Issues in knowledge representation

The main objective of knowledge representation is to draw the conclusions from the knowledge, but there are many issues associated with the use of knowledge representation techniques.

Some of them are listed below:

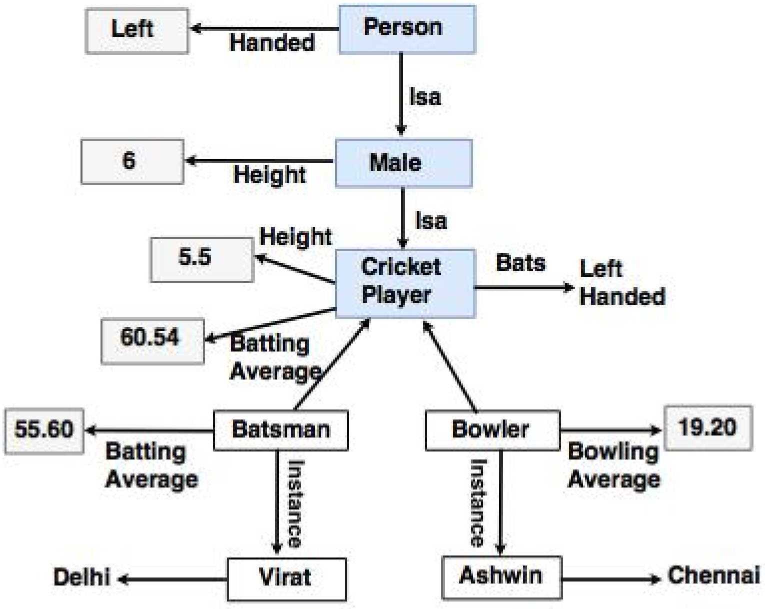


Fig: Inheratable Knowledge Representation

Refer to the above diagram to refer to the following issues.

1. Important attributes

There are two attributes shown in the diagram, **instance** and **isa.** Since these attributes support property of inheritance, they are of prime importance.

2. Relationships among attributes

Basically, the attributes used to describe objects are nothing but the entities. However, the attributes of an object do not depend on the encoded specific knowledge.

3. Choosing the granularity of representation

While deciding the granularity of representation, it is necessary to know the following:

i. What are the primitives and at what level should the knowledge be represented?

ii. What should be the number (small or large) of low-level primitives or high-level facts?

High-level facts may be insufficient to draw the conclusion while Low-level primitives may require a lot of storage.

For example: Suppose that we are interested in following facts:
John spotted Alex.

Now, this could be represented as "Spotted (agent(John), object (Alex))"

Such a representation can make it easy to answer questions such as:
Who spotted Alex?

Suppose we want to know : "Did John see Sue?"
Given only one fact, user cannot discover that answer.

Hence, the user can add other facts, such as "Spotted (x, y) → saw (x, y)"

4. Representing sets of objects.
There are some properties of objects which satisfy the condition of a set together but not as individual;

Example: Consider the assertion made in the sentences:
"There are more sheep than people in Australia", and "English speakers can be found all over the world."
These facts can be described by including an assertion to the sets representing people, sheep, and English.

5. Finding the right structure as needed
To describe a particular situation, it is always important to find the access of right structure. This can be done by selecting an initial structure and then revising the choice.

While selecting and reversing the right structure, it is necessary to solve following problem statements. **They include the process on how to:**

- Select an initial appropriate structure.
- Fill the necessary details from the current situations.
- Determine a better structure if the initially selected structure is not appropriate to fulfill other conditions.
- Find the solution if none of the available structures is appropriate.
- Create and remember a new structure for the given condition.
- There is no specific way to solve these problems, but some of the effective knowledge representation techniques have the potential to solve them.

Logic Representation

Facts are the general statements that may be either True or False. Thus, logic can be used to represent such simple facts.

To build a Logic-based representation:

- User has to define a set of primitive symbols along with the required semantics.
- The symbols are assigned together to define legal sentences in the language for representing TRUE facts.

- New logical statements are formed from the existing ones. The statements which can be either TRUE or false but not both , are called propositions. A declarative sentence expresses a statement with a proposition as content;
  Example: The declarative "Cotton is white" expresses that Cotton is white. So, the sentence "Cotton is white" is a true statement.

Monotonic Reasoning

Monotonic Reasoning is the process that does not change its direction or can say that it moves in the one direction.

Non-monotonic Reasoning

Non-monotonic Reasoning is the process that changes its direction or values as the knowledge base increases.

- It is also known as NMR in Artificial Intelligence.
- Non-monotonic Reasoning will increase or decrease based on the condition.
- Since that Non-monotonic Reasoning depends on knowledge and facts, it will change itself with improving knowledge or facts.
  Example:
    - Consider a bowl of water, If we put it on the stove and turn the flame on it will obviously boil hot and as we will turn off the flame it will cool down gradually.

- In Non-monotonic reasoning, some conclusions may be invalidated if we add some more information to our knowledge base.
- Logic will be said as non-monotonic if some conclusions can be invalidated by adding more knowledge into our knowledge base.
- Non-monotonic reasoning deals with incomplete and uncertain models.
- "Human perceptions for various things in daily life, "is a general example of non-monotonic reasoning.

Example: Let suppose the knowledge base contains the following knowledge:

- **Birds can fly**
- **Penguins cannot fly**
- **Pitty is a bird**

So from the above sentences, we can conclude that **Pitty can fly**.

However, if we add one another sentence into knowledge base "**Pitty is a penguin**", which concludes "**Pitty cannot fly**", so it invalidates the above conclusion.

Advantages of Non-monotonic reasoning:

- For real-world systems such as Robot navigation, we can use non-monotonic reasoning.
- In Non-monotonic reasoning, we can choose probabilistic facts or can make assumptions.

Disadvantages of Non-monotonic Reasoning:

- In non-monotonic reasoning, the old facts may be invalidated by adding new sentences.
- It cannot be used for theorem **proving**.

Monotonic Reasoning vs Non-monotonic Reasoning

	Monotonic Reasoning	Non-Monotonic Reasoning
1	Monotonic Reasoning is the process which does not change its direction or can say that it moves in the one direction.	Non-monotonic Reasoning is the process which changes its direction or values as the knowledge base increases.
2	Monotonic Reasoning deals with very specific type of models, which has valid proofs.	Non-monotonic reasoning deals with incomplete or not known facts.
3	The addition in knowledge won't change the result.	The addition in knowledge will invalidate the previous conclusions and change the result.
4	In monotonic reasoning, results are always true, therefore, set of prepositions will only increase.	In non-monotonic reasoning, results and set of prepositions will increase and decrease based on condition of added knowledge.
5	Monotonic Reasoning is based on true facts.	Non-monotonic Reasoning is based on assumptions.
6	Deductive Reasoning is the type of monotonic reasoning.	Abductive Reasoning and Human Reasoning is a non-monotonic type of reasoning.

Bayes' theorem in Artificial intelligence

Bayes' theorem:

Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.

In probability theory, it relates the conditional probability and marginal probabilities of two random events.

Bayes' theorem was named after the British mathematician **Thomas Bayes**. The **Bayesian inference** is an application of Bayes' theorem, which is fundamental to Bayesian statistics.

It is a way to calculate the value of $P(B|A)$ with the knowledge of $P(A|B)$.

Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world.

Example: If cancer corresponds to one's age then by using Bayes' theorem, we can determine the probability of cancer more accurately with the help of age.

Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:

As from product rule we can write:

1. $P(A \wedge B)= P(A|B) P(B)$ or

Similarly, the probability of event B with known event A:

1. $P(A \wedge B)= P(B|A) P(A)$

Equating right hand side of both the equations, we will get:

$$\boxed{P(A|B) = \frac{P(B|A)\, P(A)}{P(B)}} \quad \text{....(a)}$$

The above equation (a) is called as **Bayes' rule** or **Bayes' theorem**. This equation is basic of most modern AI systems for **probabilistic inference**.

It shows the simple relationship between joint and conditional probabilities. Here,

$P(A|B)$ is known as **posterior**, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.

$P(B|A)$ is called the likelihood, in which we consider that hypothesis is true, then we calculate the probability of evidence.

$P(A)$ is called the **prior probability**, probability of hypothesis before considering the evidence

$P(B)$ is called **marginal probability**, pure probability of an evidence.

In the equation (a), in general, we can write P (B) = P(A)*P(B|Ai), hence the Bayes' rule can be written as:

$$P(A_i|B) = \frac{P(A_i)*P(B|A_i)}{\sum_{i=1}^{k} P(A_i)*P(B|A_i)}$$

Where A_1, A_2, A_3,........, A_n is a set of mutually exclusive and exhaustive events.

Applying Bayes' rule:

Bayes' rule allows us to compute the single term P(B|A) in terms of P(A|B), P(*B*), and P(A). This is very useful in cases where we have a good probability of these three terms and want to determine the fourth one. Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Bayes' rule becomes:

$$\textbf{P(cause|effect)} = \frac{P(effect|cause)\ P(cause)}{P(effect)}$$

Example-1:

Question: what is the probability that a patient has diseases meningitis with a stiff neck?

Given Data:

A doctor is aware that disease meningitis causes a patient to have a stiff neck, and it occurs 80% of the time. He is also aware of some more facts, which are given as follows:

- The Known probability that a patient has meningitis disease is 1/30,000.
- The Known probability that a patient has a stiff neck is 2%.

Let a be the proposition that patient has stiff neck and b be the proposition that patient has meningitis. , so we can calculate the following as:

P(a|b) = 0.8

P(b) = 1/30000

P(a)= .02

$$\textbf{P(b|a)} = \frac{P(a|b)P(b)}{P(a)} = \frac{0.8*(\frac{1}{30000})}{0.02} = 0.001333333.$$

Hence, we can assume that 1 patient out of 750 patients has meningitis disease with a stiff neck.

Example-2:

Question: From a standard deck of playing cards, a single card is drawn. The probability that the card is king is 4/52, then calculate posterior probability P(King|Face), which means the drawn face card is a king card.

Solution:

$$\mathbf{P(king|face)} = \frac{P(Face|king) * P(King)}{P(Face)} \quad ....(i)$$

P(king): probability that the card is King= 4/52= 1/13

P(face): probability that a card is a face card= 3/13

P(Face|King): probability of face card when we assume it is a king = 1

Putting all values in equation (i) we will get:

$$P(king|face) = \frac{1 * (\frac{1}{13})}{(\frac{3}{13})} = \mathbf{1/3}\text{, it is a probability that a face card is a king card.}$$

Application of Bayes' theorem in Artificial intelligence:

Following are some applications of Bayes' theorem:

- It is used to calculate the next step of the robot when the already executed step is given.
- Bayes' theorem is helpful in weather forecasting.
- It can solve the Monty Hall problem.

Uncertainty:

Till now, we have learned knowledge representation using first-order logic and propositional logic with certainty, which means we were sure about the predicates. With this knowledge representation, we might write A→B, which means if A is true then B is true, but consider a situation where we are not sure about whether A is true or not then we cannot express this statement, this situation is called uncertainty.

So to represent uncertain knowledge, where we are not sure about the predicates, we need uncertain reasoning or probabilistic reasoning.

Causes of uncertainty:

Following are some leading causes of uncertainty to occur in the real world.

1. Information occurred from unreliable sources.
2. Experimental Errors
3. Equipment fault
4. Temperature variation

5. Climate change.

Probabilistic reasoning:

Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge. In probabilistic reasoning, we combine probability theory with logic to handle the uncertainty.

We use probability in probabilistic reasoning because it provides a way to handle the uncertainty that is the result of someone's laziness and ignorance.

In the real world, there are lots of scenarios, where the certainty of something is not confirmed, such as "It will rain today," "behavior of someone for some situations," "A match between two teams or two players." These are probable sentences for which we can assume that it will happen but not sure about it, so here we use probabilistic reasoning.

Need of probabilistic reasoning in AI:

- When there are unpredictable outcomes.
- When specifications or possibilities of predicates becomes too large to handle.
- When an unknown error occurs during an experiment.

In probabilistic reasoning, there are two ways to solve problems with uncertain knowledge:

- **Bayes' rule**
- **Bayesian Statistics**

Note: We will learn the above two rules in later chapters.

As probabilistic reasoning uses probability and related terms, so before understanding probabilistic reasoning, let's understand some common terms:

Probability: Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0 and 1 that represent ideal uncertainties.

1. 0 ≤ P(A) ≤ 1,   where P(A) is the probability of an event A.
1. P(A) = 0, indicates total uncertainty in an event A.
1. P(A) =1, indicates total certainty in an event A.

We can find the probability of an uncertain event by using the below formula.

$$\textbf{Probability of occurrence} = \frac{\text{Number of desired outcomes}}{\text{Total number of outcomes}}$$

- P(¬A) = probability of a not happening event.
- P(¬A) + P(A) = 1.

Event: Each possible outcome of a variable is called an event.

Sample space: The collection of all possible events is called sample space.

Random variables: Random variables are used to represent the events and objects in the real world.

Prior probability: The prior probability of an event is probability computed before observing new information.

Posterior Probability: The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

Conditional probability:

Conditional probability is a probability of occurring an event when another event has already happened.

Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it can be written as:

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

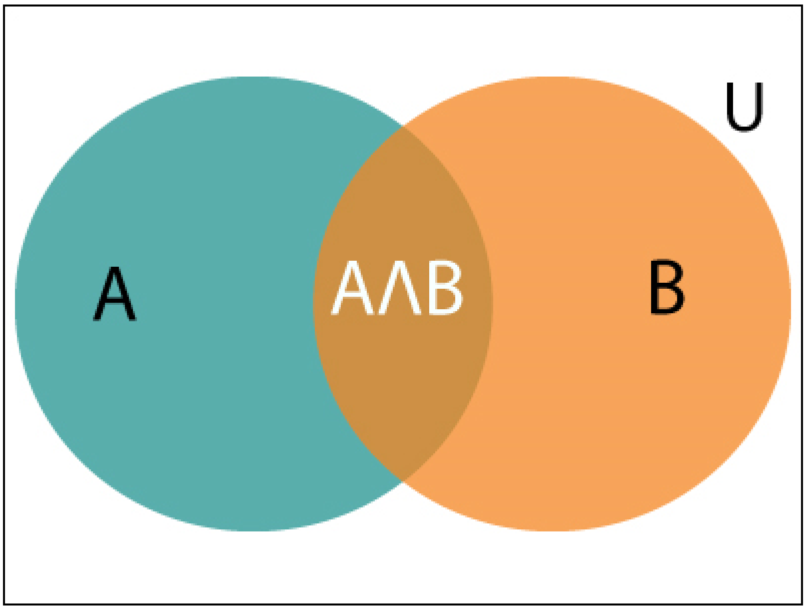
Where P(*A*∧*B*)= Joint probability of a and B

P(B)= Marginal probability of B.

If the probability of A is given and we need to find the probability of B, then it will be given as:

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

It can be explained by using the below Venn diagram, where B is occurred event, so sample space will be reduced to set B, and now we can only calculate event A when event B is already occurred by dividing the probability of **P(A∧*B*) by P(B)**.



Example:

In a class, there are 70% of the students who like English and 40% of the students who likes English and mathematics, and then what is the percent of students those who like English also like mathematics?

Solution:

Let, A is an event that a student likes Mathematics

B is an event that a student likes English.

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{0.4}{0.7} = 57\%$$

Hence, 57% are the students who like English also like Mathematics.

Representing Knowledge in an uncertain domain

There are various ways of representing uncertainty. Here we consider three different approaches, representing three different areas of uncertainty:

Probability theory:

Probabilistic assertions and queries are not usually about particular possible worlds, but about sets of them.

In probability theory, the set of all possible worlds is called the sample space. The Greek letter Ω (uppercase omega) is used to refer to the sample space, and ω (lowercase omega) refers to elements of the space, that is, particular possible worlds.

A fully specified probability model associates a numerical probability P (ω) with each possible world.1 the basic axioms of probability theory say that every possible world has a probability between 0 and 1 and that the total probability of the set of possible worlds is 1:

$0 \leq P(\omega) \leq 1$ for every ω and _$\omega \in \Omega$

$P(\omega) = 1$

i. If a coin is flipped there is an equal chance of it landing on head side or tail side, consider H1 is for heads and H2 for tails. This scenario is expressed as $P(H1)=0.5$ and $P(H2)=0.5$.

ii. The probability of 1st and 2nd toss both landing on heads is 0.5*0.5=0.25.

iii. We can write this as P(H1^H2)-0.25 and in general two independent events P and Q, P(P^Q)=P(P)*P(Q).

Fuzzy logic:

In the existing expert systems, uncertainty is dealt with through a combination of predicate logic and probability-based methods. A serious shortcoming of these methods is that they are not capable of coming to grips with the pervasive fuzziness of information in the knowledge base, and, as a result, are mostly ad hoc in nature.

An alternative approach to the management of uncertainty which is suggested in this paper is based on the use of fuzzy logic, which is the logic underlying approximate or, equivalently, fuzzy reasoning. A feature of fuzzy logic which is of particular importance to the management of uncertainty in expert systems is that it provides a systematic framework for dealing with fuzzy quantifiers, e.g., most, many, few, not very many, almost all, infrequently, about 0.8, etc.

In this way, fuzzy logic subsumes both predicate logic and probability theory, and makes it possible to deal with different types of uncertainty within a single conceptual framework.

In fuzzy logic, the deduction of a conclusion from a set of premises is reduced, in general, to the solution of a nonlinear program through the application of projection and extension principles. This approach to deduction leads to various basic syllogisms which may be used as rules of combination of evidence in expert systems.

Truth maintenance System:

To choose their actions, reasoning programs must be able to make assumptions and subsequently revise their beliefs when discoveries contradict these assumptions.

The Truth Maintenance System (TMS) is a problem solver subsystem for performing these functions by recording and maintaining the reasons for program beliefs. Such recorded reasons are useful in constructing explanations of program actions and in guiding the course of action of a problem solver

TMS are another form of knowledge representation which is best visualized in terms of graphs.

It stores the latest truth value of any predicate. The system is developed with the idea that truthfulness of a predicate can change with time, as new knowledge is added or exiting knowledge is updated.

It keeps a record showing which items of knowledge is currently believed or disbelieved.