Module 4 - Knowledge and Reasoning

Explain Bayesian Belief Network or Belief Network in Detail

●Bayesian Belief Network or Bayesian Network or Belief Network is a Probabilistic Graphical Model (PGM) that represents conditional dependencies between random variables through a Directed Acyclic Graph (DAG)

● Bayesian networks are probabilistic, because these networks are built from a probability

What do we use the Bayesian Networks for?

● Bayesian Networks are applied in many fields. For example, disease diagnosis, optimized web search, spam filtering, gene regulatory networks, etc. And this list can be extended.

● The main objective of these networks is trying to understand the structure of causality relations.

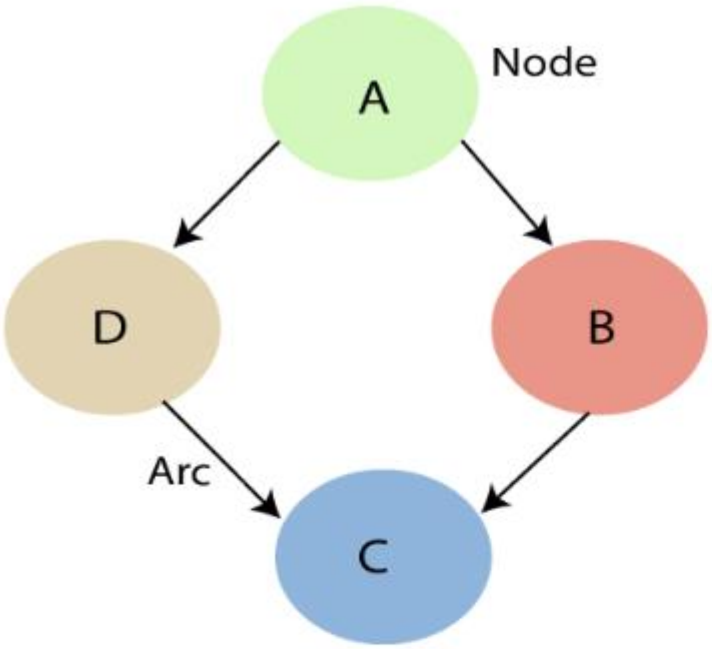
● To clarify this, let's consider a disease diagnosis problem. With given symptoms and their resulting disease, we construct our Belief Network and when a new patient comes, we can infer which disease or diseases may have the new patient by providing probabilities for each disease.

● Similarly, these causality relations can be constructed for other problems and inference techniques can be applied to interesting results

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

Note: The Bayesian network graph does not contain any cyclic graph. Hence, it is known as a directed acyclic graph or DAG

The main components of a Bayesian Network are:

1. Nodes: Nodes represent variables in the Bayesian Network. Each node is assigned a name and a probability distribution that describes the conditional probabilities of the variable given its parents in the DAG.

2. Edges: Edges represent the probabilistic dependencies between variables in the Bayesian Network. Directed edges indicate causality, where a parent node influences the value of its child node.

3. Conditional probability tables (CPTs): CPTs are tables associated with each node in the network that define the conditional probability distribution of the node given the values of its parents.

4. Evidence: Evidence refers to any available information about the values of variables in the network. Evidence can be used to update the probabilities of variables in the network and make inferences.

5. Inference algorithms: Inference algorithms are used to perform probabilistic inference in the network, such as computing the probability of a variable given evidence or determining the most likely explanation for a set of observations.

6. Learning algorithms: Learning algorithms are used to learn the structure and parameters of the Bayesian Network from data. This includes both learning the structure of the DAG and estimating the CPTs for each node in the network.

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

Note : "Burglar" is a term used to refer to a person who illegally enters a building, typically a home, in order to steal property or commit a crime

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 neighbours 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.

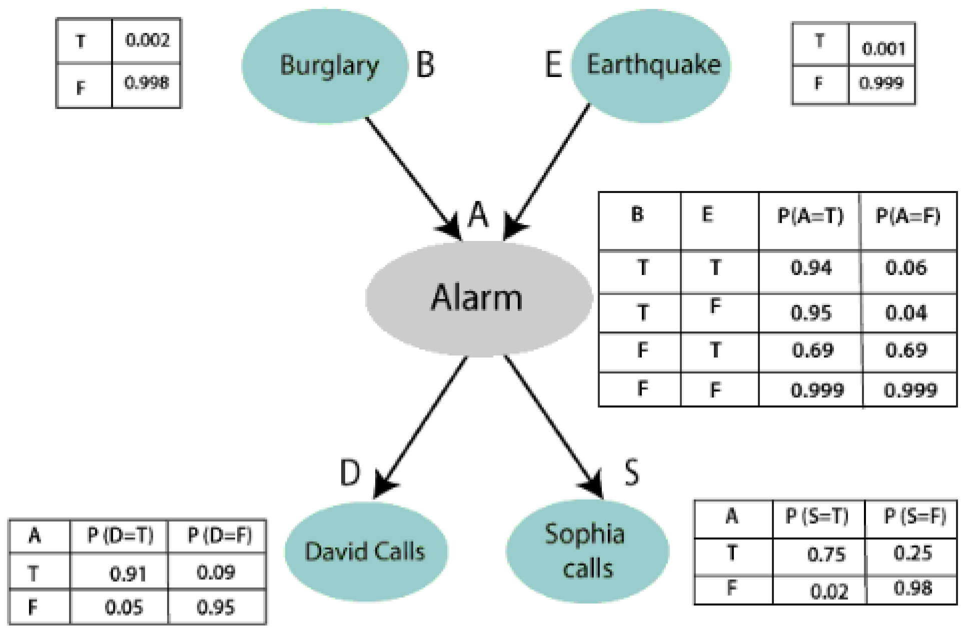
○ In CPT, a boolean variable with k boolean parents contains 2K probabilities. Hence, if there are two parents, then CPT will contain 4 probability values

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]

Let's take the observed probability for the Burglary and earthquake component:



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 table

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

Define Belief Network. Describe the steps of constructing a belief network with an example. What types of inferences can be drawn from that?

A belief network, also known as a Bayesian network or probabilistic graphical model, is a graphical representation of the probabilistic relationships among a set of variables. It is a type of directed acyclic graph (DAG) where each node represents a random variable and edges between nodes represent conditional dependence relationships.

The construction of a belief network involves the following steps:
1. Identify the variables and their relationships: Determine the set of variables that are relevant to the problem and how they are related to each other. This can be done through expert knowledge, data analysis, or a combination of both.

2. Define the structure of the network: Once the variables and their relationships are identified, the network structure can be defined. Each variable is represented by a node, and the relationships among the variables are represented by directed edges.

3. Assign probabilities to the nodes: Assign probabilities to each node in the network based on expert knowledge or data analysis. These probabilities represent the likelihood of each variable taking on a specific value given the values of its parent nodes.

4. Validate and refine the network: Test the network against real-world data to ensure that it accurately represents the relationships among the variables. Refine the network as necessary based on the results of the validation.

For example, consider a belief network for predicting the likelihood of a person having a heart attack based on their age, gender, and cholesterol levels. The variables in this network would be age, gender, cholesterol levels, and the likelihood of having a heart attack. The network structure would have directed edges connecting age and gender to the likelihood of having a heart attack, and cholesterol levels to the likelihood of having a heart attack. Probabilities would be assigned to each node based on expert knowledge or data analysis.
In this DAG, the variables are represented by nodes, and the directed edges between the nodes indicate the causal relationships between the variables. For example, age and gender are parents of cholesterol levels, and cholesterol levels is a parent of the likelihood of having a heart attack. This indicates that age and gender have a direct effect on cholesterol levels, and cholesterol levels has a direct effect on the likelihood of having a heart attack.

Once a belief network is constructed, various types of inferences can be drawn from it, including:

1. Marginal inference: We can use the network to figure out the chance of one thing happening, like the chance of a person having a heart attack, based on what we know about other things, like their age and gender.

2. Joint inference: We can use the network to figure out the chance of all the things happening together, like the chance of a person having a heart attack, given their age, gender, and cholesterol levels.

3. Conditional inference: We can use the network to figure out the chance of something happening if we know certain things are true, like the chance of a person having a heart attack if we know their age and gender.

```
        Age        Gender
         |           |
         v           v
   Cholesterol Levels
         |
         v
   Likelihood of Heart Attack
```

In this DAG, the variables are represented by nodes, and the directed edges between the nodes indicate the causal relationships between the variables. For example, age and gender are parents of cholesterol levels, and cholesterol levels is a parent of the likelihood of having a heart attack. This indicates that age and gender have a direct effect on cholesterol levels, and cholesterol levels has a direct effect on the likelihood of having a heart attack.

Once a belief network is constructed, various types of inferences can be drawn from it, including:

1. Marginal inference: We can use the network to figure out the chance of one thing happening, like the chance of a person having a heart attack, based on what we know about other things, like their age and gender.

2. Joint inference: We can use the network to figure out the chance of all the things happening together, like the chance of a person having a heart attack, given their age, gender, and cholesterol levels.

3. Conditional inference: We can use the network to figure out the chance of something happening if we know certain things are true, like the chance of a person having a heart attack if we know their age and gender.

4. Prediction: We can use the network to make predictions about what might happen in the future, like predicting if someone is likely to have a heart attack based on what we know about their age, gender, and cholesterol levels.

5. Explanation: We can use the network to understand how different things are connected and how they contribute to a certain outcome, like how age, gender, and cholesterol levels are related to the likelihood of having a heart attack

Define Belief Network. Explain conditional Independence relation in Belief Network with example

A Belief Network is a type of probabilistic graphical model that helps us understand and reason about uncertain situations. It consists of a collection of nodes that represent different variables or factors, and the relationships or dependencies between them.

Conditional independence is a key concept in Belief Networks that allows us to simplify the structure of the network and make it easier to reason about.

● In a Belief Network, two nodes are said to be conditionally independent given a set of evidence nodes if the probability of one node does not depend on the value of the other node once we know the values of the evidence nodes. In other words, the nodes are independent of each other given the evidence.

● Let's consider a simple example of a Belief Network that models the relationship between smoking and lung cancer.

● The network includes two nodes: one for smoking and one for lung cancer. Let's say that we know from previous studies that smoking is a risk factor for lung cancer, but there are also other risk factors such as exposure to air pollution and genetics.

● Now, let's say we introduce a new node for air pollution. If we know the level of air pollution in a particular area, it may affect the risk of lung cancer directly, but it may also affect the relationship between smoking and lung cancer. If the air pollution is very high, smoking may have a stronger effect on the risk of lung cancer, but if the air pollution is low, smoking may not have as much of an effect.

● In this case, the smoking and lung cancer nodes are no longer conditionally independent given the air pollution node. However, if we know both the smoking status and the genetics of an individual, the air pollution node does not provide any new information that would affect the relationship between smoking and lung cancer. In this case, the smoking and lung cancer nodes are still conditionally independent given the genetics node.

● By identifying conditional independence relationships in a Belief Network, we can simplify the structure of the network and make it easier to calculate probabilities and make predictions.

```
 Smoking      Air Pollution
    |               |
    v               v
Lung Cancer <--- Genetics
```

Direct Acyclic Graph (DAG) of the Give Example

● In this DAG, the nodes are represented by circles, and the arrows between the nodes indicate the causal relationships between them. The arrows point from the cause to the effect, indicating the direction of influence.

● So, in this example, smoking and air pollution are the two causes of lung cancer, while genetics may also play a role but is not directly influenced by either smoking or air pollution. This DAG shows the causal structure of the Belief Network, which we can use to calculate probabilities and make predictions about the relationships between the variables

Conditional probability and its role in AI

1 Conditional probability is the probability of an event occurring given that another event has already occurred.

2. In AI, conditional probability is often used to model the relationships between different variables in a system

3. Consider a simple scenario where a student's performance in a math test depends on two factors: whether they attended a tutoring session (T) and the number of hours they studied(H).

4. The probability of a student performing well on the math test is uncertain, but we can use conditional probability to model this relationship.

5. Let's say we know the probability of a student attending a tutoring session (T) is 0.6, and the probability of them studying for more than 3 hours (H) is 0.5.

6. We can use conditional probability to calculate the probability of a student performing well on the math test given that they attended a tutoring session and studied for more than 3 hours.

7. By using the conditional probability formula, we can calculate this probability as:
P(perform well | T and H) = P(T and H | perform well) * P(perform well) / P(T and H)

6. This information can be used to make predictions and decisions about the student's performance. For example, a school might use this information to identify students who are at risk of performing poorly and provide them with additional support.

7. Bayesian Networks are often used in AI to model relationships between variables and make predictions based on the available evidence. They rely heavily on conditional probability to calculate the probabilities of different outcomes based on the available evidence.

8. Overall, conditional probability is a fundamental concept in AI that is used to model uncertain situations and make predictions based on the available evidence.

Propositional Logic

1.Propositional logic is a way of representing knowledge in AI using logical statements.
2. These statements, called propositions, are either true or false.
3. Propositions can be combined using logical operators such as AND, OR, and NOT.
4. For example, the proposition "It is raining" can be combined with "I have an umbrella" using the AND operator to form the compound proposition "It is raining AND I have an umbrella."
5. Propositional logic can be used to represent simple knowledge, but it has limitations when it comes to representing more complex knowledge.
6. Propositional logic can also be used to create rules that allow an AI system to make decisions or draw conclusions based on available information.
7. One common application of propositional logic in AI is in expert systems, which use a set of rules to solve problems in a specific domain.

Syntax

● Propositional logic (PL), in order to be effective, then we need to follow a language structure that should be agreed upon by everyone, and it should be easy to adopt by all. PL Language structure consists of simple undividable statements joined together with logical connectors.

● A sentence in any language contains a combination of words like the verb, noun, pronoun, prepositions, etc., Syntax of PL Language also follows a similar rule, and it consists of

Sr. No	Subject	Syntax
1	Simple undividable statement represent true or false (not both) and it is Boolean in nature	Upper Case letters A, B, C, P, Q, R are used to represent statements
2	Logical Connectors or operators used to connect two statements	^, v, →, ↔, ¬ are used to represent AND, OR, Implies, bi-conditional and NOT condition.
3	Complex conditions	Complex conditions are handled by coding connectors within parenthesis.

Other Features

● A simple sentence is called Atomic Proposition, and it should be either true or false. Example: 9+2 =11 is one such proposition, and it is true. Sunrises in the west is another example, and it is false.
● A combination of simple sentences connected by logical connectors is called Compound. Example: Today is Friday and people visit Temple today. It's raining, and the match is called off.
● A Proposition that is always true is known as Tautology (another name for Valid Sentence).
● A Proposition that is always false is known as Contradiction.
● Sentences that are questions and command in nature do not belong to this Proposition Category

Logical Connectives

It connects two undividable simple sentences or expresses a sentence in a logical sense. Complex statements can be created using logical connectives. There are 5 types of connectors, namely

Sl.	Type	Symbol	Description
1	Negation	$\neg P$	It represents a Negative condition. P is a positive statement, and $\neg P$ indicates NOT condition. Example: Today is Monday (P), Today is not a Monday ($\neg P$)
2	Conjunction	P ^ Q	It joins two statements P, Q with AND clause. Example: Ram is a cricket player (P). Ram is a Hockey player (Q). Ram plays both cricket and Hockey is represented by (P ^ Q)
3	Disjunction	P v Q	It joins two statements P, Q with OR Clause. Example: Ram leaves for Mumbai (P) and Ram leaves for Chennai (Q). Ram leaves for Chennai or Mumbai is represented by (P v Q). In this complex statement, at any given point of time if P is True Q is not true and vice versa.
4	Implication	$P \rightarrow Q$	Sentence (Q) is dependent on sentence (P), and it is called implication. It follows the rule of If then clause. If sentence P is true, then sentence Q is true. The condition is unidirectional. Example: If it is Sunday (P) then I will go to Movie (Q), and it is represented as $P \rightarrow Q$
5	Bi-conditional	$P \Leftrightarrow Q$	Sentence (Q) is dependent on sentence (P), and vice versa and conditions are bi-directional in this connective. If a conditional statement and its converse are true, then it is called as bi-conditional connective (Implication condition in both the directions $P \rightarrow Q$ and $Q \rightarrow P$). If and only if all conditions are true, then the end statement is true. Example: If I have 1000 Rupees then only I will go to Bar. The converse condition that I will go to Bar if and only if I have Rs 1000. The first statement covers necessity and the second one covers sufficiency.

Due to its ability in solving complex problems this logic is used quite extensively in Business, Education and Medical fields. But it has some limitations viz.,

1. It cannot address relations like Some, ALL,
2. It can neither handle logical relationships.
3. It has limited expressive ability

Steps involved in converting the propositional logic statement into CNF with a suitable example.

PROPOSITIONAL LOGIC:

1. Propositional Logic is the simplest logic.
2. All higher order logic is based on it.
3. It has less expression abilities.
4. The world is expressed as facts.
5. Fact can either be true or false.

CAUSAL NORMAL FORM:

1. It is also known as Conjunctive Normal Form.
2. CNF is the simplest representation of sentences.
3. CNF is the easy to work for compute

Steps to Convert PL into CNF

To convert a propositional logic statement into CNF, we need to follow a set of steps, which are as follows:

a. Eliminate any implications in the statement by replacing them with an equivalent logical statement using the material implication operator.

i. Remove Implication or eliminate all→ connectives

(P → Q) => (P ^ ¬ Q)

ii. Convert bi-condition into implications or eliminate all ↔ connectives.

(P ↔ Q) => ((P → Q) ^ (Q → P))

b. Use De Morgan's laws to move any negations inside the brackets of any compound propositions.

¬ (¬ P) => P
¬ (P ^ Q) => ¬ P v ¬ Q
¬ (P v Q) => ¬ P ^ ¬ Q
¬ (∀ x)P ⇒ (∃ x) ¬ P
¬ (∃ x)P ⇒ (∀ x) ¬ P

c. Apply Distributive and Commutative Law

Distribute any disjunctions over conjunctions using the distributive property of logical operators.

P ^ (Q v R) => (P ^ Q) v (P ^ R) ------ [Distributive Law]
P v (Q ^ R) => (P v Q) ^ (P v R) ------ [Distributive Law]
P ^ Q => Q ^ P ------ [Commutative Law]
P v Q => Q v P ------ [Commutative Law]

Resolution Technique or Explain Resolution by Refutation

1. Resolution by refutation is a logical inference technique used in artificial intelligence and automated theorem proving to establish the validity or satisfiability of a given set of logical statements.

2. The method starts by assuming that the negation of the statement we want to prove (the goal) is true.

3. The goal is then transformed into clausal form, which is a set of clauses where each clause is a disjunction of literals (propositional variables or their negations).

4. The resolution algorithm attempts to refute the negation of the goal by deriving a contradiction from the set of clauses.

5. The algorithm does this by applying the resolution rule, which states that if two clauses contain complementary literals (i.e., one is the negation of the other), then we can derive a new clause that is the union of the remaining literals.

6. The process of applying the resolution rule continues until either a contradiction is derived (an empty clause) or it is determined that the goal cannot be refuted.

7. If an empty clause is derived, then the original goal is proven to be true.

8. If it is determined that the goal cannot be refuted, then the negation of the goal is proven to be true.

9. Resolution by refutation is a powerful technique that allows us to automate the process of logical inference and derive new knowledge from existing statements.

10. It is a key component of automated theorem proving and is widely used in areas such as natural language processing and knowledge representation

Forward Chaining :

1. Forward chaining is a bottom-up approach to reasoning used in artificial intelligence.

2. The method starts with a set of facts or initial data and iteratively applies a set of production rules to derive new facts.

3. The production rules are in the form of condition-action statements, where the condition is a set of antecedents and the action is a set of consequents.
4. The antecedents are evaluated against the set of known facts, and if they are all true, then the consequents are added to the set of known facts.
5. This process continues until either a goal is reached or no new facts can be derived.
6. Forward chaining is particularly useful in domains where the knowledge is highly structured and the rules are well-defined.
7. It is commonly used in expert systems, decision support systems, and data mining applications

Backward Chaining:
1. Backward chaining is a top-down approach to reasoning used in artificial intelligence.
2. The method starts with a goal or a query and works backward through a set of production rules to determine the set of facts needed to satisfy the goal.
3. The production rules are in the form of condition-action statements, where the condition is a set of antecedents and the action is a set of consequents.
4. The antecedents are evaluated against the set of known facts, and if they are all true, then the action is performed.
5. This process continues recursively until either the goal is reached or no further actions can be taken.
6. Backward chaining is particularly useful in domains where the knowledge is less structured and the rules are less well-defined.
7. It is commonly used in diagnostic systems, planning systems, and natural language processing applications.
8. One limitation of backward chaining is that it may require significant computational resources to search through all possible paths of the production rules, which can be problematic in domains with large or complex knowledge bases

Criteria	Forward Chaining	Backward Chaining
Approach	Bottom-up	Top-down
Starting point	Given facts	Query/goal
Execution	Iterative application of production rules	Recursive application of production rules
Knowledge base	Highly structured, well-defined rules	Less structured, less well-defined rules
Goal	Derive new facts from known data	Determine facts needed to satisfy goal
Application	Expert systems, decision support systems, data mining	Diagnostic systems, planning systems, natural language processing
Computational resources	Less computational resources needed	May require significant computational resources
Efficiency	More efficient when data is highly structured	May be less efficient when data is less structured
Examples	Inference engine, Rule-based system	Expert system, Planning system, Natural Language Processing