AITA : Treatment of Uncertainty

1. Sources and Types of Uncertainty
2. Probability Theory
3. Probabilistic Reasoning
4. Bayesian Networks
5. Certainty Factors and MYCIN
6. Dempster-Shafer Theory
7. Fuzzy Logic
8. Comparing Methods of Inexact Reasoning

Sources of Uncertainty

There are many potential sources of uncertainty that AI systems (such as expert systems) must be able to cope with, but most can be attributed to one of:

1. Imperfect Domain Knowledge

The theory of the domain may be vague or incomplete. Incompleteness necessitates the use of rules of thumb (or heuristics) which may not always give optimal or correct results. Even if the domain theory is complete, an expert may use approximations or heuristics to save time or simplify the problem solving.

2. Imperfect Case Data

Sensors have only finite resolving power and less than 100% reliability. Human reports may be ambiguous or inaccurate. Evidence from different sources may be missing or in conflict. Even if exact data were available, it may be too costly in time or resources to get it.

Whatever the source of uncertainty, we need our AI systems to be able to deal with it.

Types of Uncertainty

We can identify three major types of uncertainty that our systems may need to process:

1. **Randomness**

   Some events simply ***are*** random (rolling dice, sensor faults, etc.) and we need to work with probabilities. For example, nine times out of ten a faulty smoke detector will be caused by low battery power.

2. **Vagueness**

   Sometimes there is imprecision in the representation language (vague concepts such as ‘tall’, ‘likely’, ‘little’) and if such uncertainty occurs in rule conditions, the corresponding rule actions will need to carry similar imprecision.

3. **Inadequacy**

   Sometimes, in order to model an expert’s behaviour, rules must be given weights according to how useful/reliable they are.

Probability Theory

Most aspects of uncertainty can be represented as probabilities. We can adopt a simple ***frequentist*** approach to probability and say that if event A occurs N_A times out of a total of N occasions, then the probability of event A is

$$p(A) = \frac{N_A}{N}$$

Sometimes we really do estimate the probabilities empirically, e.g. rolling a pair of dice ten billion times to find $p(12)$.

Sometimes we can enumerate all the possibilities and determine the probabilities that way, e.g. list all combinations of two dice and find $p(12) = 1/36$.

Sometimes, we cannot take either approach and are forced to just make a good guess based on what information we might have, e.g. the probability of the temperature in Birmingham on 25th November 2004 must be estimated from known temperature distributions from other years and other places.

Prior or Unconditional Probabilities

When we have no other information, it is appropriate to use the ***prior*** or ***unconditional probability*** $p(A)$ of event A happening. As soon as further information is known, we must use ***conditional probabilities***.

Each random variable X will have a ***domain*** of possible values $D_X = \{x_1, x_2, \ldots, x_n\}$ that it can take on. We shall deal with discrete sets of values, though often continuous random variables need to be considered, in which case we need to talk about ***probability distributions*** rather than probabilities.

Clearly, probabilities must sum to 1, so we have two important results

$$\sum_{i=1}^{n} p(x_i) = 1 \qquad , \qquad p(\neg x_i) = 1 - p(x_i)$$

We can easily deal with probabilities over many independent domains, e.g. if we have independent variables X and Y, we have $p(x_1, y_2) \equiv p(x_1 \wedge y_2) = p(x_1).p(y_2)$.

Conditional or Posterior Probabilities

Once we have some information concerning our domain, the prior probabilities are no longer applicable. Instead we use ***conditional*** or ***posterior probabilities***

$$p(A \mid B) = \text{"probability of A given B"}$$

Combining this with a prior probability by the ***product rule*** gives the ***joint probability***

$$p(A \wedge B) = p(A \mid B).p(B)$$

Reversing this gives the conditional probabilities in terms of unconditional probabilities

$$p(A \mid B) = \frac{p(A \wedge B)}{p(B)}$$

By consideration of the ***Venn diagram*** we see that the probability of a disjunction is given by $p(A \vee B) = p(A) + p(B) - p(A \wedge B)$. Finally, notice that if A and B are independent, then $p(A|B) = p(A)$, and $p(A \wedge B) = p(A).p(B)$ as we had before.

Bayes’ Rule

Since $A \wedge B = B \wedge A$, the product rule can be written in two forms

$$p(A \wedge B) = p(A \mid B).p(B) = p(B \mid A).p(A)$$

from which we can obtain ***Bayes’ Rule*** (also known as Bayes’ Law or Bayes’ Theorem)

$$p(A \mid B) = \frac{p(B \mid A).p(A)}{p(B)}$$

This simple equation underlies all modern AI systems for probabilistic inference. This is because we often have the conditional probabilities in one direction but not the other.

Let *M* = *Meningitis* and *S* = *Stiff Neck.* Estimating *p(M|S)* might be hard, but we might know that *p(S|M)* = 1/2, *p(M)* = 1/50000, *p(S)* = 1/20, so we can easily use Bayes’ rule to give *p(M|S)* = 1/5000. A doctor ***might*** know this, but it will change in an epidemic and the doctor is unlikely to know the probability in the new situation. The Bayes’ rule user simply needs to update the probability *p(M)* which will be easily measurable.

Relative Likelihood

Often when we are only interested in the ***relative likelihood*** of two events, some of the probabilities cancel out leaving an easier computation.

Suppose W = *Whiplash* and we are interested in determining the relative likelihood that a patient with a stiff neck has whiplash rather than meningitis. From Bayes' Rule:

$$p(M \mid S) = \frac{p(S \mid M).p(M)}{p(S)} \qquad p(W \mid S) = \frac{p(S \mid W).p(W)}{p(S)}$$

and we can compute the relative likelihood by simple division

$$\frac{p(M \mid S)}{p(W \mid S)} = \frac{p(S \mid M).p(M)}{p(S \mid W).p(W)}$$

Given $p(S|M)$, $p(M)$, $p(S|W)$, $p(W)$ we don't need to know $p(S)$ in order to determine the relative likelihood of M or W given S.

Probabilistic Reasoning

We can now see how ***in principle*** to perform ***probabilistic reasoning***. Our system will be given a set of initial facts, possibly facts in terms of probabilities, and its inference engine will use the various rules of probability theory we have just looked at to compute the probabilities of the various solutions.

Note that ***in practice*** this approach will generally be intractable, because for cases where we have N variables we will need of the order of 2^N joint/conditional probabilities to compute the answers. For many problems of interest we couldn't even write down such a list, even if we could compute or determine all the values.

The way we get round this practical problem in AI systems (and in humans) is by making use of ***conditional independencies*** among the variables. If $p(A|\{x_i\},\{y_i\}) = p(A|\{x_i\})$, then A is conditionally independent of $\{y_i\}$ and, as far as A is concerned, if we know $\{x_i\}$ we can ignore $\{y_i\}$. Such simplifications, which occur naturally in many practical applications, render the idea of probabilistic inference feasible.

Bayesian Networks

With ***Bayesian Networks*** we preserve the probability theory formalism, but rely on the modularity of the world to reduce the complexity.

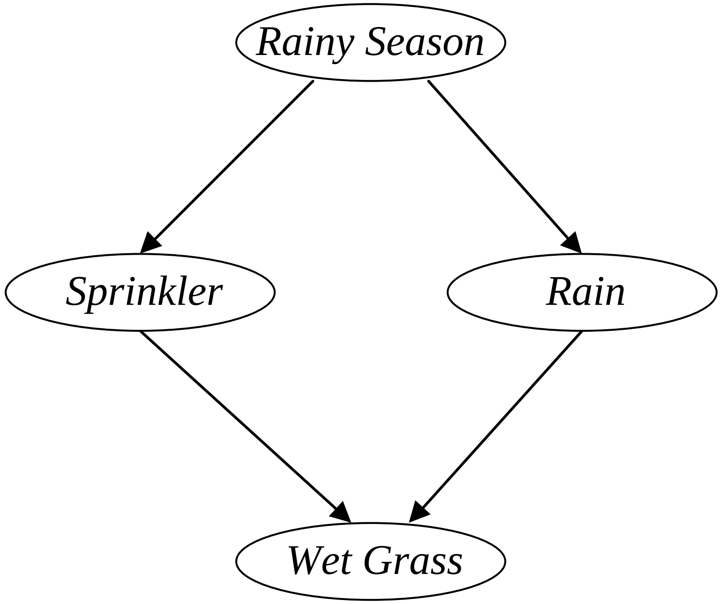
The idea is that most events are ***conditionally independent*** of most other events, and so their interactions need not be considered. This allows us to use a much more local representation in which we only describe clusters of events that do interact.

A Bayesian Network is a ***directed acyclic graph*** that represents causality relationships. The directed links between the nodes in the graph indicate direct influence. Missing links (usually much greater in number) correspond to the conditional independencies.

Each node has either a prior probability, or a conditional probability table that quantifies the effects of its parents (i.e. the nodes that influence it). We need a mechanism for computing the influence of any node on any other. We also need to ensure that the probabilities are transmitted correctly.

Bayesian Network Example

Consider the simple example of grass that may be wet because of rain or sprinkler:



Attribute	Probability
p(Wet|Sprinkler,Rain)	0.95
p(Wet|Sprinkler,¬Rain)	0.9
p(Wet|¬Sprinkler,Rain)	0.8
p(Wet|¬Sprinkler,¬Rain)	0.1
p(Sprinkler|RainySeason)	0.0
p(Sprinkler|¬RainySeason)	1.0
p(Rain|RainySeason)	0.9
p(Rain|¬RainySeason)	0.1
p(RainySeason)	0.5

In general, exact inference in Bayesian networks is known to be NP hard, but at least the approach generates consistently sensible results.

Certainty Factors and MYCIN

An alternative to full scale probabilistic reasoning was explored by the famous expert system MYCIN which diagnosed bacterial infections.

MYCIN was based on ***certainty factors*** rather than probabilities. These certainty factors CF are in the range [–1,+1] where –1 means certainly false and +1 means certainly true. The system was based on rules of the form:

IF: the patient has signs and symptoms $s_1 \wedge s_2 \wedge \ldots \wedge s_n$, and
certain background conditions $t_1 \wedge t_2 \wedge \ldots \wedge t_m$ hold
THEN: conclude that the patient had disease d_i with certainty C_R

The idea was to use production rules of this kind in an attempt to approximate the calculation of the conditional probabilities $p(d_i | s_1 \wedge s_2 \wedge \ldots \wedge s_n)$, and provide a scheme for accumulating evidence that approximated the reasoning process of an expert. MYCIN did not actually achieve that, but it nevertheless remains a useful approach.

Processing the MYCIN Rules

Clearly, if the symptoms s_i and background conditions t_j themselves have certainties less than 1, then the degree of certainty produced by a rule must be reduced appropriately.

Generally the certainty factor of a conjunction will be the minimum certainty of their individual certainties

$$\mathrm{CF}(s_1 \wedge \ldots \wedge s_n \wedge t_1 \wedge \ldots \wedge t_m) = \min(\mathrm{CF}(s_1), \ldots \mathrm{CF}(s_n), \mathrm{CF}(t_1), \ldots \mathrm{CF}(t_m))$$

The idea is that we are only confident of the conjunction to the extent to which we are confident in the least inspiring element, or that the chain is only as strong as its weakest link. To get the final certainty this is multiplied by the certainty C_R of the rule R giving

$$\mathrm{CF}(d_i, s_1 \wedge \ldots \wedge s_n \wedge t_1 \wedge \ldots \wedge t_m) = C_R \,.\, \min(\mathrm{CF}(s_1), \ldots \mathrm{CF}(s_n), \mathrm{CF}(t_1), \ldots \mathrm{CF}(t_m))$$

The background conditions t_j tend to have binary values and act in a trivial way.

Combining Certainty Factors

In order to derive or estimate the likelihood of a fact *A*, all the rules that generate *A* must be investigated, and their certainty factors must be combined.

If *X* and *Y* are the certainties derived from two independent rules, then the combined certainty *CC* is given by

$$CC = \begin{cases} X + Y - X.Y & \text{if X, Y} > 0 \\ X + Y + X.Y & \text{if X, Y} < 0 \\ (X + Y)/(1 - \min(|X|, |Y|)) & \text{otherwise} \end{cases}$$

We can see intuitively what happens. If two pieces of evidence both confirm (or disconfirm) the hypothesis, then the confidence in the hypothesis goes up (or down). If two pieces of evidence conflict, then the denominator dampens the effect. The formula can be used repeatedly if there are many certainties to combine, and the order in which it is done does not matter. (In practice we improve the efficiency, at the expense of accuracy, by ignoring low certainty values, e.g. |X|<0.2)

Advantages and Disadvantages of Certainty Factors

The principal advantages and disadvantages of the Certainty Factors approach are:

Advantages

1. They have been very successful in a number of useful applications.
2. They can be used to reduce search by pruning out branches with low certainty.

Disadvantages

1. Getting an expert to produce a sufficiently consistent and accurate enough set of certainty factors on which to base the system is extremely difficult.
2. It is not always clear what the certainty factors really mean.
3. It is debatable whether they really correspond to human-like reasoning.
4. Perhaps most importantly: one can find situations where they produce the opposite results to proper probability theory (e.g. see Jackson, Section 9.2.3).

Dempster-Shafer Theory

Another alternative to Bayesian Networks is ***Dempster-Shafer Theory*** which is designed to deal directly with the distinction between ***uncertainty*** and ***ignorance***. Rather than computing probabilities of propositions, it computes probabilities that evidence supports the propositions. This measure of belief is called a ***belief function***, written *Bel(X)*.

Suppose a shady lecturer offers to bet you £20 that his coin will come up heads next flip. Clearly you should not trust his coin to be fair. Dempster-Shafer theory tells you that since there is no evidence either way, you should have *Bel(Heads)* = 0 and *Bel(¬Heads)* = 0. But if an expert says she is 90% certain that the coin is fair, your beliefs are different with *Bel(Heads)* = 0.9 × 0.5 = 0.45 and also *Bel(¬Heads)* = 0.9 × 0.5 = 0.45. There is now only a 0.1 "gap" still not accounted for by the evidence. The expert evidence has reduced the probability interval for *Heads* from [0, 1] to [0.45, 0.55].

Dempster's rule shows you how to combine such evidence, and ***Shafer***'s work extends this into a complete computational model.

Dempster-Shafer Theory : The Notation

To define the belief *Bel(X)* precisely we need to start with a full list of mutually exclusive hypotheses. This hypothesis space is called the ***frame of discernment***, denoted by Θ. In a simple medical diagnosis systems we might have $\Theta = \{allergy, cold, flu, pneumonia\}$. The goal is to attach some measure of belief to each element of Θ.

Note that the evidence will often support more than one hypothesis. Also, because the hypotheses are mutually exclusive, evidence in favour of one hypothesis may affect our belief an another. In a purely Bayesian system, we handle this by listing all the combinations of conditional probabilities. Dempster-Shafer theory attempts to avoid the need for that by manipulating the sets of hypotheses directly.

If Θ has n elements, it will have 2^n sub-sets and the ***basic probability assignment*** *m(X)* measures the amount of belief currently assigned to each sub-set X. We assign these so that the sum of *m(X)* over all sub-sets $X \subseteq \Theta$ is 1. Although this means dealing with 2^n values, many will be zero as the corresponding sets have no relevance to the problem.

Dempster-Shafer Theory : Combining Evidence

In practice, we will have a belief function $m_i(X)$ corresponding to each piece i of evidence and we need to combine then. The basic ***Dempster rule*** is

$$m(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) m_2(Y)}{1 - \sum_{X \cap Y = \varnothing} m_1(X) m_2(Y)}$$

With no evidence we start with $m_1(\Theta) = 1$ and $m_1(X \neq \Theta) = 0$. Now suppose, in the above example, we then get evidence of Fever and of Runny nose, and individually they imply

Fever ⇒ $m_1(\{Flu, Cold, Pneu\}) = 0.6$, $m_1(\Theta) = 0.4$

Runny nose ⇒ $m_2(\{Allergy, Flu, Cold\}) = 0.8$, $m_2(\Theta) = 0.2$

Then these two pieces of evidence can be combined using Dempster's rule to give

$$m(\{Flu, Cold\}) = 0.48 \quad , \quad m(\{Allergy, Flu, Cold\}) = 0.32 \quad ,$$
$$m(\{Flu, Cold, Pneu\}) = 0.12 \quad , \quad m(\Theta) = 0.08$$

As more evidence accumulates we hope to get a single hypothesis with high belief.

Fuzzy Set Theory

Another useful alternative to probability theory for uncertain reasoning in AI systems is ***fuzzy logic***. It is based on the idea that many concepts are not sharply defined (e.g. ‘fast’, ‘tall’, ‘hot’) and consequently we cannot use standard set theory and *if-then* rules when reasoning with them. Fuzzy logic is built upon the underlying idea of ***fuzzy set theory***.

Classical set theory is based on two-valued logic, in which relations such as $X \in S$ are either true or false. Such classical sets are sometimes called ***crisp*** sets. We can define a crisp set of fast cars as those cars which have a top speed greater than 150mph :

$$FastCars = \{ X \in Cars : TopSpeed(X) > 150\text{mph} \}$$

But the concept of fast car is not really precise like that. It is more reasonable to define it as a ***fuzzy set*** with elements that are members ***to a certain degree***. A fuzzy set is thus defined as a function from the appropriate domain to the interval [0, 1] such that $f(X) = 1$ denotes X is definitely a member, $f(X) = 0$ denotes X is definitely not a member, and other values denote intermediate degrees of membership.

Fuzzy Logic

Just as classical set theory is governed by two-valued logic, fuzzy set theory can be related to a many valued logic in which propositions such as *FastCars(X)* have values in the interval [0, 1], and we define appropriate extensions to the standard rules of logic.

We can define the negation of a fuzzy predicate *f(X)* as in probability theory:

$$\neg f(X) = 1 - f(X)$$

The rules for evaluating the fuzzy truth of other operators are less obvious:

$$f(A \wedge B) = \min(\ f(A), f(B)\ )$$

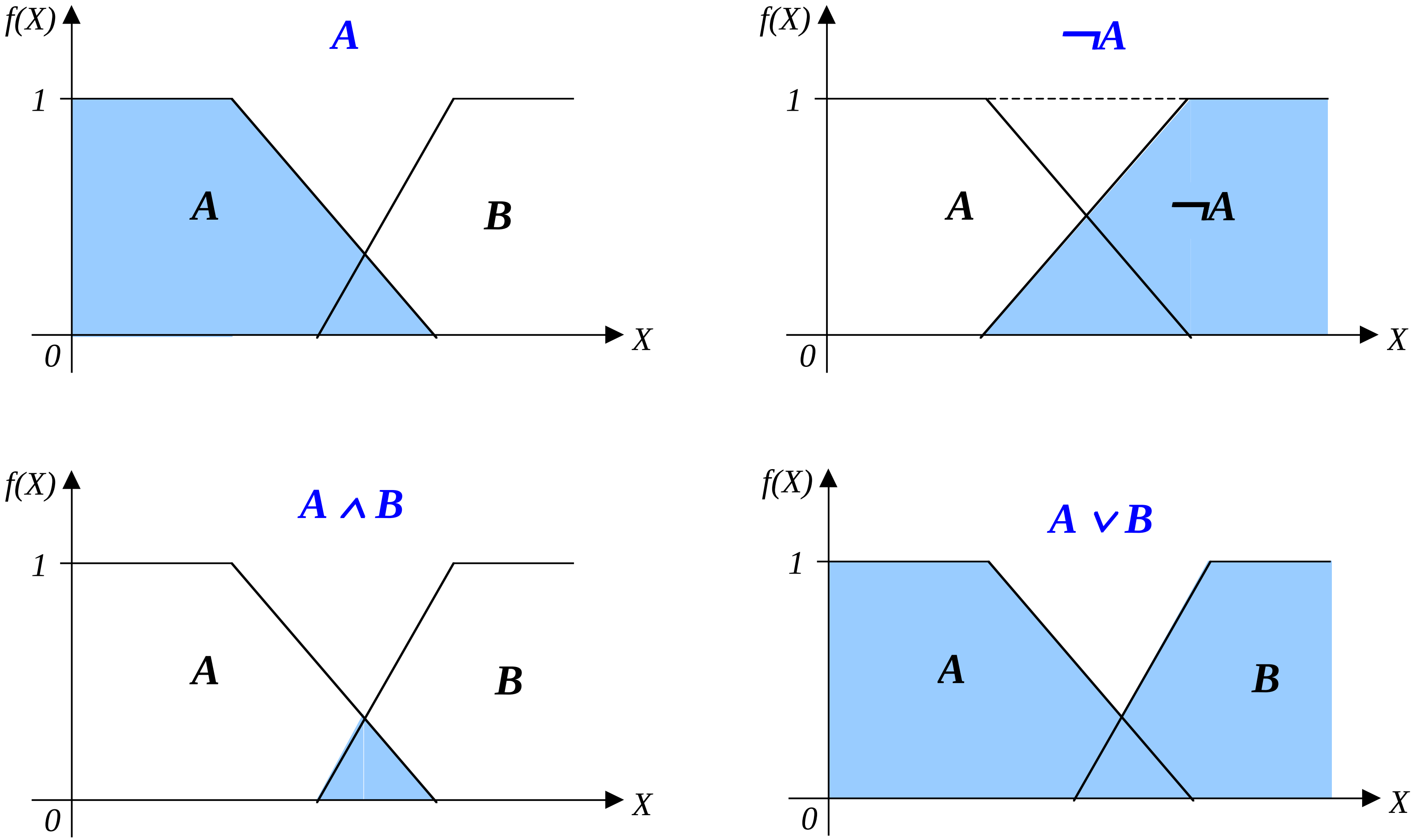
$$f(A \vee B) = \max(\ f(A), f(B)\ )$$

$$f(A \Rightarrow B) = \min(\ 1, 1 - f(A) + f(B)\ )$$

These definitions are similar to those used with the certainty factors in MYCIN. Note that they imply $f(A \vee \neg A) \neq f(\text{True})$ which we might expect to lead to problems...

Examples of Fuzzy Operations

Some simple examples should clarify the thinking behind the fuzzy operator definitions:



Fuzzy Rule-Based Systems

We can apply the fuzzy predicates and fuzzy logic to rule based systems (e.g. production systems or expert systems) in a similar manner to MYCIN style certainty factors.

We can use the fuzzy logic operators to compute the ***degree of truth*** of the conditions and take that to be the degree of truth of the action. If the rule itself is only true to a certain degree, then we simply have to apply that factor as well.

Building fuzzy expert systems follows the same procedures as any other expert system, except that we have to get the expert to define all the fuzzy sets and fuzzy rules, and make sure they are all ***internally consistent***. Usually a lot of fine tuning is required!

Fuzzy logic based expert systems have been very successful in commercial applications, but these have been rather small, with limited levels of inference, and parameters ***tuned by machine learning***. Various counter intuitive features, such as $f(A \vee \neg A) \neq f(\text{True})$, have led some people to regard the use of fuzzy logic in more complex expert systems to be as problematic and unreliable as the use of MYCIN style certainty factors.

Comparing Methods of Inexact Reasoning

We can now compare the main alternative approaches for treating uncertainty:

Bayesian Probability Theory / Bayesian Networks : The obvious consistent approach, but it is often extremely *computationally intensive*.

MYCIN Style Certainty Factor Models : These make stronger assumptions (than probabilistic models of belief) that can actually render the system *self-inconsistent*.

Dempster-Shafer Theory : This allows one to state that certain prior and conditional probabilities cannot be assessed, and provides the notion of a compatibility relation between beliefs. It appears to be a consistent *generalization of probability theory*.

Fuzzy Logic : The fuzzification of truth values is inconsistent with the basic idea of conditional probabilities because of the modified definition of conjunction. We also have the non-standard property $f(A \vee \neg A) \neq f(\text{True})$. However, it appears to be a consistent *alternative to probability theory*.

For more detailed comparisons refer to Jackson, Sections 21.3 and 21.4.

Overview and Reading

1. We began by looking at the sources and types of uncertainty that our AI systems must be able to deal with.
2. We then went through the ideas of standard probability theory, including Bayes' Rule, and how this leads to Bayesian Belief Networks.
3. We then considered MYCIN style Certainty Factors, Dempster-Shafer Theory and Fuzzy Logic, which are often more tractable approaches.
4. We ended with a brief comparison of the various uncertainty treatments.

Reading

1. Russell & Norvig: Chapters 13 & 14
2. Jackson: Chapters 9 & 21
3. Nilsson: Chapters 19 & 20
4. Rich & Knight: Chapters 7 & 8
5. Negnevitsky: Chapters 3 & 4