CHANDIGARH ENGINEERING COLLEGE----------B.TECH.(CSE 6th Sem.)

Notes

Subject: Artificial Intelligence

Subject Code: ((BTCS 602-18))

Unit- 3 Probabilistic Reasoning

Probability

Using Uncertain Knowledge- Agents don't have complete knowledge about the world. Agents need to make decisions based on their uncertainty. It isn't enough to assume what the world is like. Example: wearing a seat belt. An agent needs to reason about its uncertainty.

Why Probability?

There is lots of uncertainty about the world, but agents still need to act. Predictions are needed to decide what to do: I definitive predictions: you will be run over tomorrow I point probabilities: probability you will be run over tomorrow is 0.002 I probability ranges: you will be run over with probability in range [0.001,0.34] Acting is gambling: agents who don't use probabilities will lose to those who do — Dutch books. Probabilities can be learned from data. Bayes' rule specifies how to combine data and prior knowledge. Probability is an agent's measure of belief in some proposition — subjective probability. An agent's belief depends on its prior assumptions and what the agent observes.

Numerical Measures of Belief

Belief in proposition, f , can be measured in terms of a number between 0 and 1 — this is the probability of f . I The probability f is 0 means that f is believed to be definitely false. I The probability f is 1 means that f is believed to be definitely true. Using 0 and 1 is purely a convention. f has a probability between 0 and 1, means the agent is ignorant of its truth value. Probability is a measure of an agent's ignorance. Probability is not a measure of degree of truth.

Random Variables

A random variable is a term in a language that can take one of a number of different values. The range of a variable X, written range(X), is the set of values X can take. A tuple of random variables hX1, . . . , Xni is a complex random variable with range range(X1) × · · · × range(Xn). Often the tuple is written as X1, . . . , Xn. Assignment X = x means variable X has value x. A proposition is a Boolean formula made from assignments of values to variables.

Possible World Semantics

A possible world specifies an assignment of one value to each random variable. A random variable is a function from possible worlds into the range of the random variable. ω |= X = x means variable X is assigned value x in world ω. Logical connectives have their standard

meaning: ω |= α ∧ β if ω |= α and ω |= β ω |= α ∨ β if ω |= α or ω |= β ω |= ¬α if ω 6|= α Let Ω be the set of all possible worlds.

Semantics of Probability

For a finite number of possible worlds: Define a nonnegative measure μ(ω) to each world ω so that the measures of the possible worlds sum to 1. The probability of proposition f is defined by: P(f) = X ω|=f μ(ω)

Axioms of Probability: finite case

Three axioms define what follows from a set of probabilities:

Axiom 1 $0 \leq P(a)$ for any proposition a.

Axiom 2 P(true) = 1

Axiom 3 P(a ∨ b) = P(a) + P(b) if a and b cannot both be true. These axioms are sound and complete with respect to the semantics.

Probabilistic reasoning in Artificial intelligence

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**

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. $0 \leq P(A) \leq 1$, 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.

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

- $P(\neg A)$ = probability of a not happening event.
- $P(\neg A) + P(A) = 1$.

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:

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

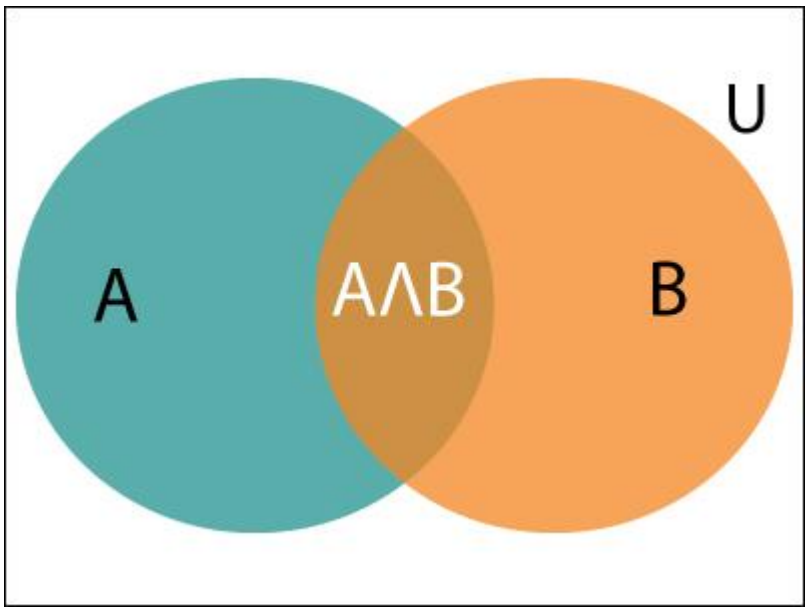
Where P($A \wedge B$)= Joint probability of a and B

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$\wedge B$) 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.

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:

$$P(A \wedge B) = P(A|B)\, P(B) \text{ or}$$

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

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

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

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

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:

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

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

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

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:

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

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:

$$P(king|face) = \frac{1 * (\frac{1}{13})}{(\frac{3}{13})} = 1/3,$$ **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.

Bayesian Belief Network in artificial intelligence

Bayesian belief network is key computer technology for dealing with probabilistic events and to solve a problem which has uncertainty. We can define a Bayesian network as:

"A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph."

It is also called a **Bayes network, belief network, decision network**, or **Bayesian model**.

Bayesian networks are probabilistic, because these networks are built from a **probability distribution**, and also use probability theory for prediction and anomaly detection.

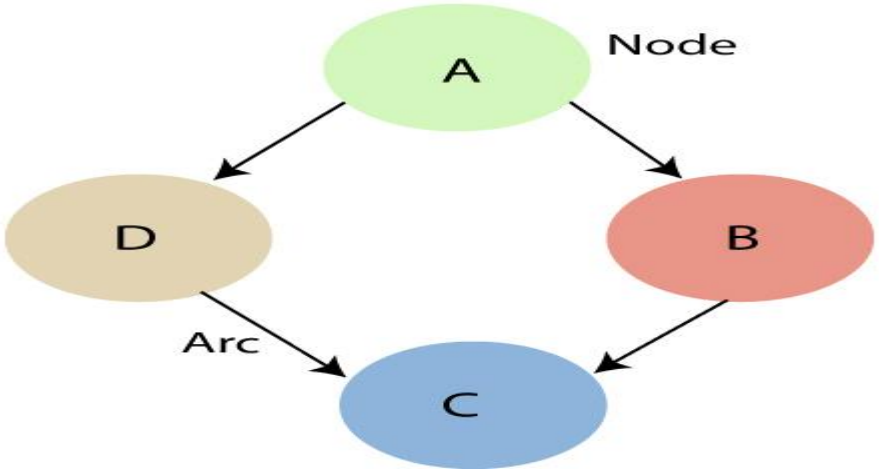
Real world applications are probabilistic in nature, and to represent the relationship between multiple events, we need a Bayesian network. It can also be used in various tasks including **prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction**, and **decision making under uncertainty**.

Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:

- **Directed Acyclic Graph**
- **Table of conditional probabilities.**

The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an **Influence diagram**.

A Bayesian network graph is made up of nodes and Arcs (directed links), where:



- Each **node** corresponds to the random variables, and a variable can be **continuous** or **discrete**.
- **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph.
  These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other
  - **In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.**
  - **If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B.**
  - **Node C is independent of node A.**

The Bayesian network has mainly two components:

- **Causal Component**
- **Actual numbers**

Each node in the Bayesian network has condition probability distribution **$P(X_i | Parent(X_i))$**, which determines the effect of the parent on that node.

Bayesian network is based on Joint probability distribution and conditional probability. So let's first understand the joint probability distribution:

Joint probability distribution:

If we have variables x1, x2, x3,....., xn, then the probabilities of a different combination of x1, x2, x3.. xn, are known as Joint probability distribution.

$P[x_1, x_2, x_3,....., x_n]$, it can be written as the following way in terms of the joint probability distribution.

$= P[x_1 | x_2, x_3,....., x_n]P[x_2, x_3,....., x_n]$

$= P[x_1 | x_2, x_3,....., x_n]P[x_2|x_3,....., x_n]....P[x_{n-1}|x_n]P[x_n].$

In general for each variable Xi, we can write the equation as:

$P(X_i|X_{i-1},....., X_1) = P(X_i | Parents(X_i))$

Explanation of Bayesian network:

Let's understand the Bayesian network through an example by creating a directed acyclic graph:

Example: Harry installed a new burglar alarm at his home to detect burglary. The alarm reliably responds at detecting a burglary but also responds for minor earthquakes. Harry has two neighbors David and Sophia, who have taken a responsibility to inform Harry at work when they hear the alarm. David always calls Harry when he hears the alarm, but sometimes he got confused with the phone ringing and calls at that time too. On the other hand, Sophia likes to listen to high music, so sometimes she misses to hear the alarm. Here we would like to compute the probability of Burglary Alarm.

Problem:

Calculate the probability that alarm has sounded, but there is neither a burglary, nor an earthquake occurred, and David and Sophia both called the Harry.

Solution:

- The Bayesian network for the above problem is given below. The network structure is showing that burglary and earthquake is the parent node of the alarm and directly affecting the probability of alarm's going off, but David and Sophia's calls depend on alarm probability.
- The network is representing that our assumptions do not directly perceive the burglary and also do not notice the minor earthquake, and they also not confer before calling.
- The conditional distributions for each node are given as conditional probabilities table or CPT.

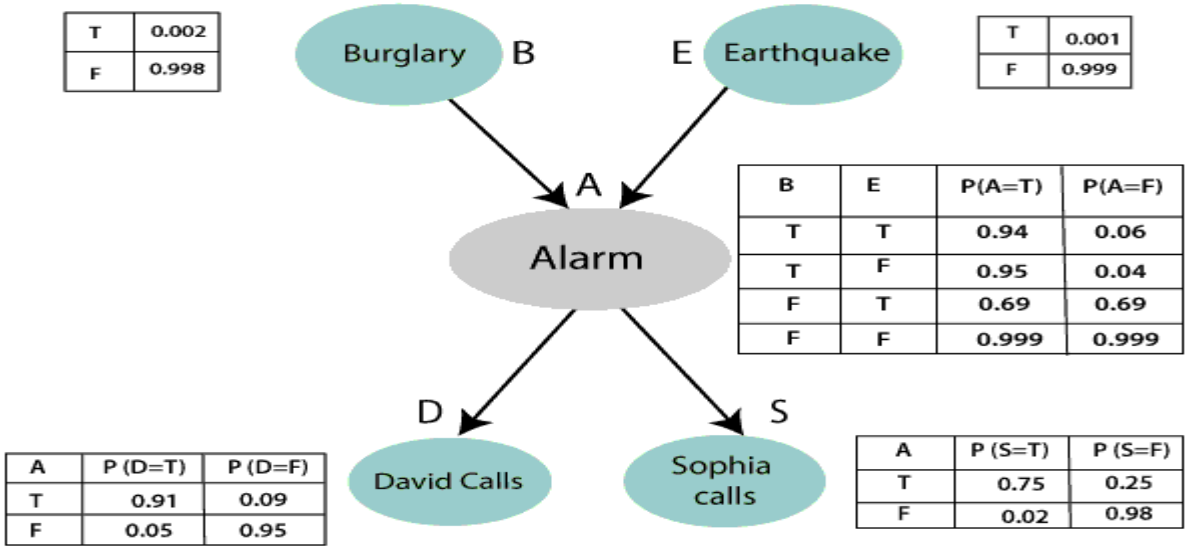
- Each row in the CPT must be sum to 1 because all the entries in the table represent an exhaustive set of cases for the variable.
- In CPT, a boolean variable with k boolean parents contains 2^K probabilities. Hence, if there are two parents, then CPT will contain 4 probability values

List of all events occurring in this network:

- **Burglary (B)**
- **Earthquake(E)**
- **Alarm(A)**
- **David Calls(D)**
- **Sophia calls(S)**

We can write the events of problem statement in the form of probability: **P[D, S, A, B, E]**, can rewrite the above probability statement using joint probability distribution:

P[D, S, A, B, E]= P[D | S, A, B, E]. P[S, A, B, E]
=P[D | S, A, B, E]. P[S | A, B, E]. P[A, B, E]
= P [D| A]. P [S| A, B, E]. P[A, B, E]
= P[D | A]. P[S | A]. P[A| B, E]. P[B, E]
= P[D | A]. P[S | A]. P[A| B, E]. P[B |E]. P[E]



T	0.002
F	0.998

T	0.001
F	0.999

B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.999	0.999

A	P (D=T)	P (D=F)
T	0.91	0.09
F	0.05	0.95

A	P (S=T)	P (S=F)
T	0.75	0.25
F	0.02	0.98

Let's take the observed probability for the Burglary and earthquake component:
P(B= True) = 0.002, which is the probability of burglary.
P(B= False)= 0.998, which is the probability of no burglary.
P(E= True)= 0.001, which is the probability of a minor earthquake
P(E= False)= 0.999, Which is the probability that an earthquake not occurred.
We can provide the conditional probabilities as per the below tables:

Conditional probability table for Alarm A:

The Conditional probability of Alarm A depends on Burglar and earthquake:

B	E	P(A= True)	P(A= False)
True	True	0.94	0.06
True	False	0.95	0.04

False	True	0.31	0.69
False	False	0.001	0.999

Conditional probability table for David Calls:
The Conditional probability of David that he will call depends on the probability of Alarm.

A	P(D= True)	P(D= False)
True	0.91	0.09
False	0.05	0.95

Conditional probability table for Sophia Calls:
The Conditional probability of Sophia that she calls is depending on its Parent Node "Alarm."

A	P(S= True)	P(S= False)
True	0.75	0.25
False	0.02	0.98

From the formula of joint distribution, we can write the problem statement in the form of probability distribution:
P(S, D, A, ¬B, ¬E) = P (S|A) *P (D|A)*P (A|¬B ^ ¬E) *P (¬B) *P (¬E).
= 0.75* 0.91* 0.001* 0.998*0.999
= 0.00068045.
Hence, a Bayesian network can answer any query about the domain by using Joint distribution.
The semantics of Bayesian Network:
There are two ways to understand the semantics of the Bayesian network, which is given below:
1. To understand the network as the representation of the Joint probability distribution.
It is helpful to understand how to construct the network.
2. To understand the network as an encoding of a collection of conditional independence statements.
It is helpful in designing inference procedure.

Bayesian Networks- representation, construction and inference,

Bayesian networks are a type of probabilistic graphical model that uses Bayesian inference for probability computations. Bayesian networks aim to model conditional dependence, and therefore causation, by representing conditional dependence by edges in a directed graph. Through these relationships, one can efficiently conduct inference on the random variables in the graph through the use of factors.
Probability
Before going into exactly what a Bayesian network is, it is first useful to review probability theory.
First, remember that the joint probability distribution of random variables A_0, A_1, …, A_n, denoted as P(A_0, A_1, …, A_n), is equal to P(A_1 | A_2, …, A_n) * P(A_2 | A_3, …, A_n) *

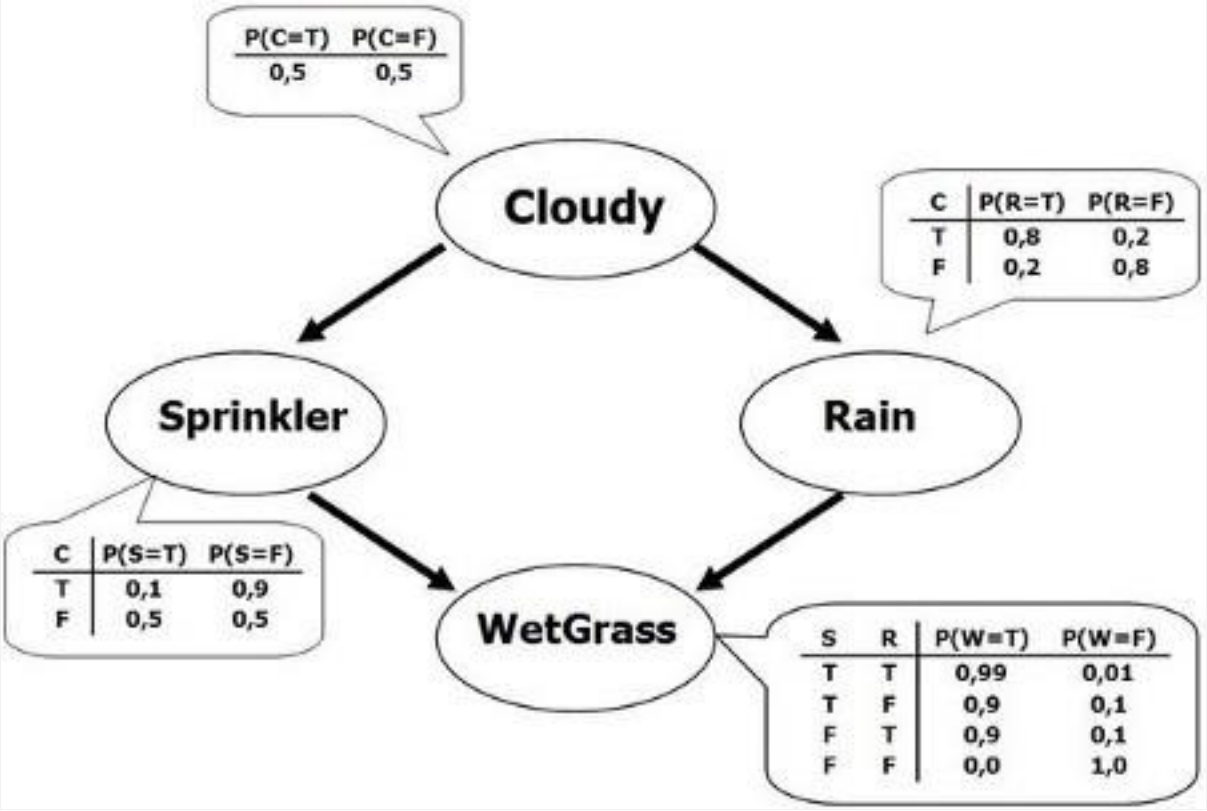
… * P(A_n) by the chain rule of probability. We can consider this a **factorized** representation of the distribution, since it is a product of N factors that are localized probabilities.

$$\mathrm{P}\left(\bigcap_{k=1}^{n} A_k\right) = \prod_{k=1}^{n} \mathrm{P}\left(A_k \,\middle|\, \bigcap_{j=1}^{k-1} A_j\right)$$

Next, recall that **conditional independence** between two random variables, A and B, given another random variable, C, is equivalent to satisfying the following property: P(A,B|C) = P(A|C) * P(B|C). In other words, as long as the value of C is known and fixed, A and B are independent. Another way of stating this, which we will use later on, is that P(A|B,C) = P(A|C).

The Bayesian Network

Using the relationships specified by our Bayesian network, we can obtain a compact, factorized representation of the joint probability distribution by taking advantage of conditional independence.



A Bayesian network is a **directed acyclic graph** in which each edge corresponds to a conditional dependency, and each node corresponds to a unique random variable. Formally, if an edge (A, B) exists in the graph connecting random variables A and B, it means that P(B|A) is a **factor** in the joint probability distribution, so we must know P(B|A) for all values of B and A in order to conduct inference. In the above example, since Rain has an edge going into WetGrass, it means that P(WetGrass|Rain) will be a factor, whose probability values are specified next to the WetGrass node in a conditional probability table.

Bayesian networks satisfy the **local Markov property**, which states that a node is conditionally independent of its non-descendants given its parents. In the above example, this means that

P(Sprinkler|Cloudy, Rain) = P(Sprinkler|Cloudy) since Sprinkler is conditionally independent of its non-descendant, Rain, given Cloudy. This property allows us to simplify the joint distribution, obtained in the previous section using the chain rule, to a smaller form. After simplification, the joint distribution for a Bayesian network is equal to the product of P(node|parents(node)) for all nodes, stated below:

$$P(X_1,\ldots,X_n)=\prod_{i=1}^{n}P(X_i \mid X_1,\ldots,X_{i-1})=\prod_{i=1}^{n}P(X_i \mid Parents(X_i))$$

In larger networks, this property allows us to greatly reduce the amount of required computation, since generally, most nodes will have few parents relative to the overall size of the network.

Inference

Inference over a Bayesian network can come in two forms.

The first is simply evaluating the joint probability of a particular assignment of values for each variable (or a subset) in the network. For this, we already have a factorized form of the joint distribution, so we simply evaluate that product using the provided conditional probabilities. If we only care about a subset of variables, we will need to marginalize out the ones we are not interested in. In many cases, this may result in underflow, so it is common to take the logarithm of that product, which is equivalent to adding up the individual logarithms of each term in the product.

The second, more interesting inference task, is to find P(x|e), or, to find the probability of some assignment of a subset of the variables (x) given assignments of other variables (our evidence, e). In the above example, an example of this could be to find P(Sprinkler, WetGrass | Cloudy), where {Sprinkler, WetGrass} is our x, and {Cloudy} is our e. In order to calculate this, we use the fact that P(x|e) = P(x, e) / P(e) = αP(x, e), where α is a normalization constant that we will calculate at the end such that P(x|e) + P(¬x | e) = 1. In order to calculate P(x, e), we must marginalize the joint probability distribution over the variables that do not appear in x or e, which we will denote as Y.

$$P(x|e)=\alpha\sum_{\forall y\in Y}P(x,e,Y)$$

For the given example, we can calculate P(Sprinkler, WetGrass | Cloudy) as follows:

$$P(Sprinkler,WetGrass\,|\,Cloudy)=\alpha\sum_{Rain}P(WetGrass|Sprinkler,Rain)P(Sprinkler|Cloudy)P(Rain|Cloudy)P(Cloudy)$$
$$=\alpha P(WetGrass|Sprinkler,Rain)P(Sprinkler|Cloudy)P(Rain|Cloudy)P(Cloudy)+\alpha P(WetGrass|Sprinkler,\neg Rain)P(Sprinkler|Cloudy)P(\neg Rain|Cloudy)P(Cloudy)$$

We would calculate P(¬x | e) in the same fashion, just setting the value of the variables in x to false instead of true. Once both P(x | e) and P(¬x | e) are calculated, we can solve for α, which equals 1 / (P(x | e) + P(¬x | e)).

Note that in larger networks, Y will most likely be quite large, since most inference tasks will only directly use a small subset of the variables. In cases like these, exact inference as shown above is very computationally intensive, so methods must be used to reduce the amount of computation. One more efficient method of exact inference is through variable elimination, which takes advantage of the fact that each factor only involves a small number of variables. This means that the summations can be rearranged such that only factors involving a given

variable are used in the marginalization of that variable. Alternatively, many networks are too large even for this method, so approximate inference methods such as MCMC are instead used; these provide probability estimations that require significantly less computation than exact inference methods.

Temporal Models

- Agents in uncertain environments must be able to keep track of the current state of the environment, just as logical agents must.
- This is difficult by partial and noisy data, because the environment is uncertain over time.
- At best, the agent will be able to obtain only a probabilistic assessment of the current situation.

Temporal Models

- Two sections in Temporal Model,
  - Time and Uncertainty
    - **States and observations**
    - **Stationary processes and the Markov assumption**
  - Inference in Temporal Model

Temporal Models - Time and Uncertainty

- A changing world is modeled using a random variable for each aspect of the environment state, *at each point in time.*
- The relations among these variables describe how the state evolves.

Example - Treating a Diabetic Patient.

- We have evidence, such as, recent insulin doses, food intake, blood sugar measurements, and other physical signs.
- The task is to assess the current state of the patient, including the actual blood sugar level and insulin level.
- Given this information, the doctor (or patient) makes a decision about the patient's food intake and insulin dose.

Example - Treating a Diabetic Patient...

- The dynamic aspects of the problem are essential.
- Blood sugar levels and measurements thereof can change rapidly over time, depending on one's recent food intake and insulin doses, one's metabolic activity, the time of day, and so on.
- To assess the current state from the history of evidence and to predict the outcomes of treatment actions, we must model these changes.

- Two sections in Temporal Model,
  - Time and Uncertainty
    - **States and observations**
    - **Stationary processes and the Markov assumption**
  - Inference in Temporal Model

States and Observations

- The process of change can be viewed as a series of snapshots (results), describes the state of the world at a particular time.
- Each snapshot or time slice, contains a set of random variables, some of which are observable and some of which are not.

State and observation ...

- We will assume that the same subset of variables is observable in each slice
- X_t – set of unobservable state variable at time t
- E_t – set of observable evidence variable
- The observation at time t is $E_t = e_t$ for some set of values e_t

State and observation ...

- Example: **Umbrella and Rain**
- Suppose you are a security guard for some secret underground installation
- You want to know whether it is raining today
- But, your only access to the outside world occurs each morning, when you see the director coming in with, or without an umbrella.

Example: Umbrella and Rain...

- For each day t, the set E_t (observable evidence variables) thus contains a single evidence variable U_t (whether the umbrella appears) and the set X_t (unobservable state variable) contains a single state variable R_t (whether raining or not)
- Hence we can assume Et = Ut and Xt = Rt
- And if Et = true then Xt = true
- i.e. if Ut = true then Rt = true

State and observation ...

- The interval between time slices also depends on the problem.
- For diabetes monitoring, a suitable interval might be an hour rather than a day.
- We generally assume a fixed, finite interval; this means that times can be labeled by integers.
- We will assume that evidence starts arriving at t = 1 rather than t = 0.
- Hence, our umbrella world is represented by state variables Ro will be, R1, R2, . . . and evidence variables *U1* , U2, ...
- We will use the notation a:b to denote the sequence of integers from *a* to b and the notation Xa:b to denote the corresponding set of variables from Xa, to Xb.
- For example, $U_{1:3}$ corresponds to the variables U1, U2, U3.

- Two sections in Temporal Model,
  - Time and Uncertainty
    - **States and observations**
    - Stationary processes and the Markov assumption
  - Inference in Temporal Model

Stationary Processes and the Markov assumption

- With the set of state and evidence variables for a given problem, we need to specify the dependencies among the variables.
- Order the variables in their natural temporal order
- Since cause usually precedes effect so we need to add the variables in causal order.

Stationary Processes and the Markov assumption...

- The set of variables is unbounded, because it includes the state and evidence variables for every time slice.
- This actually creates two problems:
  - **first**, we might have to specify an unbounded number of conditional probability tables (CPT), one for each variable in each slice;
  - **second,** each one might involve an unbounded number of parents.

Stationary Processes and the Markov assumption...

- Solution for the problems
- The first problem is solved by assuming that changes in the world state are caused by a stationary process-that is, a process of change that is governed by laws that do not themselves change over time.
- In the umbrella world, the conditional probability that the umbrella appears, P(Ut | Parents(Ut)), is the same for all t.

Stationary Processes and the Markov assumption...

- The second problem, handling the infinite number of parents, is solved by making a Markov assumption-that is, that the current state depends on only a finite history of previous states.
- the simplest is the first-order Markov process
- **in which the current state depends only on the previous state and not on any earlier states.**
- Using our notation, the corresponding conditional independence assertion states that, for all *t*,

$$\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$$

Stationary Processes and the Markov assumption...

- The transition model for a second-order Markov process is the conditional distribution $P(X_t \mid X_{t-2}, X_{t-1})$.
- current state depends on only two previous states

Stationary Processes and the Markov assumption...

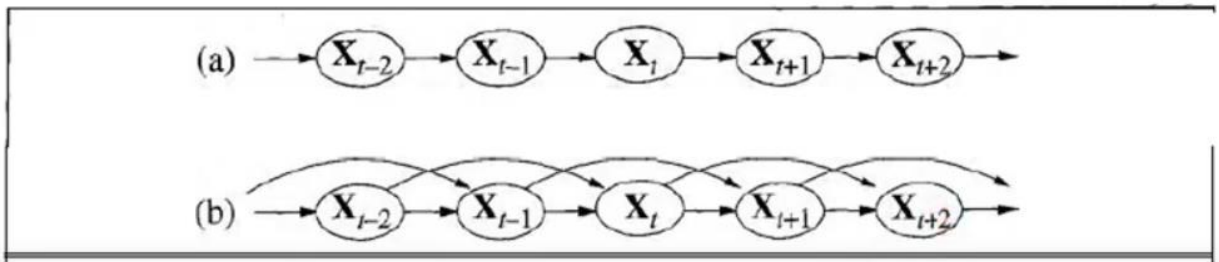


Figure 15.1 (a) Bayesian network structure corresponding to a first-order Markov process with state defined by the variables $\mathbf{X}_t$. (b) A second-order Markov process.

Stationary Processes and the Markov assumption...

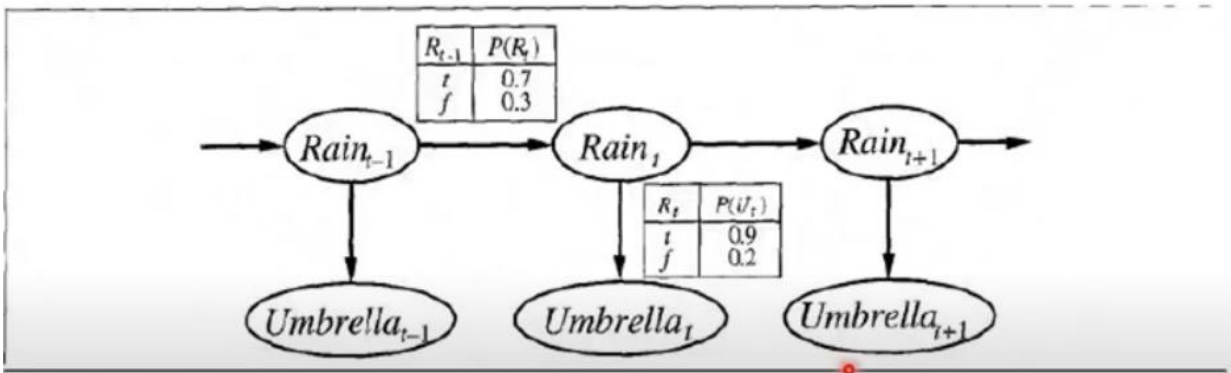


Figure 15.2 Bayesian network structure and conditional distributions describing the umbrella world. The transition model is $P(Rain_t|Rain_{t-1})$ and the sensor model is $P(Umbrella_t|Rain_t)$.

Hidden Markov Model

Hidden Markov Models or HMMs are the most common models used for dealing with temporal Data. They also frequently come up in different ways in a Data Science Interview usually without the word HMM written over it. In such a scenario it is necessary to discern the problem as an HMM problem by knowing characteristics of HMMs.

In the Hidden Markov Model we are constructing an inference model based on the assumptions of a Markov process.

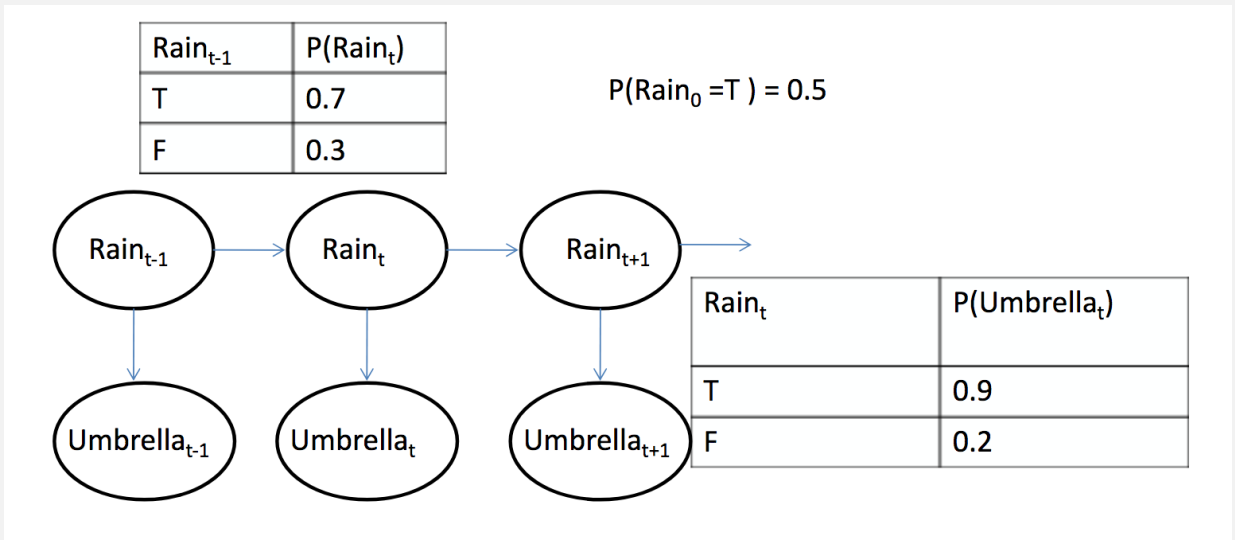
The Markov process assumption is that the "*future is independent of the past given that we know the present*".

It means that the future state is related to the immediately previous state and not the states before that. These are the first order HMMs.

What is Hidden?

With HMMs, we don't know which state matches which physical events **instead** each state matches a given output. We observe the output over time to determine the sequence of states.

Example: If you are staying indoors you will be dressed up a certain way. Lets say you want to step outside. Depending on the weather, your clothing will change. Over time, you will observe the weather and make better judgements on what to wear if you get familiar with the area/climate. In an HMM, we observe the outputs over time to determine the sequence based on how likely they were to produce that output.



HMMs — Adapted from Russel and Norvig, Chapter 15.

Let us consider the situation where you have no view of the outside world when you are in a building. The only way for you to know if it is raining outside it so see someone carrying an umbrella when they come in. Here, the evidence variable is the *Umbrella*, while the hidden variable is *Rain*. See the probabilities in the diagram above.

$$P(\mathrm{R}_0, \mathrm{R}_1, \ldots, \mathrm{R}_t, U_0, U_1, \ldots, U_t) = P(R_0)\prod_{i=1}^{t} P(R_i \mid R_{i-1})P(U_i \mid R_i)$$

HMM representation

Since this is a Markov model, *R(t)* depends only on *R(t-1)*

A number of related tasks ask about the probability of one or more of the latent variables, given the model's parameters and a sequence of observations which is sequence of *umbrella* observations in our scenario.

Hidden Markov Models: Matrix Representations

- **Transition model:** $P(X_t \mid X_{t-1}) = \mathbf{T}$ ($S \times S$ matrix) where

$$\mathbf{T}_{i,j} = P(X_t = j \mid X_{t-1} = i)$$

- For umbrella model:

$$\mathbf{T} = P(X_t \mid X_{t-1}) = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{matrix} T \\ F \end{matrix} \quad \begin{matrix} T & F \end{matrix}$$

- **Sensor model:** $P(e_t \mid X_t = i) = \mathbf{O}$ ($S \times S$ diagonal matrix) where

$$\mathbf{O}_{i,j} = \begin{cases} P(e_t \mid X_t = i), i = j \\ 0 \text{ otherwise} \end{cases}$$

- For umbrella model:

$$\mathbf{O} = P(e_t \mid X_t) = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$$

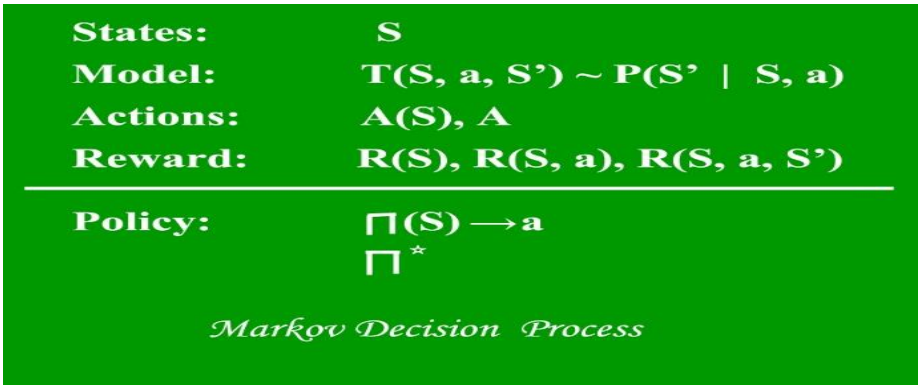
Markov Decision Process

Reinforcement Learning is a type of Machine Learning. It allows machines and software agents to automatically determine the ideal behavior within a specific context, in order to maximize its performance. Simple reward feedback is required for the agent to learn its behavior; this is known as the reinforcement signal.

There are many different algorithms that tackle this issue. As a matter of fact, Reinforcement Learning is defined by a specific type of problem, and all its solutions are classed as Reinforcement Learning algorithms. In the problem, an agent is supposed to decide the best action to select based on his current state. When this step is repeated, the problem is known as a **Markov Decision Process**.

A **Markov Decision Process (MDP)** model contains:

- A set of possible world states S.
- A set of Models.
- A set of possible actions A.
- A real valued reward function R(s,a).
- A policy the solution of **Markov Decision Process**.



What is a State?

A **State** is a set of tokens that represent every state that the agent can be in.

What is a Model?

A **Model** (sometimes called Transition Model) gives an action's effect in a state. In particular, T(S, a, S') defines a transition T where being in state S and taking an action 'a' takes us to state S' (S and S' may be same). For stochastic actions (noisy, non-deterministic) we also define a probability P(S'|S,a) which represents the probability of reaching a state S' if action 'a' is taken in state S. Note Markov property states that the effects of an action taken in a state depend only on that state and not on the prior history.

What is Actions?

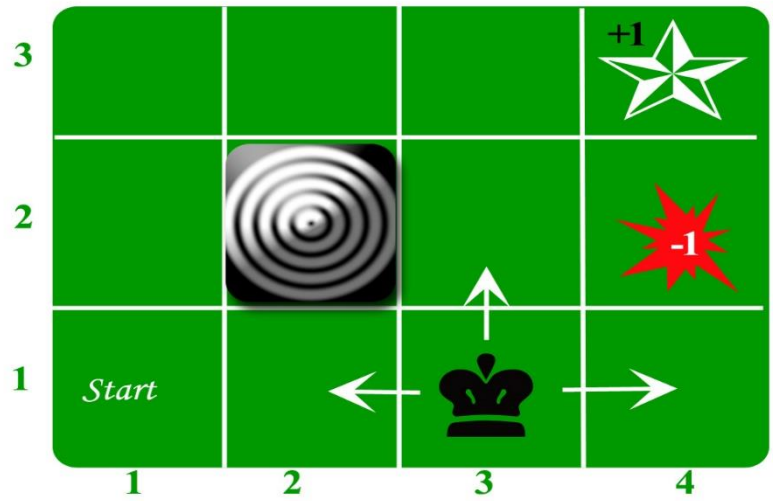
An **Action** A is set of all possible actions. A(s) defines the set of actions that can be taken being in state S.

What is a Reward?

A **Reward** is a real-valued reward function. R(s) indicates the reward for simply being in the state S. R(S,a) indicates the reward for being in a state S and taking an action 'a'. R(S,a,S') indicates the reward for being in a state S, taking an action 'a' and ending up in a state S'.

What is a Policy?

A **Policy** is a solution to the Markov Decision Process. A policy is a mapping from S to a. It indicates the action 'a' to be taken while in state S.
Let us take the example of a grid world:



An agent lives in the grid. The above example is a 3*4 grid. The grid has a START state(grid no 1,1). The purpose of the agent is to wander around the grid to finally reach the Blue Diamond (grid no 4,3). Under all circumstances, the agent should avoid the Fire grid (orange color, grid no 4,2). Also the grid no 2,2 is a blocked grid, it acts like a wall hence the agent cannot enter it.

The agent can take any one of these actions: **UP, DOWN, LEFT, RIGHT**
Walls block the agent path, i.e., if there is a wall in the direction the agent would have taken, the agent stays in the same place. So for example, if the agent says LEFT in the START grid he would stay put in the START grid.

First Aim: To find the shortest sequence getting from START to the Diamond. Two such sequences can be found:

- **RIGHT RIGHT UP UP RIGHT**
- **UP UP RIGHT RIGHT RIGHT**

Let us take the second one (UP UP RIGHT RIGHT RIGHT) for the subsequent discussion. The move is now noisy. 80% of the time the intended action works correctly. 20% of the time the action agent takes causes it to move at right angles. For example, if the agent says UP the probability of going UP is 0.8 whereas the probability of going LEFT is 0.1 and probability of going RIGHT is 0.1 (since LEFT and RIGHT is right angles to UP).

The agent receives rewards each time step:-

- Small reward each step (can be negative when can also be term as punishment, in the above example entering the Fire can have a reward of -1).
- Big rewards come at the end (good or bad).
- The goal is to Maximize sum of rewards.

****