7.14 Bayesian network for the LEXMED application

LEXMED as a Bayesian Network

The LEXMED system described in Sect. 7.3 can also be modeled as a Bayesian network. By making the outer, thinly-drawn lines directed (giving them arrows), the independence graph in Fig. 7.8 on page 151 can be interpreted as a Bayesian network. The resulting network is shown in Fig. 7.14.

In Sect. 7.3.2 the size of the distribution for LEXMED was calculated as the value 20 643 839. The Bayesian network on the other hand can be fully described with only 521 values. This value can be determined by entering the variables from Fig. 7.14 into (7.24) on page 165. In the order (Leuko, TRek, Gua, Age, Reb, Sono, Tapp, BowS, Sex, P4Q, P1Q, P2Q, RecP, Urin, P3Q, Diag4) we calculate

$$\begin{aligned}\sum_{i=1}^{n}(|v_i|-1)\prod_{j=1}^{k_i}|e_{ij}| &= 6\cdot 6\cdot 4+5\cdot 4+2\cdot 4+9\cdot 7\cdot 4+1\cdot 3\cdot 4+1\cdot 4+1\cdot 2\cdot 4\\ &\quad +3\cdot 3\cdot 4+1\cdot 4+1\cdot 4\cdot 2+1\cdot 4\cdot 2+1\cdot 4+1\cdot 4+1\cdot 4\\ &\quad +1\cdot 4+1 = 521.\end{aligned}$$

This example demonstrates that it is practically impossible to build a full distribution for real applications. A Bayesian network with 22 edges and 521 probability values on the other hand is still manageable.

Causality and Network Structure

Construction of a Bayesian network usually proceeds in two stages.

1. *Design of the network structure*: This step is usually performed manually and will be described in the following.
2. *Entering the probabilities in the CPTs*: Manually entering the values in the case of many variables is very tedious. If (as for example with LEXMED) a database is available, this step can be automated through estimating the CPT entries by counting frequencies.

We will now describe the construction of the network in the alarm example (see Fig. 7.15). At the beginning we know the two causes *Burglary* and *Earthquake* and the two symptoms *John* and *Mary*. However, because John and Mary do not directly react to a burglar or earthquake, rather only to the alarm, it is appropriate to add this as an additional variable which is not observable by Bob. The process of adding edges starts with the causes, that is, with the variables that have no parent nodes. First we choose *Burglary* and next *Earthquake*. Now we must check whether *Earthquake* is independent of *Burglary*. This is given, and thus no edge is added from *Burglary* to *Earthquake*. Because *Alarm* is directly dependent on *Burglary* and *Earthquake*, these variables are chosen next and an edge is added from both *Burglary* and *Earthquake* to *Alarm*. Then we choose *John*. Because *Alarm* and *John* are not independent, an edge is added from alarm to John. The same is true for *Mary*. Now we must check whether *John* is conditionally independent of *Burglary* given *Alarm*. If this is not the case, then another edge must be inserted from *Burglary* to *John*. We must also check whether edges are needed from *Earthquake* to *John* and from *Burglary* or *Earthquake* to *Mary*. Because of conditional independence, these four edges are not necessary. Edges between *John* and *Mary* are also unnecessary because *John* and *Mary* are conditionally independent given *Alarm*.

The structure of the Bayesian network heavily depends on the chosen variable ordering. If the order of variables is chosen to reflect the causal relationship beginning with the causes and proceeding to the diagnosis variables, then the result will be a simple network. Otherwise the network may contain significantly more edges. Such non-causal networks are often very difficult to understand and have a higher complexity for reasoning. The reader may refer to Exercise 7.11 on page 173 for better understanding.

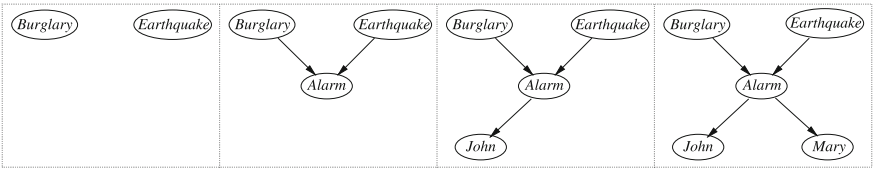


Fig. 7.15 Stepwise construction of the alarm network considering causality

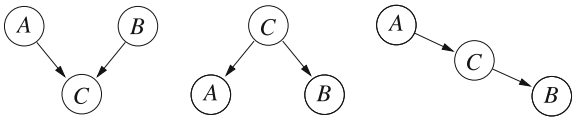


Fig. 7.16 There is no edge between A and B if they are independent (*left*) or conditionally independent (*middle*, *right*)

7.4.7 Semantics of Bayesian Networks

As we have seen in the previous section, no edge is added to a Bayesian network between two variables A and B when A and B are independent or conditionally independent given a third variable C. This situation is represented in Fig. 7.16.

We now require the Bayesian network to have no cycles and we assume that the variables are numbered such that no variable has a lower number than any variable that precedes it. This is always possible when the network has no cycles.[19] Then, when using all conditional independencies, we have

$$\boldsymbol{P}(X_n|X_1,\ldots,X_{n-1}) = \boldsymbol{P}(X_n|\text{Parents}(X_n)).$$

This equation thus is a proposition that an arbitrary variable X_i in a Bayesian network is conditionally independent of its ancestors, given its parents. The somewhat more general proposition depicted in Fig. 7.17 on page 169 can be stated compactly as

Theorem 7.6 *A node in a Bayesian network is conditionally independent from all non-successor nodes, given its parents.*

Now we are able to greatly simplify the chain rule ((7.1) on page 132):

$$\boldsymbol{P}(X_1,\ldots,X_n) = \prod_{i=1}^{n} \boldsymbol{P}(X_i|X_1\ldots,X_{i-1}) = \prod_{i=1}^{n} \boldsymbol{P}(X_i|\text{Parents}(X_i)). \tag{7.25}$$

Using this rule we could, for example, write (7.22) on page 162 directly

$$\boldsymbol{P}(J,Bur,Al) = \boldsymbol{P}(J|Al)\,\boldsymbol{P}(Al|Bur)\,\boldsymbol{P}(Bur).$$

We now know the most important concepts and foundations of Bayesian networks. Let us summarize them [Jen01]:

[19] If for example three nodes X_1, X_2, X_3 form a cycle, then there are the edges (X_1, X_2), (X_2, X_3) and (X_3, X_1) where X_1 has X_3 as a successor.

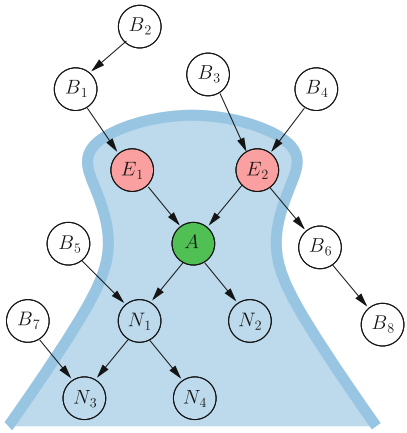


Fig. 7.17 Example of conditional independence in a Bayesian network. If the parent nodes E_1 and E_2 are given, then all non-successor nodes $B_1, \ldots, B_8$ are independent of A

Definition 7.7 A Bayesian network is defined by:

- A set of variables and a set of directed edges between these variables.
- Each variable has finitely many possible values.
- The variables together with the edges form a directed acyclic graph (DAG). A DAG is a graph without cycles, that is, without paths of the form $(A, \ldots, A)$.
- For every variable A the CPT (that is, the table of conditional probabilities $\boldsymbol{P}(A|\text{Parents}(A))$) is given.

Two variables A and B are called *conditionally independent* given C if $\boldsymbol{P}(A, B|C) = \boldsymbol{P}(A|C) \cdot \boldsymbol{P}(B|C)$ or, equivalently, if $\boldsymbol{P}(A|B, C) = \boldsymbol{P}(A|C)$.
Besides the foundational rules of computation for probabilities, the following rules are also true:

Bayes' Theorem: $P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$

Marginalization: $P(B) = P(A, B) + P(\neg A, B) = P(B|A) \cdot P(A) + P(B|\neg A) \cdot P(\neg A)$

Conditioning: $P(A|B) = \sum_c P(A|B,\ C = c)\, P(C = c|B)$

A variable in a Bayesian network is conditionally independent of all non-successor variables given its parent variables. If $X_1, \ldots, X_{n-1}$ are no successors of X_n, we have $\boldsymbol{P}(X_n|X_1, \ldots, X_{n-1}) = \boldsymbol{P}(X_n|\text{Parents}(X_n))$. This condition must be honored during the construction of a network.
During construction of a Bayesian network the variables should be ordered according to causality. First the causes, then the hidden variables, and the diagnosis variables last.

Chain rule: $\boldsymbol{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \boldsymbol{P}(X_i|\text{Parents}(X_i))$

In [Pea88] and [Jen01] the term d-separation is introduced for Bayesian networks, from which a theorem similar to Theorem 7.6 on page 168 follows. We will refrain from introducing this term and thereby reach a somewhat simpler, though not a theoretically as clean representation.

7.5 Summary

In a way that reflects the prolonged, sustained trend toward probabilistic systems, we have introduced probabilistic logic for reasoning with uncertain knowledge. After introducing the language—propositional logic augmented with probabilities or probability intervals—we chose the natural, if unusual approach via the method of maximum entropy as an entry point and showed how we can model non-monotonic reasoning with this method. Bayesian networks were then introduced as a special case of the MaxEnt method.

Why are Bayesian networks a special case of MaxEnt? When building a Bayesian network, assumptions about independence are made which are unnecessary for the MaxEnt method. Furthermore, when building a Bayesian network, all CPTs must be completely filled in so that a complete probability distribution can be constructed. Otherwise reasoning is restricted or impossible. With MaxEnt, on the other hand, the developer can formulate all the knowledge he has at his disposal in the form of probabilities. MaxEnt then completes the model and generates the distribution. Even if (for example when interviewing an expert) only vague propositions are available, this can be suitably modeled. A proposition such as "I am pretty sure that A is true." can for example be modeled using $P(A) \in [0.6, 1]$ as a probability interval. When building a Bayesian network, a concrete value must be given for the probability, if necessary by guessing. This means, however, that the expert or the developer put ad hoc information into the system. One further advantage of MaxEnt is the possibility of formulating (almost) arbitrary propositions. For Bayesian networks the CPTs must be filled.

The freedom that the developer has when modeling with MaxEnt can be a disadvantage (especially for a beginner) because, in contrast to the Bayesian approach, it is not necessarily clear what knowledge should be modeled. When modeling with Bayesian networks the approach is quite clear: according to causal dependencies, from the causes to the effects, one edge after the other is entered into the network by testing conditional independence.[20] At the end all CPTs are filled with values.

However, the following interesting combinations of the two methods are possible: we begin by building a network according to the Bayesian methodology, enter all the edges accordingly and then fill the CPTs with values. Should certain values for the CPTs be unavailable, then they can be replaced with intervals or by other probabilistic logic formulas. Naturally such a network—or better: a rule

[20] This is also not always quite so simple.

set—no longer has the special semantics of a Bayesian network. It must then be processed and completed by a MaxEnt system.

The ability to use MaxEnt with arbitrary rule sets has a downside, though. Similarly to the situation in logic, such rule sets can be inconsistent. For example, the two rules $P(A) = 0.7$ and $P(A) = 0.8$ are inconsistent. While the MaxEnt system PIT for example can recognize the inconsistency, if cannot give a hint about how to remove the problem.

We introduced the medical expert system LEXMED, a classic application for reasoning with uncertain knowledge, and showed how it can be modeled and implemented using MaxEnt and Bayesian networks, and how these tools can replace the well-established, but too weak linear scoring systems used in medicine.[21]

In the LEXMED example we showed that it is possible to build an expert system for reasoning under uncertainty that is capable of discovering (learning) knowledge from the data in a database. We will give more insight into the methods of learning of Bayesian networks in Chap. 8, after the necessary foundations for machine learning have been laid.

Today Bayesian reasoning is a large, independent field, which we can only briefly describe here. We have completely left out the handling of continuous variables. For the case of normally distributed random variables there are procedures and systems. For arbitrary distributions, however, the computational complexity is a big problem. In addition to the directed networks that are heavily based on causality, there are also undirected networks. Connected with this is a discussion about the meaning and usefulness of causality in Bayesian networks. The interested reader is directed to excellent textbooks such as [Pea88, Jen01, Whi96, DHS01], as well as the proceedings of the annual conference of the *Association for Uncertainty in Artificial Intelligence* (*AUAI*) (www.auai.org).

7.6 Exercises

Exercise 7.1 Prove the proposition from Theorem 7.1 on page 129.

Exercise 7.2 The gardening hobbyist Max wants to statistically analyze his yearly harvest of peas. For every pea pod he picks he measures its length x_i in centimeters and its weight y_i in grams. He divides the peas into two classes, the good and the bad (empty pods). The measured data (x_i, y_i) are

good peas:

x	1	2	2	3	3	4	4	5	6
y	2	3	4	4	5	5	6	6	6

bad peas:

x	4	6	6	7
y	2	2	3	3

[21] In Sect. 8.7 and in Exercise 8.17 on page 242 we will show that the scores are equivalent to the special case naive Bayes, that is, to the assumption that all symptoms are conditionally independent given the diagnosis.

(a) From the data, compute the probabilities $P(y > 3|Class = good)$ and $P(y \leq 3|Class = good)$. Then use Bayes' formula to determine $P(Class = good|y > 3)$ and $P(Class = good|y \leq 3)$.
(b) Which of the probabilities computed in subproblem (a) contradicts the statement "All good peas are heavier than 3 grams"?

Exercise 7.3 You are supposed to predict the afternoon weather using a few simple weather values from the morning of this day. The classical probability calculation for this requires a complete model, which is given in the following table.

Sky	*Bar*	*Prec*	$P(Sky, Bar, Prec)$
Clear	Rising	Dry	0.40
Clear	Rising	Raining	0.07
Clear	Falling	Dry	0.08
Clear	Falling	Raining	0.10
Cloudy	Rising	Dry	0.09
Cloudy	Rising	Raining	0.11
Cloudy	Falling	Dry	0.03

Sky: The sky is clear or cloudy in the morning
Bar: Barometer rising or falling in the morning
Prec: Raining or dry in the afternoon

(a) How many events are in the distribution for these three variables?
(b) Compute $P(Prec = \text{dry}|Sky = \text{clear}, Bar = \text{rising})$.
(c) Compute $P(Prec = \text{rain}|Sky = \text{cloudy})$.
(d) What would you do if the last row were missing from the table?

❄ **Exercise 7.4** In a television quiz show, the contestant must choose between three closed doors. Behind one door the prize awaits: a car. Behind both of the other doors are goats. The contestant chooses a door, e.g. number one. The host, who knows where the car is, opens another door, e.g. number three, and a goat appears. The contestant is now given the opportunity to choose between the two remaining doors (one and two). What is the better choice from his point of view? To stay with the door he originally chose or to switch to the other closed door?

Exercise 7.5 Using the Lagrange multiplier method, show that, without explicit constraints, the uniform distribution $p_1 = p_2 = \cdots = p_n = 1/n$ represents maximum entropy. Do not forget the implicitly ever-present constraint $p_1 + p_2 + \cdots + p_n = 1$. How can we show this same result using indifference?

Exercise 7.6 Use the PIT system (http://www.pit-systems.de) or SPIRIT (http://www.xspirit.de) to calculate the MaxEnt solution for $P(B)$ under the constraint $P(A) = \alpha$ and $P(B|A) = \beta$. Which disadvantage of PIT, compared with calculation by hand, do you notice here?

Exercise 7.7 Given the constraints $P(A) = \alpha$ and $P(A \vee B) = \beta$, manually calculate $P(B)$ using the MaxEnt method. Use PIT to check your solution.

❄ **Exercise 7.8** Given the constraints from (7.10), (7.11), (7.12): $p_1 + p_2 = \alpha$, $p_1 + p_3 = \gamma$, $p_1 + p_2 + p_3 + p_4 = 1$. Show that $p_1 = \alpha\gamma$, $p_2 = \alpha(1-\gamma)$, $p_3 = \gamma(1-\alpha)$, $p_4 = (1-\alpha)(1-\gamma)$ represents the entropy maximum under these constraints.

❄ **Exercise 7.9** A probabilistic algorithm calculates the likelihood p that an inbound email is spam. To classify the emails in classes *delete* and *read*, a cost matrix is then applied to the result.

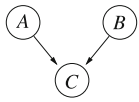
(a) Give a cost matrix (2×2 matrix) for the spam filter. Assume here that it costs the user 10 cents to delete an email, while the loss of an email costs 10 dollars (compare this to Example 1.1 on page 17 and Exercise 1.7 on page 21).

(b) Show that, for the case of a 2×2 matrix, the application of the cost matrix is equivalent to the application of a threshold on the spam probability and determine the threshold.

Exercise 7.10 Given a Bayesian network with the three binary variables A, B, C and $P(A) = 0.2$, $P(B) = 0.9$, as well as the CPT shown below:

(a) Compute $P(A|B)$.
(b) Compute $P(C|A)$.

A	B	$P(C)$
t	f	0.1
t	t	0.2
f	t	0.9
f	f	0.4



Exercise 7.11 For the alarm example (Example 7.10 on page 159), calculate the following conditional probabilities:

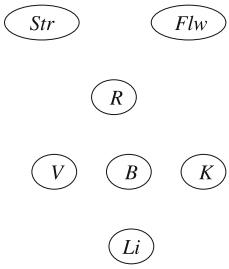
(a) Calculate the a priori probabilities $P(Al)$, $P(J)$, $P(M)$.
(b) Calculate $P(M|Bur)$ using the product rule, marginalization, the chain rule, and conditional independence.
(c) Use Bayes' formula to calculate $P(Bur|M)$
(d) Compute $P(Al|J, M)$ and $P(Bur|J, M)$.
(e) Show that the variables J and M are not independent.
(f) Check all of your results with JavaBayes and with PIT (see [Ert11] for demo programs).
(g) Design a Bayesian network for the alarm example, but with the altered variable ordering M, Al, Ear, Bur, J. According to the semantics of Bayesian networks, only the necessary edges must be drawn in. (Hint: the variable order given here does NOT represent causality. Thus it will be difficult to intuitively determine conditional independences.)
(h) In the original Bayesian network of the alarm example, the earthquake nodes is removed. Which CPTs does this change? (Why these in particular?)
(i) Calculate the CPT of the alarm node in the new network.

Exercise 7.12 A diagnostic system is to be made for a dynamo-powered bicycle light using a Bayesian network. The variables in the following table are given.

Abbr.	Meaning	*Values*
Li	Light is on	*t*/*f*
Str	Street condition	*dry, wet, snow_covered*
Flw	Dynamo flywheel worn out	*t*/*f*
R	Dynamo sliding	*t*/*f*
V	Dynamo shows voltage	*t*/*f*
B	Light bulb o.k.	*t*/*f*
K	Cable o.k.	*t*/*f*

The following variables are pairwise independent: *Str*, *Flw*, *B*, *K*. Furthermore: (*R*, *B*), (*R*, *K*), (*V*, *B*), (*V*, *K*) are independent and the following equation holds:

$$P(Li|V,R) = P(Li|V)$$
$$P(V|R,Str) = P(V|R)$$
$$P(V|R,Flw) = P(V|R)$$



(a) Draw all of the edges into the graph (taking causality into account).
(b) Enter all missing CPTs into the graph (table of conditional probabilities). Freely insert plausible values for the probabilities.
(c) Show that the network does not contain an edge (*Str*, *Li*).
(d) Compute $P(V|Str = snow\_covered)$.

V	*B*	*K*	*P*(*Li*)
t	*t*	*t*	0.99
t	*t*	*f*	0.01
t	*f*	*t*	0.01
t	*f*	*f*	0.001
f	*t*	*t*	0.3
f	*t*	*f*	0.005
f	*f*	*t*	0.005
f	*f*	*f*	0