Chapter 12
Reasoning with Imperfect Knowledge

If we reason about propositions in AI systems which are based on classic logic, we use only two possible logic values, i.e., *true* and *false*. However, in the case of reasoning about the real (physical) world such a two-valued evaluation is inadequate, because of the aspect of uncertainty. There are two sources of this problem: *imperfection of knowledge* about the real world which is gained by the system and *vagueness of notions* used for describing objects/phenomena of the real world.

We discuss models which are applied for solving the problem of *imperfect knowledge* in this chapter.[1] There are three aspects of the imperfection of knowledge: *uncertainty of knowledge* (information can be uncertain), *imprecision of knowledge* (measurements of signals received by the AI system can be imprecise) and *incompleteness of knowledge* (the system does not know all required facts).

In the first section the model of *Bayesian inference* based on a probability measure is introduced. This measure is used to express our uncertainty concerning knowledge, not for assessing the degree of truthfulness of propositions. *Dempster-Shafer theory*, which allows us to express a lack of complete knowledge, i.e., our ignorance, with specific measures is considered in the second section. Various models of non-monotonic reasoning can also be applied for solving the problem of incompleteness of knowledge. Three such models, namely *default logic*, *autoepistemic logic*, and *circumscription reasoning* are discussed in the third section.

12.1 Bayesian Inference and Bayes Networks

In Sect. 10.2 we have discussed the use of the Bayesian *probability a posteriori* model[2] for constructing a classifier in statistical pattern recognition. In this section we interpret notions of the Bayesian model in a different way, in another application context.

[1] Models applied for solving a problem of vague notions are introduced in the next chapter.

[2] Mathematical foundations of probabilistic reasoning are introduced in Appendix I.

M. Flasiński, *Introduction to Artificial Intelligence*,
DOI 10.1007/978-3-319-40022-8_12

Let e be an observation of some event (situation, behavior, symptom, etc.).[3] Let $h_1, h_2, \ldots, h_n$ be various (distinct) hypotheses which can explain the occurrence of the observation e. Let us consider a hypothesis h_k with an a priori *probability* (i.e., without knowledge concerning the observation e) of $P(h_k)$. Let us assume that the probability of an occurrence of the observation e assuming the truthfulness of the hypothesis h_k, i.e., the *conditional probability* $P(e|h_k)$, is known. Then, the *a posteriori probability*, i.e., the probability of the hypothesis h_k assuming an occurrence of the observation e, is defined by the following formula:

$$P(h_k|e) = \frac{P(e|h_k) \cdot P(h_k)}{P(e)}, \tag{12.1}$$

where $P(e)$ is the probability of an occurrence of the observation e given hypotheses $h_1, h_2, \ldots, h_n$. The probability $P(e)$ is computed according to the following formula:

$$P(e) = \sum_{i=1}^{n} P(e|h_i) \cdot P(h_i). \tag{12.2}$$

Let us analyze this model with the help of the following example.[4] Let us assume that we would like to diagnose a patient *John Smith*. Then $h_1, h_2, \ldots, h_n$ denote possible disease entities.[5] We assume that the bird flu, denoted with h_p, is spreading throughout our country. The a priori probability of going down with the bird flu can be evaluated as the percentage of our countrymen who have the bird flu.[6] Then, let an observation e mean the patient has a temperature which is more than 39.5 °C. The probability that a patient having the bird flu has a temperature above 39.5 °C, i.e., $P(e|h_p)$, can be evaluated as the percentage of our countrymen having the bird flu who also have a temperature which is more than 39.5 °C.

Now, we can diagnose *John Smith*. If he has a temperature higher than 39.5 °C, i.e., we observe an occurrence of the symptom e, then the probability that he has gone down with the bird flu, $P(h_p|e)$, can be computed with the help of formula (12.1).[7]

Of course, making a hypothesis on the basis of one symptom (one observation) is not sound. Therefore, we can extend our formulas to the case of m observations $e_1, e_2, \ldots, e_m$. If we assume that the observations are conditionally independent

[3] Such an observation is represented as a *fact* in a knowledge base.

[4] Of course, all examples are simplified.

[5] Strictly speaking, h_i means making a diagnosis (hypothesis) that the patient has the disease entity denoted by an index i.

[6] Let us notice that it is really an a priori probability in the case of diagnosing *John Smith*, because for such an evaluation we do not take into account any symptoms/factors concerning him.

[7] Let us notice that in order to use $P(e)$ in formula (12.1) we have to compute this probability with formula (12.2). Thus, a priori probabilities $h_1, h_2, \ldots, h_n$ should be known for all disease entities. We should also know the probabilities that a patient having an illness denoted by an index $i = 1, 2, \ldots, n$ has a temperature which is more than 39.5 °C, i.e., $P(e|h_i)$. We are able to evaluate these probabilities if we have corresponding statistical data.

given each hypothesis $h_i, i = 1, \ldots, n$,[8] then we obtain the following formula for the probability of a hypothesis h_k given observations $e_1, e_2, \ldots, e_m$:

$$P(h_k|e_1, e_2, \ldots, e_m) = \frac{P(e_1|h_k) \cdot P(e_2|h_k) \cdot \cdots \cdot P(e_m|h_k) \cdot P(h_k)}{P(e_1, e_2, \ldots, e_m)}, \qquad (12.3)$$

where $P(e_1, e_2, \ldots, e_m)$ is the probability of observations $e_1, e_2, \ldots, e_m$ occurring given hypotheses $h_1, h_2, \ldots, h_n$. This probability is computed according to the following formula:

$$P(e_1, e_2, \ldots, e_m) = \sum_{i=1}^{n} P(e_1|h_i) \cdot P(e_2|h_i) \cdot \cdots \cdot P(e_m|h_i) \cdot P(h_i). \qquad (12.4)$$

Summing up, the Bayesian model allows us to compute the probability of the truthfulness of a given hypothesis on the basis of observations/facts which are stored in the knowledge base. We will come back to this model at the end of this section, when we present Bayes networks. First, however, we introduce basic notions of probabilistic reasoning.

In probabilistic reasoning a problem domain is represented by a set of *random variables*.[9] For example, in medical diagnosis random variables can represent symptoms (e.g., *a body temperature*, *a runny nose*), disease entities (e.g., *hepatitis*, *lung cancer*), risk factors (*smoking*, *excess alcohol*), etc.

For each random variable its domain (i.e., a set of events for which it is defined) is determined. For example, in the case of car diagnosis for a variable *Engine failure cause* we can determine its domain in the following way:

$$Engine\ failure\ cause\colon \langle piston\ seizing\ up,\ timing\ gear\ failure,\ starter\ failure,\ exhaust\ train\ failure,\ broken\ inlet\ valve\rangle.$$

Random variables are often *logic (Boolean) variables*, i.e., they take either the value *1* (*true* (T)) or *0* (*false (F)*). Sometimes we write *smoking* in case this variable takes the value 1 and we write ¬ *smoking* otherwise.

For a random variable which describes a problem domain we define its *distribution*. The distribution determines the probabilities that the variable takes specific values. For example, assuming that a random variable *Engine failure cause* takes values: *1*, *2*, …, *5* for the events listed above, we can represent its distribution with the help of the one-dimensional table shown in Fig. 12.1a.[10] Let us notice that the probabilities should add up to 1.0.

[8] This means that $P(e_1, e_2, \ldots, e_m|h_i) = P(e_1|h_i) \cdot P(e_2|h_i) \cdot \cdots \cdot P(e_m|h_i)$.

[9] In this chapter we consider discrete random variables. Formal definitions of a random variable, a random vector, and distributions are contained in Appendix B.1.

[10] In the first column of the table elementary events are placed. For each elementary event the value which is taken by the random variable is also defined.

(a)

Engine failure cause	P(Engine failure cause)
1 (piston seizing up)	**0.05**
2 (timing gear failure)	**0.1**
3 (starter failure)	**0.4**
4 (exhaust train failure)	**0.3**
5 (broken inlet valve)	**0.15**

(b)

	Unstable gas flow	¬ Unstable gas flow
Gas supply subsystem failure	**0.008**	**0.01**
¬ Gas supply subsystem failure	**0.002**	**0.98**

X_2

X_1

(c)

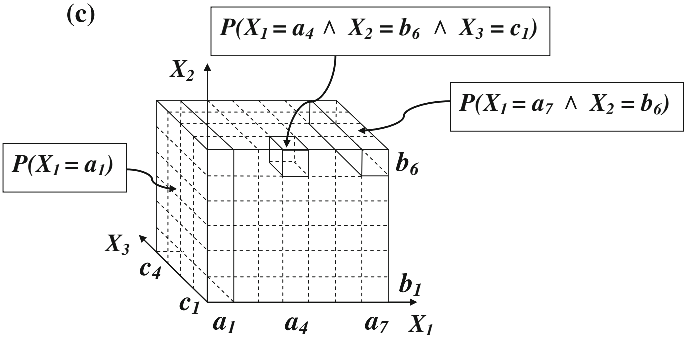

$P(X_1 = a_4 \wedge X_2 = b_6 \wedge X_3 = c_1)$
X_2
$P(X_1 = a_7 \wedge X_2 = b_6)$
$P(X_1 = a_1)$
b_6
X_3
c_4
c_1
b_1
a_1
a_4
a_7
X_1

Fig. 12.1 **a** An example of the distribution of a random variable, **b** an example of the distribution of a two-dimensional random vector, **c** the scheme of the table of the joint probability distribution

In the general case, if there are n random variables $X_1, X_2, \dots X_n$, which describe a problem domain, they create a *random vector* $(X_1, X_2, \dots X_n)$. In such a case we define the *distribution of a random vector*. This determines all the possible combinations of values that can be assigned to all variables. The distribution of a random vector $(X_1, X_2, \dots X_n)$ is called the *joint probability distribution, JPD*, of random variables $X_1, X_2, \dots X_n$.

In the case of two discrete random variables X_1 and X_2 taking values from domains which have m_1 and m_2 elements respectively, their joint probability distribution can be represented by a two-dimensional $m_1 \times m_2$ table $P = [p_{ij}], i = 1, \dots, m_1,$ $j = 1, \dots, m_2$. An element p_{ij} of the table determines the probability that the variable X_1 takes the value i and the variable X_2 takes the value j. For example, if there are two logical variables, *Unstable gas flow* and *Gas supply subsystem failure*, then their joint probability distribution can be represented as shown in Fig. 12.1b. Then, for example

$$P(Unstable\, gas\ flow \wedge \neg Gas\ supply\, subsystem\ failure) = 0.002.$$

Similarly to the one-dimensional case, the probabilities of all cells of the table should add up to 1.0. Let us notice that we can determine probabilities not only for *complete* propositions which concern a problem domain, i.e., for propositions which include all variables of a random vector with values assigned. We can also determine probabilities for propositions containing only some of the variables. For example, we can compute the probability of the proposition ¬ *Unstable gas flow* by adding the probabilities of the second column of the table, i.e., we can sum the probabilities for all the values which are taken by the other variable *Gas supply subsystem failure*[11]:

$$P(\neg Unstable\, gas\ flow) = 0.01 + 0.98 = 0.99.$$

Returning to an n-dimensional random vector $(X_1, X_2, \ldots X_n)$, the joint probability distribution of its variables is represented by an n-dimensional table. For example, the scheme of such a table is shown in Fig. 12.1c. As we can see, the variables take values $X_1 = a_1, a_2, \ldots, a_7$, $X_2 = b_1, b_2, \ldots, b_6$, $X_3 = c_1, c_2, c_3, c_4$. Each elementary cell of the table contains the probability for the proposition including all the variables. Thus, for example for the proposition $X_1 = a_4 \wedge X_2 = b_6 \wedge X_3 = c_1$ the probability included in the elementary cell defined by the given coordinates of the table is determined. The probability of the proposition $X_1 = a_7 \wedge X_2 = b_6$ is computed by adding the probabilities included in the cells which belong to the rightmost upper "beam". (It is defined according to the marginal distribution for variables X_1 and X_2, whereas X_3 takes any values.) For example, the probability of the proposition $X_1 = a_1$ is computed by adding the probabilities included in the cells which belong to the leftmost "wall". (It is defined according to the marginal distribution for the variable X_1, whereas X_2 and X_3 take any values.)

There are two disadvantages of using the table of the joint probability distribution. Firstly, we should be able to evaluate all values of a random vector distribution. This is very difficult and sometimes impossible in practice. Secondly, it is inefficient, since in practical applications we have hundreds of variables and each variable can take thousands of values. Thus, the number of cells of the table of the joint probability

[11] Let us assume that a random vector (X_1, X_2) is given, where X_1 takes values $a_1, \ldots, a_{m1}$ and X_2 takes values $b_1, \ldots, b_{m2}$. If we are interested only in the distribution of one variable and the other variable can take any values, then we talk about the *marginal distribution* of the first variable. Then, for example the marginal distribution of the variable X_1 is determined in the following way: $P(X_1 = a_i) = P(X_1 = a_i, X_2 = b_1) + \cdots + P(X_1 = a_i, X_2 = b_{m2}),\ i = 1, \ldots, m_1$.
The marginal distribution for the second variable X_2 is determined in an analogous way. For an n-dimensional random vector we can determine the marginal distribution for any subset of variables, assuming that the remaining variables take any values.

distribution can be huge. For example, if there are n variables and each variable can take k values on average, then the number of cells is k^n. Now, we can come back to the Bayesian model, which inspired Pearl [222] to define a method which allows probabilistic reasoning without using the joint probability distribution.

The Pearl method is based on a graph representation called a *Bayes network*. A Bayes network is a directed acyclic graph.[12] Nodes of the graph correspond to random variables, which describe a problem domain. Edges of the graph represent a direct dependency between variables. If an edge goes from a node labeled by X_1 to a node labeled by X_2, then a direct cause-effect relation holds for the variable X_1 (a direct cause) and the variable X_2 (an effect). We say the node labeled by X_1 is the *predecessor* of the node labeled by X_2. Further on, the node labeled by X is equated with the random variable X.

In a Bayes network, for each node which has predecessors we define a table showing an influence of the predecessors on this node. Let a node X_i have p predecessors, $X_{i1}, \ldots, X_{ip}$. Then, the conditional probabilities of all the possible values taken by the variable X_i depending on all possible values of the variables $X_{i1}, \ldots, X_{ip}$ are determined by the table. For example, let a node X_3 have two predecessors X_1 and X_2. Let these variables take values as follows: $X_1 = a_1, \ldots, a_{m1}$, $X_2 = b_1, \ldots, b_{m2}$, $X_3 = c_1, \ldots, c_{m3}$. Then, the table defined for the node X_1 is of the following form ($p_{(i)(j)}$ denotes the corresponding probability):

X_1	X_2	$P(X_3 \mid X_1, X_2)$		
		c_1	$\ldots$	c_{m3}
a_1	b_1	$p_{(1)(1)}$	$\ldots$	$p_{(1)(m3)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_1	b_{m2}	$\ldots$	$\ldots$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_{m1}	b_1	$\ldots$	$\ldots$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_{m1}	b_{m2}	$p_{(m1 \cdot m2)(1)}$	$\ldots$	$p_{(m1 \cdot m2)(m3)}$

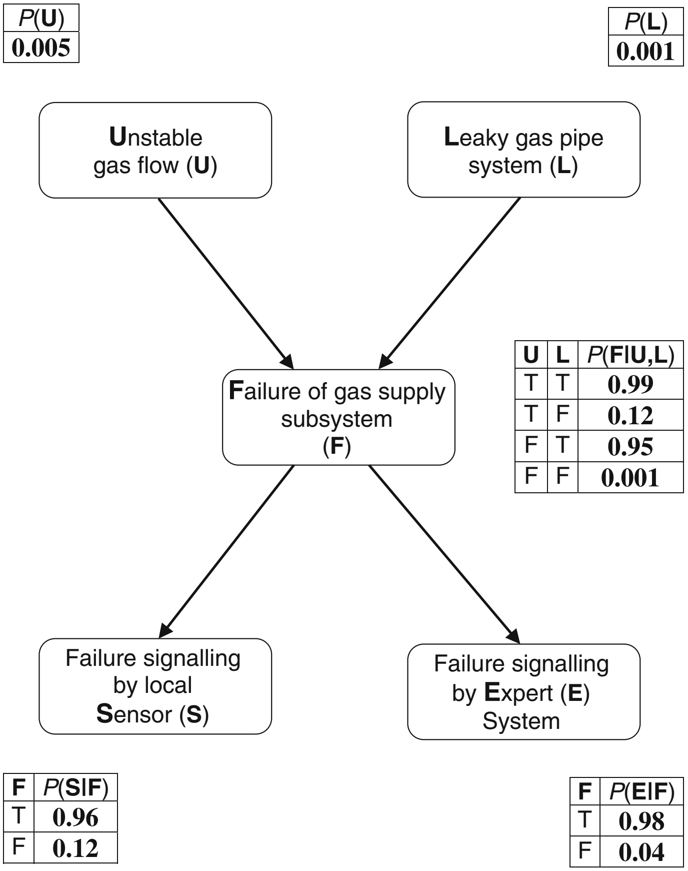
Let us notice that the values of all probabilities in any row of the table should add up to 1.0. If a node X has no predecessors, then we define the table of the distribution of the random variable X as we have done for the example table shown in Fig. 12.1a.

We consider an example of a Bayes network for logical variables.[13] Let us notice that if variables X_1, X_2 in the table above are logical, then there are only four combinations of (logic) values, i.e., *1-1* (i.e. *True-True*), *1-0*, *0-1*, *0-0*.[14] If the variable X_3 is also a logical variable, we can write its value only if it is *True*, because we can

[12] That is, there are no directed cycles in the graph.

[13] In order to simplify our considerations, without loss of generality of principles.

[14] In our examples they are denoted *T-T*, *T-F*, *F-T*, *F-F*, respectively.


Unstable gas flow (U)
Leaky gas pipe system (L)
Failure of gas supply subsystem (F)
Failure signalling by local Sensor (S)
Failure signalling by Expert (E) System

P(U)
0.005

P(L)
0.001

U	L	*P*(F|U,L)
T	T	0.99
T	F	0.12
F	T	0.95
F	F	0.001

F	*P*(S|F)
T	0.96
F	0.12

F	*P*(E|F)
T	0.98
F	0.04

Fig. 12.2 An example of a Bayes network

compute the value corresponding to *False*, taking into account the fact that the two values should add up to 1.0.

An example of a Bayes network defined for diagnosing a gas supply subsystem is shown in Fig. 12.2. The two upper nodes of the network represent the logic random variables *Unstable gas flow* and *Leaky gas pipe system* and they correspond to possible causes of a failure. Tables of distributions of these variables contain probabilities of the causes occurring. For example, the probability of a leaky gas pipe system $P(L)$ equals 0.001. The table for the variable L determines the whole distribution, because the probability that leaking does not occur $P(\neg L)$ is defined in an *implicit* way as the complement of $P(L)$.[15]

Each of the causes U and L can result in *Failure of gas supply subsystem (F)*, which is represented by edges of the network. The edges denote the *direct* dependency of the

[15] We can compute it as follows: $P(\neg L) = 1.0 - P(L) = 1.0 - 0.001 = 0.999$.

variable *F* on variables *U* and *L*. For the node *F* we define a table which determines the conditional probabilities of all possible values taken by the variable *F*,[16] depending on all value assignments to variables *U* and *L*. For example, the probability of the failure *F*, if there is an unstable gas flow and the gas pipe system is not leaking equals 0.12.

As we can see, the failure *F* can be signalled by a *local Sensor (S)* or, independently, by an *Expert (E) System*. For example, the table for the *local Sensor (S)* can be interpreted in the following way:

- the probability that the sensor signals a failure, if the failure occurs, $P(S|F)$, equals 0.96,
- the probability that the sensor signals a failure, if a failure does not occur, (i.e., it signals improperly), $P(S \mid\neg F)$, equals 0.12,
- the probability that the sensor does not signal a failure, if a failure occurs, $P(\neg S \mid F)$, equals $1.0 - 0.96 = 0.04$,
- the probability that the sensor does not signal a failure, if a failure does not occur, $P(\neg S \mid\neg F)$, equals $1.0 - 0.12 = 0.88$.

A Bayes network allows us to assign probabilities to propositions defined with the help of random variables which describe a problem domain according to the following formula:

$$P(X_1, \ldots X_n) = P(X_n|Predecessors(X_n)) \cdot P(X_{n-1}|Predecessors(X_{n-1})) \cdot \cdots \\ \cdots \cdot P(X_2|Predecessors(X_2)) \cdot P(X_1|Predecessors(X_1)), \tag{12.5}$$

where $Predecessors(X_i)$ denotes all the nodes of the Bayes network which are direct predecessors of the node X_i. If the node X_i has no predecessors, then $P(X_k|Predecessors(X_k)) = P(X_k)$.

Formula (12.5) says that if we want to compute the probability of a proposition defined with variables $X_1, \ldots, X_n$, then we should multiply conditional probabilities representing dependency of $X_i, i = 1, \ldots, n$, only for those variables which influence X_i *directly*.

For example, if we want to compute the probability that neither the local sensor nor the expert system signals the failure in case there is an unstable gas flow and the gas pipe system is no leaking, i.e.,

$$U\ , \neg L\ , F\ , \neg S\ , \neg E\ ,$$

then we compute it according to formula (12.5) and the network shown in Fig. 12.2 as follows:

[16] The variable *F* is a logical variable. Therefore, it is sufficient to determine the probabilities when *F* equals *True*. Probabilities for the value *False* are complements of these probabilities.

$$\begin{aligned} &P(U, \neg L, F, \neg S, \neg E) \\ &\qquad = P(U) \cdot P(\neg L) \cdot P(F|U, \neg L) \cdot P(\neg S|F) \cdot P(\neg E|F) \\ &\qquad = 0.005 \cdot 0.999 \cdot 0.12 \cdot 0.04 \cdot 0.02 = 0.00000047952. \end{aligned}$$

Finally, let us consider the main idea of constructing a Bayes network that allows us to use formula (12.5) for simplified probabilistic reasoning without using the joint probability distribution. Let network nodes be labeled by variables $X_1, \ldots, X_n$ in such a way that for a given node its predecessors have a lower index. In fact, using formula (12.5), we assume that the event represented by the variable X_i is conditionally independent[17] from earlier events[18] which are *not* its *direct* predecessors, assuming the events represented by its *direct* predecessors[19] have occurred. This means that we should define the structure of the network in accordance with this assumption if we want to make use of formula (12.5). In other words, if we add a node X_i to the Bayes network, we should connect it with all the nodes among $X_1, \ldots, X_{i-1}$ which influence it *directly* (and only with such nodes). Therefore, Bayes networks should be defined in strict cooperation with domain (human) experts.

12.2 Dempster-Shafer Theory

As we have shown in the previous section, Bayes networks allow AI systems to reason in a more efficient way than the standard models of probability theory. Apart from the issue of the efficiency of inference based on imperfect knowledge, the problem of incompleteness of knowledge makes the construction of a reasoning system difficult. In such a situation, we do not know all required facts and we suspect that the lack of complete information influences the quality of the reasoning process. Then, the problem of *expressing lack of knowledge* arises, since we should be able to differentiate between uncertainty concerning knowledge possessed and *our ignorance* (i.e., our awareness of the lack of some knowledge). This problem was noticed by Arthur P. Dempster.[20] To solve it he proposed a model based on the concept of *lower* and *upper probability* in the late 1960s [67]. This model was then developed by Glenn Shafer[21] in 1976 [271]. Today the model is known as *Dempster-Shafer Theory*, *belief function theory*, or the *mathematical theory of evidence*.[22]

[17] Conditional independence of variables is defined formally by Definition I.10 in Appendix I.

[18] *Earlier* in the sense of indexing nodes of the network.

[19] We have denoted such predecessors by $Predecessors(X_i)$.

[20] Arthur Pentland Dempster—a professor of statistics at Harvard University. John W. Tukey (the Cooley-Tukey algorithm for Fast Fourier Transforms) was an adviser of his Ph.D. thesis. His work concerns the theory introduced in this section, cluster analysis, and image processing (the *EM* algorithm).

[21] Glenn Shafer—a professor of statistics at Rutgers University. Apart from the development of DST, he proposed a new approach to probability theory based on game theory (instead of measure theory).

[22] In the context of reasoning with incomplete knowledge, *evidence* means information gained by an AI system at some moment which is used as a *premise of inference*.

A complete specification of the probability model is required in the Bayesian approach. On the contrary, in Dempster-Shafer Theory a model can be specified in an incomplete way. The second difference concerns the interpretation of the notion of *probability*[23] and, in consequence, a different way of computing it in a reasoning model. In the Bayesian approach we try to compute the probability that a given proposition (hypothesis) is true. On the contrary, in DST we try to compute the probability saying how available information, which creates the premises of our reasoning, supports our *belief* about the truthfulness of a given proposition (hypothesis). A "probability" interpreted in such a way is measured with the help of a *belief function*, usually denoted Bel.[24]

For example, let us assume that I have found *The Assayer* by Galileo in an unknown antique shop in Rome. I would like to buy it, because I like old books. On the other hand, I am not an expert. So I do not know whether it is genuine. In other words, I do not possess any information concerning the book. In such a case we should define a belief function Bel in the following way according to Dempster-Shafer Theory[25]:

$$Bel(genuine) = 0 \text{ and } Bel(\neg genuine) = 0.$$

Fortunately, I have recalled that my friend Mario, who lives in Rome, is an expert in old books. Moreover, he has a special device which allows him to perform tests. So I have phoned him and I have asked him to help me. Mario has arrived. He has brought two devices. The first one has been made to confirm the authenticity of old books according to certain criteria. The second one has been made to question the authenticity of old books according to other criteria. After taking measurements of the book, he has told me that he believes with a 0.9 degree of certainty that the book is genuine as indicated by the first device. On the other hand, he believes with 0.01 degree of certainty that the book is fake as indicated by the second device. This time the belief function Bel should be computed in the following way:

$$Bel(genuine) = 0.9 \text{ and } Bel(\neg genuine) = 0.01.$$

So I have bought the book.

According to the Dempster-Shafer approach, the belief function Bel is a *lower probability*. The *upper probability* is called a *plausibility function* Pl, which for a proposition S is defined as follows:

$$Pl(S) = 1 - Bel(\neg S).$$

[23] We mean an intuitive interpretation of this notion, not in the sense of probability theory.

[24] Basic definitions of Dempster-Shafer Theory are included in Appendix I.3.

[25] Let us notice that a probability measure P has the following property: $P(\neg genuine) = 1 - P(genuine)$. This property does not hold for a belief function Bel.

Thus, the plausibility function says how strong the evidence is against the proposition S.[26] Coming back to our example, we can compute the plausibility function for *genuine* in the following way:

$$Pl(genuine) = 1 - Bel(\neg genuine) = 1 - 0.01 = 0.99.$$

Summing up, in Dempster-Shafer Theory we define two probability measures Bel and Pl for a proposition. In other words, an interval $[Bel\ ,\ Pl]$ is determined for the proposition. In our example this interval is $[0\ ,\ 1]$ for *genuine* before getting the advice from Mario and $[0.9\ ,\ 0.99]$ after that. The width of the interval $[Bel\ ,\ Pl]$ for the proposition represents the degree of completeness/incompleteness of our information, which can be used in a reasoning process. If we receive more and more information (evidence) the interval becomes narrow. Rules which allow us to take into account new evidence for constructing a belief function are defined in Dempster-Shafer Theory as well [67, 271].

12.3 Non-monotonic Reasoning

Reasoning models based on classical logic are *monotonic*. This means that after adding new formulas to a model the set of its consequences is not reduced. Extending the set of formulas can cause the possibility of inferring additional consequences, however all consequences that have been inferred previously are sound. In the case of AI systems which are to be used for reasoning about the real (physical) world, such a reasoning scheme is not valid, because our beliefs (assumptions) are often based on uncertain and incomplete knowledge.

For example, I claim “my car has good acceleration”.[27] I can use this proposition in a reasoning process, since I have no information which contradicts it. However, I have just got a new message that my car has been crushed by a bulldozer. This means that the claim “my car has good acceleration” should be removed from the set of my beliefs, as well as all propositions which have been previously inferred on the basis of this claim. Thus, the new proposition has not extended my set of beliefs. It has reduced this set. As we can see, common-sense logic which is used for reasoning about the real (physical) world is *non-monotonic*. Now, we introduce three non-monotonic models, namely default logic, autoepistemic logic, and circumscription reasoning.

[26] The stronger evidences are the less a value of $Pl(S)$ is.

[27] Let us assume that I have only one car.

Default logic was defined by Raymond Reiter[28] in 1980 [239]. It is a formalism which is more adequate for reasoning in AI systems than classical logic. Let us notice that even such seemingly simple and obvious propositions as "Mammals do not fly", if expressed in First Order Logic, i.e.,

$$\forall x [is\_mammal(x) \Rightarrow \quad does\_not\_fly(x)],$$

is false, because there are some mammals (bats), which fly. Of course, sometimes defining a list of *all* the exceptions is impossible in practice. Therefore, in default logic, apart form standard rules of inference[29] *default inference rules* are defined. In such rules a *consistency requirement* is introduced. This is of the form *"it is consistent to assume that $P(x)$ holds"*, which is denoted by $\mathbf{M}\, P(x)$. For our example such a rule can be formulated in the following way:

$$\frac{is\_mammal(x) \ : \ \mathbf{M}\, does\_not\_fly(x)}{does\_not\_fly(x)}$$

which can be interpreted as follows: "If x is a mammal and it is consistent to assume that x does not fly, then x does not fly". In other words: "If x is a mammal, then x does not fly in the absence of information to the contrary".

Reiter introduced a very convenient rule of inference for knowledge bases, called the *Closed-World Assumption, CWA*, in 1978 [240]. It says that the information included in a knowledge base is a complete description of the world, i.e., if something is not known to be true, then it is false.

Autoepistemic logic was formulated by Robert C. Moore[30] in 1985 [206] as a result of research which was a continuation of studies into modal non-monotonic systems led by Drew McDermott[31] and Jon Doyle in 1980 [199]. The main idea of this logic can be expressed as follows. Reasoning about the world can be based on our introspective knowledge/beliefs. For example, from the fact that I am convinced that I am not the husband of Wilma Flinstone, I can infer that I am not the husband of Wilma Flinstone, because I would certainly know that I am the husband of Wilma Flinstone, if I was the husband of Wilma Flinstone. Autoepistemic logic can be viewed as a modal logic containing an operator *"I am convinced that"*. In such logic sets of beliefs are used instead of sets of facts.

A non-monotonic logic called *circumscription* was constructed by John McCarthy in 1980 [196]. We introduce its main idea with the help of our example proposition

[28] Raymond Reiter—a professor of computer science and logic at the University of Toronto. His work concerns non-monotonic reasoning, knowledge representation models, logic programming, and image analysis.

[29] *Standard rules* means such rules as the ones introduced in Chap. 6.

[30] Robert C. Moore—a researcher at Microsoft Research and NASA Ames Research Center, Ph.D. in computer science (MIT). His work concerns NLP, artificial intelligence, automatic theorem proving, and speech recognition.

[31] Drew McDermott—a professor of computer science at Yale University. His work concerns AI, robotics, and pattern recognition.

concerning mammals. This time, however, in order to handle the problem defined above we introduce the predicate $is\_peculiar\_mammal(x)$. Now, we can express our proposition in First Order Logic in the following way:

$$\forall x[is\_mammal(x) \ \wedge \ \neg is\_peculiar\_mammal(x) \ \Rightarrow \ does\_not\_fly(x)].$$

Of course, we may not know whether a specific mammal is peculiar. Therefore, we minimize the extension of such a predicate as $is\_peculiar\_mammal(x)$, i.e., we minimize its extension only to the set of objects which are known to be peculiar mammals. For example, if *Zazu* is not in this set, then the following holds: $\neg is\_peculiar\_mammal(Zazu)$, which means that $does\_not\_fly(Zazu)$. Let us notice an analogy to the concept of Closed-World Assumption introduced above.

As we have mentioned at the beginning of this section, sometimes a non-monotonic-reasoning-based system should remove a certain proposition as well as propositions inferred on the basis of this proposition after gaining new information. One question is: "Should all the propositions inferred on the basis of such a proposition be removed?" If these propositions can be inferred only from a removed proposition, then of course they should also be removed. However, the system should not remove those propositions which can be inferred without using a removed proposition. In order to solve this problem practically Jon Doyle[32] introduced *Truth Maintenance Systems, TMS*, in 1979 [73]. Such systems can work according to various scenarios. The simplest scenario consists of removing all the conclusions inferred from a removed proposition-premise and repeating the whole inference process for all conclusions. However, this simple scenario is time-consuming. An improved version consists of remembering the chronology of entering new information and inferring propositions in the system. Then, after removing some proposition-premise *P*, only those conclusions are removed which have been inferred after storing the proposition-premise *P* in the knowledge base.

Remembering sequences of *justifications* for conclusions is an even more efficient method. If any proposition is removed, then all justifications, which can be inferred *only* on the basis of this proposition are also removed. If, after such an operation, a certain proposition cannot be justified, then it is invalidated.[33] This scenario is a basis for *Justification-based Truth Maintenance Systems, JTMSs*. They were defined by Doyle in 1979 [73].

[32] Jon Doyle—a professor of computer science at the Massachusetts Institute of Technology, Stanford University, and Carnegie-Mellon University. His work concerns reasoning methods, philosophical foundations of Artificial Intelligence, and AI applications in economy and psychology.

[33] Such a proposition does not need to be removed (physically) from the knowledge base. It is enough to mark that the proposition is invalid (currently). If, for example, the removed justification is restored, then the system needs only to change its status to valid.

In 1986 Johan de Kleer[34] introduced a new class of truth maintenance systems called *Assumption-based Truth Maintenance Systems, ATMSs* [66]. Whereas in systems based on justifications a consistent *image* of the world is stored (consisting of premises and *justified* propositions), all justifications that have been assumed in the knowledge base are maintained in an ATMS (maybe some of them currently as *invalid*). Thus, the system maintains all *assumptions* that can be used for inferring a given proposition. The system can justify a given proposition given a certain set of assumptions, called a *world*. Such an approach is especially useful if we want the system to change its view depending on its set of assumptions.

The issue of maintaining a knowledge base when new data frequently come into an AI system is closely related to the *frame problem* formulated by McCarthy and Patrick J. Hayes[35] in 1969 [195]. The issue concerns defining efficient formalisms for representing elements of a world description which *do not* change during an inference process.

Bibliographical Note

The monograph [223] is a good introduction to Bayesian inference and networks. A description of Dempster-Shafer Theory can be found in the classic book [271]. A concise introduction to non-monotonic reasoning in AI can be found in [39].

[34] Johan de Kleer—a director of Systems and Practices Laboratory, Palo Alto Research Center (PARC). His work concerns knowledge engineering, model-based reasoning, and AI applications in qualitative physics.

[35] Patrick John Hayes—a British computer scientist and mathematician, a professor of prestigious universities (Rochester, Stanford, Essex, Geneva). His work concerns knowledge representation, automated inference, philosophical foundations of AI, and semantic networks.