UNIT – 5
21CSC206T/ ARTIFICIAL INTELLIGENCE

Planning and Learning
Table of Contents

- Planning – planning problem – Simple planning agent – Blocks world problem – Mean Ends analysis Learning
- Machine learning - Learning concepts, methods and models
- Introduction to expert system – architecture of expert systems.

What is Planning?

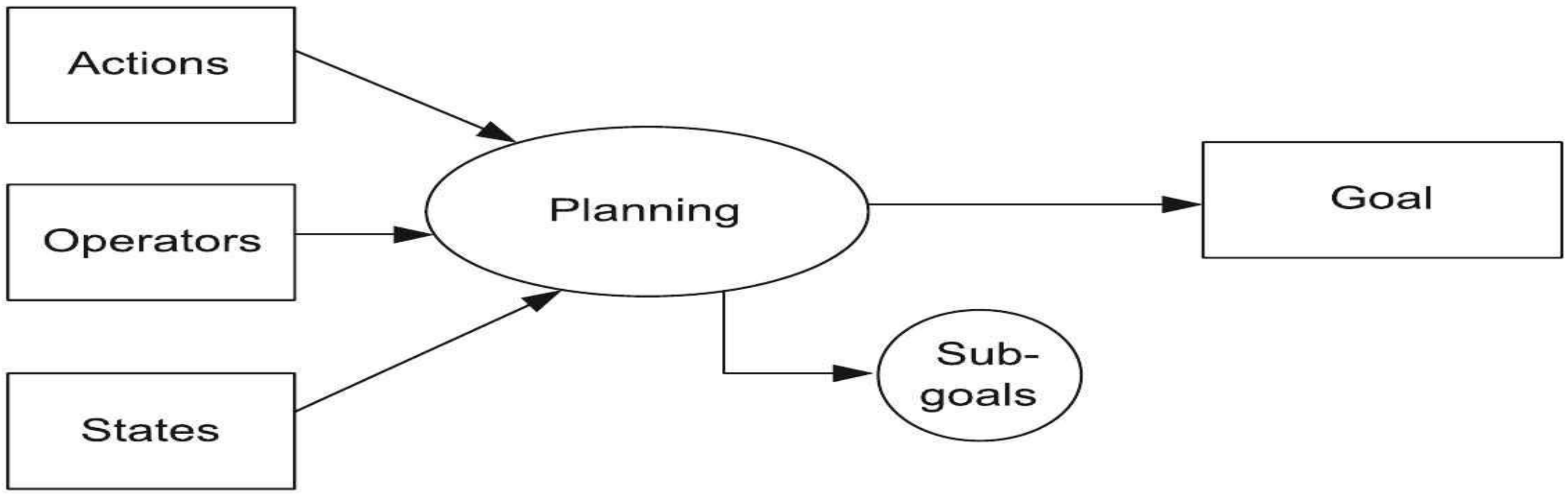
- **Planning:**
  - The task of coming up with a sequence of actions that will achieve a goal is called planning.

- **Planning Environments**
  1. Classical Planning Environments
     - Fully observable, deterministic, finite, static and discrete.
  2. Non classical Planning Environments
     - Partially observable, stochastic with different algorithms and agent designs.

PLANNING PROBLEM

The planning problem is actually the question how to go to next state or the goal state from the current state. It involves two things 'how' and 'when'.

To plan, one should be able to represent the problem properly. Representation of the planning problem is mapping of the states, actions and the goals. For the representation, the language used should be concrete, understandable and expressive. In a broader perspective, there are different representation methods or ways that are followed like propositional, first order, state variable. Figure 9.1 depicts the high-level diagram for planning.



Simple Planning Agent

- A simple planning agent

makes use of knowledge and problem solving to get the goal. It performs the following steps:

1. Defining a goal (a goal is initially set by the agent).
2. Planning (a plan is built).
3. Taking action (actions are invoked as per the plan).

- ❖ Planning involves reasoning about actions that the agent intends to carry out
- ❖ This reasoning involves the representation of the world that the agent has - representation of its actions. It may have
  - Hard constraints - objectives *have to* be achieved completely for success
  - soft constraints, or preferences, to be achieved as much as possible

Simple Planning Agent

Assumptions of agent

1. The actions occur one at a time (no parallel execution) and that they cannot be further divided.
2. There is no uncertainty and the agent is well-versed with the outcome that is specified with the action.
3. Rules/things which are not specified are assumed to be false.
4. The agent's action causes the changes in the state and it has the knowledge about it. So, the environment is fully observable.

Planning Languages

To represent planning problems we use **Artificial Intelligence planning languages** that describe environment's conditions which then lead to desired goals by generating chain of actions based on these conditions.

STRIPS (STanford Research Institute Problem Solver) -an action language which was a part of the first major planning system with the same name.

ADL (Action Description Language) is one of STRIPS extension which supports negative literals, quantified variables in goals (e.g. $\exists x$ At (P1, x) $\wedge$ At(P2, x)), conditional effects and disjunctions in goals (all not allowed in STRIPS)

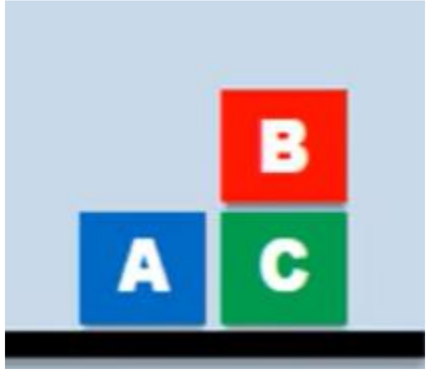
PDDL (Planning Domain Definition Language). - an attempt to standardize planning languages (STRIPS, ADL and much more other representational languages)

Block World - Using STRIPS – Goal Stack Planning

What is the Blocks World? --

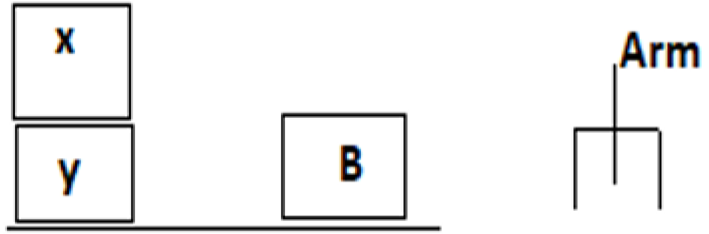
- A flat surface such as a table
- An adequate set of identical blocks which are identified by letters.
- The blocks can be stacked one on one to form towers of apparently unlimited height.
- The stacking is achieved using a robot arm which has fundamental operations and states which can be assessed using logic and combined using logical operations.
- The robot can hold one block at a time and only one block can be moved at a time.

A **STRIPS planning problem** specifies:
1) an initial state *S*
2) a goal *G*
3) a set of STRIPS actions

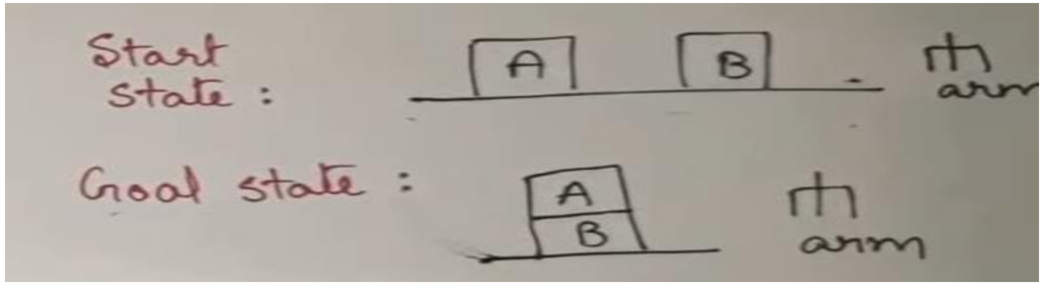


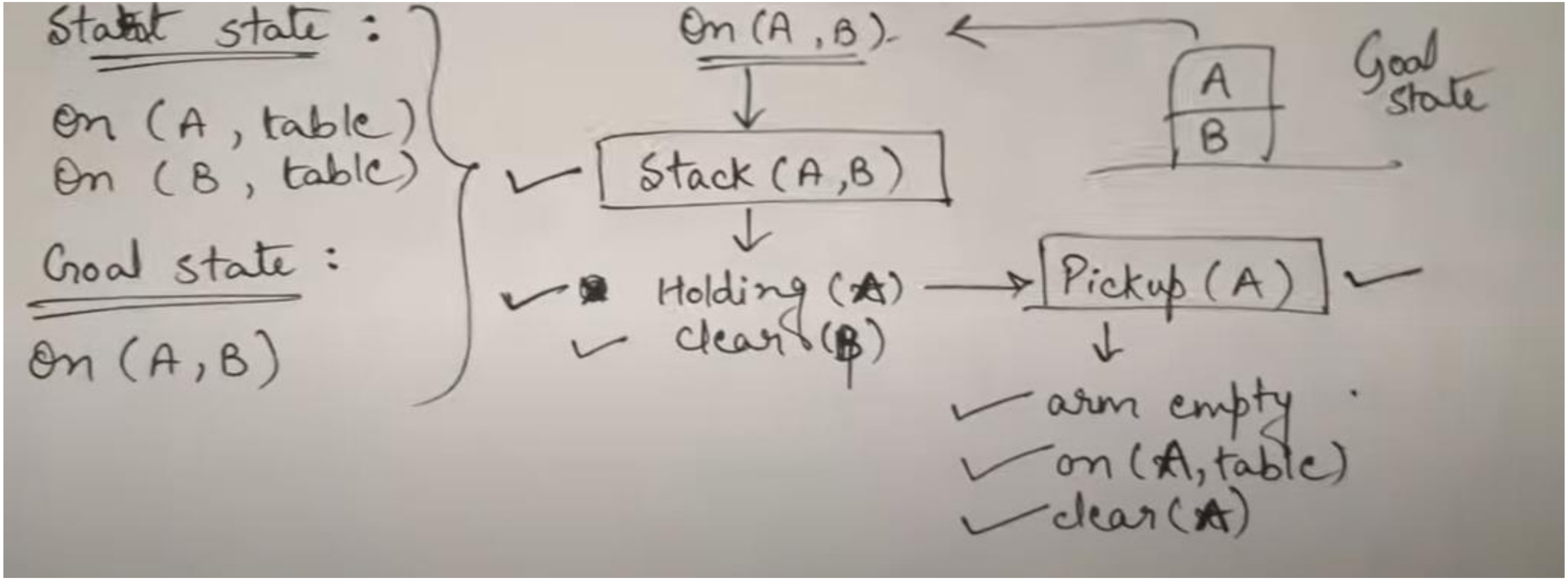
Block World Problem - Goal Stack Planning

SNo	Action functions	Precondition	Action Effect
1	Pickup(x)	Arm Empty On(x, Table) Clear(x)	Holding(x)
2	Putdown (x)	Holding(x)	Arm Empty On(x, Table) Clear(x)
3	Stack(x,y)	Holding(x) Clear(y)	On(x,y) Clear(x) Arm Empty
4	Unstack(x,y)	On(x,y) Clear(x) Arm Empty	Holding(x) Clear(y)

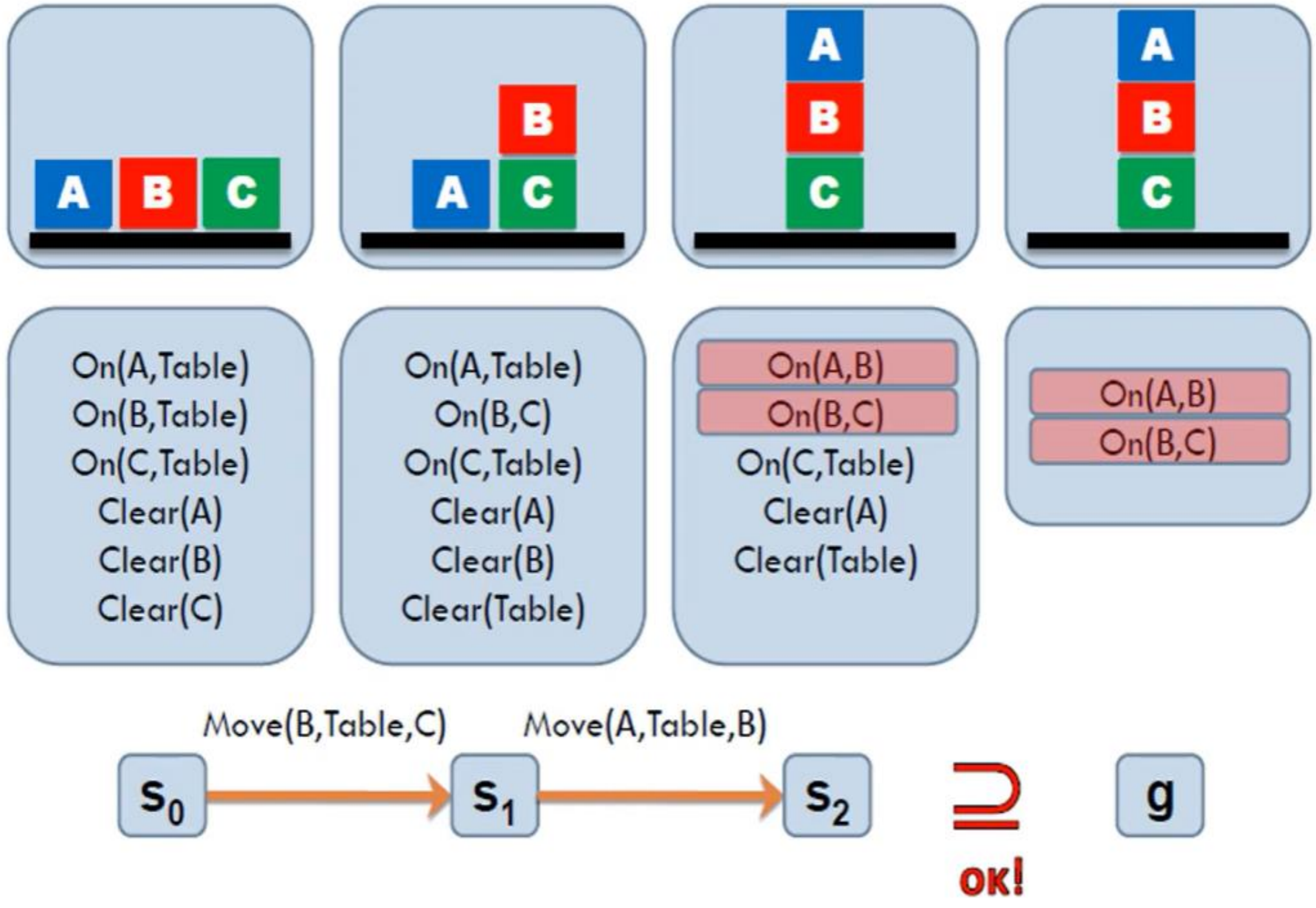


Block World Problem - Goal Stack Planning





STRIPS planning



Means - Ends Analysis

- **Means - Ends Analysis** - It is a mixture of Backward and forward search technique.
- Mixed strategy - solve the major parts of problem first and solve the smaller problems that arise when combining them together.
- The means - ends analysis process centers around finding the difference between current state and goal state.

Working of MEA technique for solving a problem.

1. First, evaluate the difference between Initial State and final State.
2. Select the various operators which can be applied for each difference.
3. Apply the operator at each difference, which reduces the difference between the current state and goal state.

Example of Mean-Ends Analysis

- Apply MEA to get the goal state.

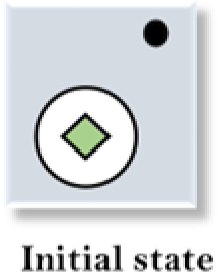
- Solution:

To solve the above problem, first find the differences between initial states and goal states, and for each difference, generate a new state and will apply the operators. The operators we have for this problem are:

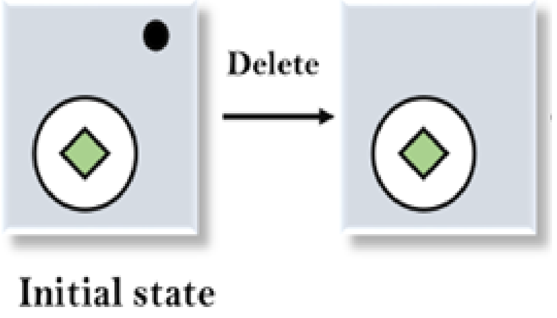
- **Move**
- **Delete**
- **Expand**

Example of Mean-Ends Analysis:

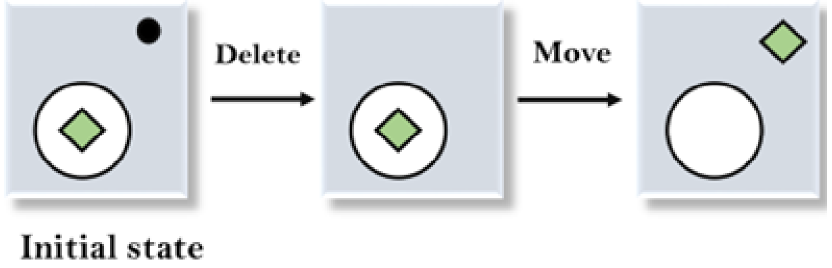
Step 1: Evaluate Initial State



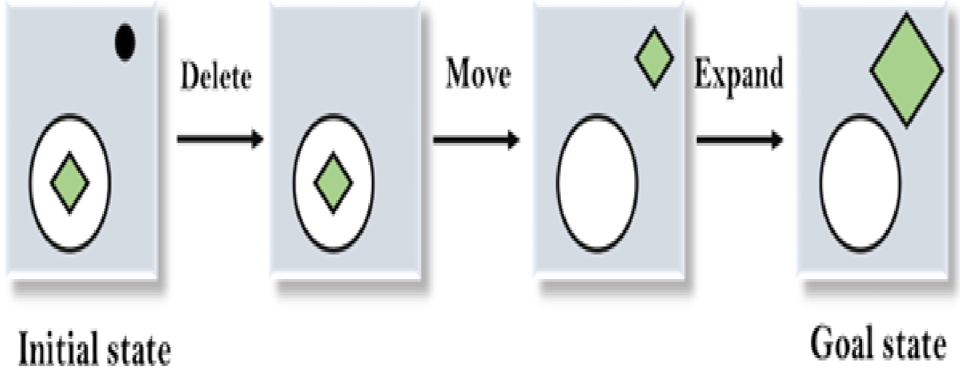
Step 2: Apply Delete Operator



Step 3: Apply Move Operator



Step 4: Apply Expand Operator



MEA Algorithm

- **Step 1:** Compare CURRENT to GOAL, if there are no differences between both then return Success and Exit.
- **Step 2:** Else, select the most significant difference and reduce it by doing the following steps until the success or failure occurs.
    a. Select a new operator O which is applicable for the current difference, and if there is no such operator, then signal failure.
    b. Attempt to apply operator O to CURRENT. Make a description of two states.
    i) O-Start, a state in which O?s preconditions are satisfied.
    ii) O-Result, the state that would result if O were applied In O-start.
    c. If
    (First-Part <------ MEA (CURRENT, O-START)
    And
    (LAST-Part <----- MEA (O-Result, GOAL), are successful, then signal Success and return the result of combining FIRST-PART, O, and LAST-PART.

Machine learning - Learning concepts, methods and models

10.1 WHAT IS MACHINE LEARNING?

Before going through the machine learning, let us understand the activities of learning, which are active in humans. There are three fundamental continuously active learning mechanisms—perceptual learning (learning of new objects, categories, relations), episodic learning (learning of events like what, where and when) and procedural learning (actions and their sequences to accomplish a task). Implementation of this human cognition can impart intelligence to a machine. So, a unified methodology around intelligent behaviour is the need of the hour.

Machine learning is building and exploring of methods for programming computer to make them Learn. Page 258

Machine learning - Learning concepts

Some model for prediction is built based on the available information and then further used to infer/ predict unknown data.

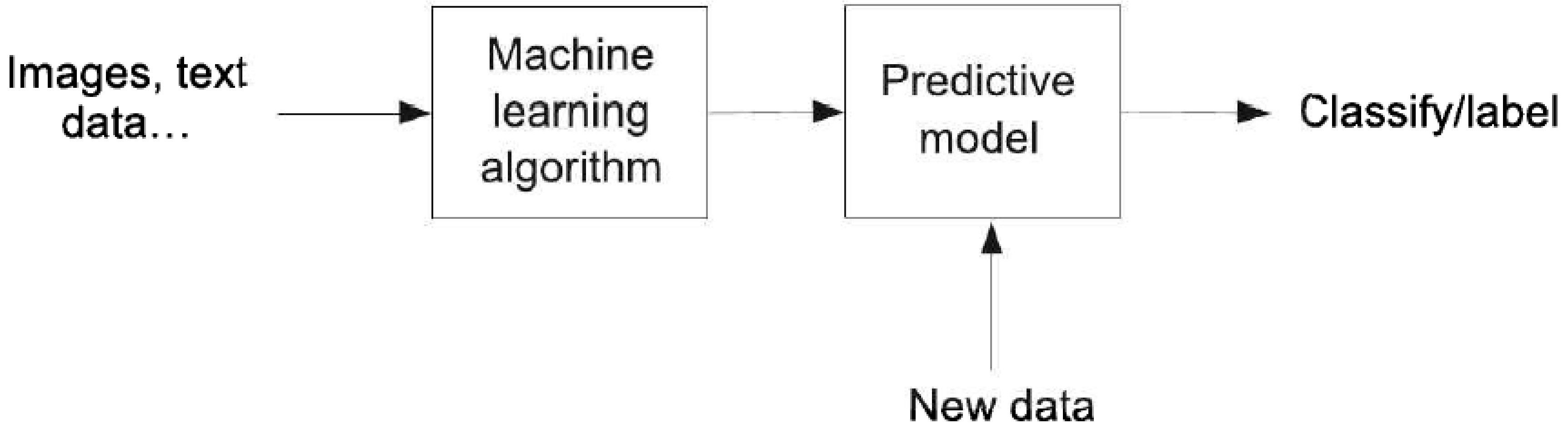


Figure 10.1 Machine learning approach.

GOALS OF MACHINE LEARNING

RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

- To produce learning algorithms with practical value.
- Development and enhancement of computer algorithms and models to meet the decision making requirements in practical scenarios.
- To facilitate in building intelligent systems (IS) that can be used in solving real time problems.

Challenges of Machine Learning

- Availability of limited learning data and unknown perspectives.
- Acquiring Accurate , compact and precise knowledge building.
- Require large working memory to store data.
- Focusing Too Much on Algorithms and Theories
- Monitoring and maintenance

Different Computational structure used in Machine learning are

Computational structure used in Machine learning:

1. Functions.
2. Logic programs and rule sets
3. Finite state machines
4. Grammars
5. Problem solving system

Traditional paradigm of Learning - types of learning Concepts

1. Rote learning
2. Learning from observations
3. Supervised learning
4. Unsupervised learning
5. Semi-supervised learning
6. Ensemble learning
7. Discovery based learning
8. Learning by problem solving

1. Rote learning

- Rote learning is basic form of learning, which focuses on memorization.
- Storing or memorizing the results improves the performance of a system
- saves a significant amount of time.
- **Example: selective paging catching**

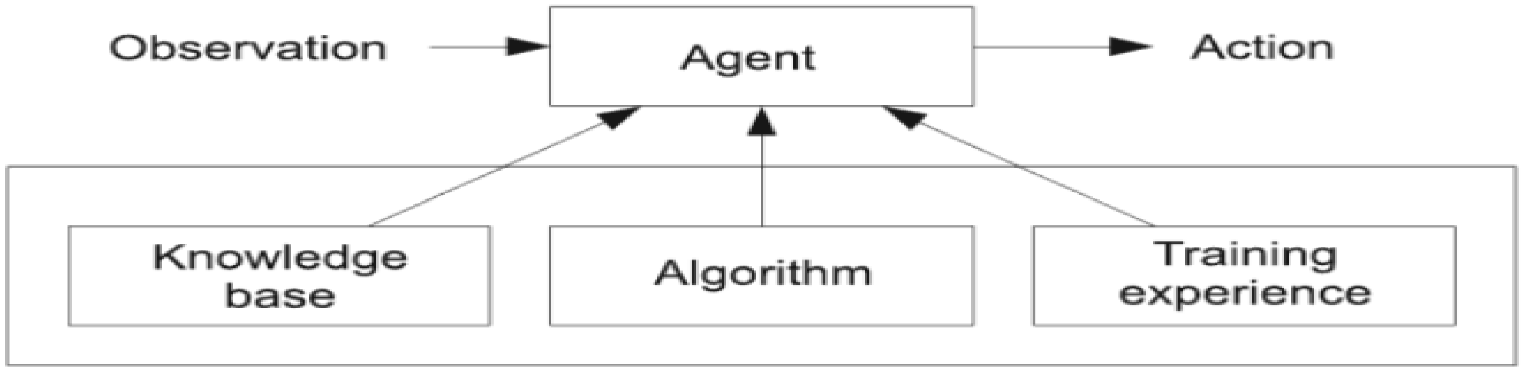
2. Learning from observations

There are 3 types

- Learning from agents
- Inductive learning
- Decision tree learning

Learning from agents

- An agent is defined the computational entity which is capable of perceiving the environment and can act based on the situation.
- The agent is composed of
  - learning element (which is responsible for making improvements)
  - Performance element (which is responsible for selecting external actions)
  - and a curiosity element. (is responsible for selecting a reward function)
- Based on the coordination between these elements the outcome of the agents behavior is measured.



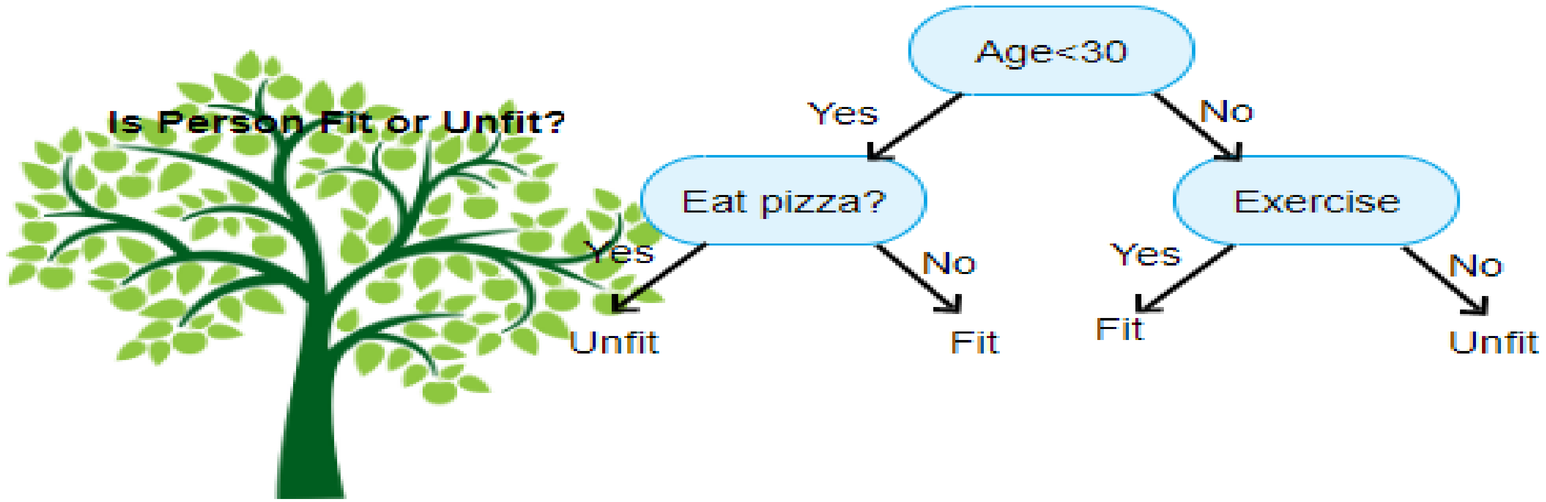
Inductive Learning

- Inductive learning involves **generating a set of classification rules .**

s.no	place type	weather	location	decision
1.	hilly	winter	kullu	Yes
2.	mountain	windy	Shimla	Yes
3.	beach	warm	goa	Yes
4.	beach	warm	Shimla	Yes

•**Rule 1:** IF the weather is warm THEN the decision is yes.
•**Rule 2:** IF the place type is hilly THEN the decision is yes.

Decision Tree Learning



•**<u>Widely used inductive learning inferencing method.</u>**

The learned function is represented by a decision tree.

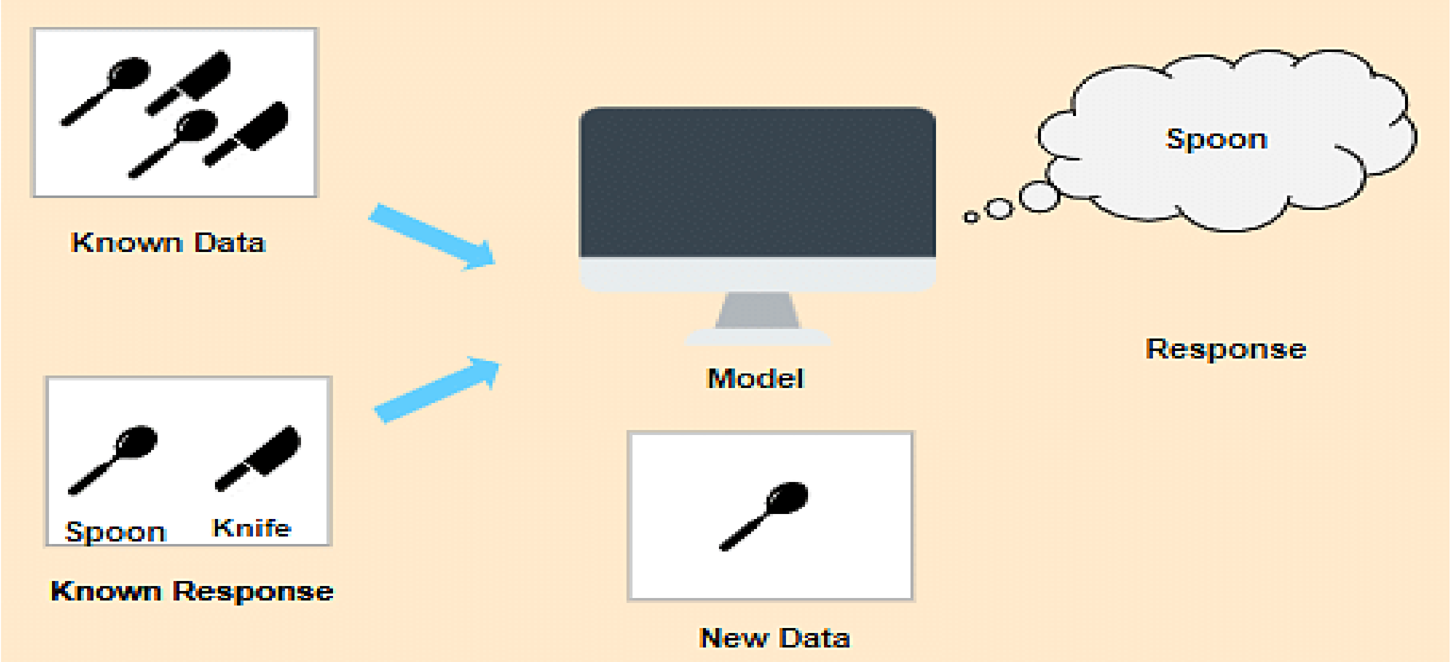
•In terms of programming it is also represented as if then rules.

•Decision tree depicts the simple learning from the observation, method.

•Based on the observation, at every node, decision is taken.

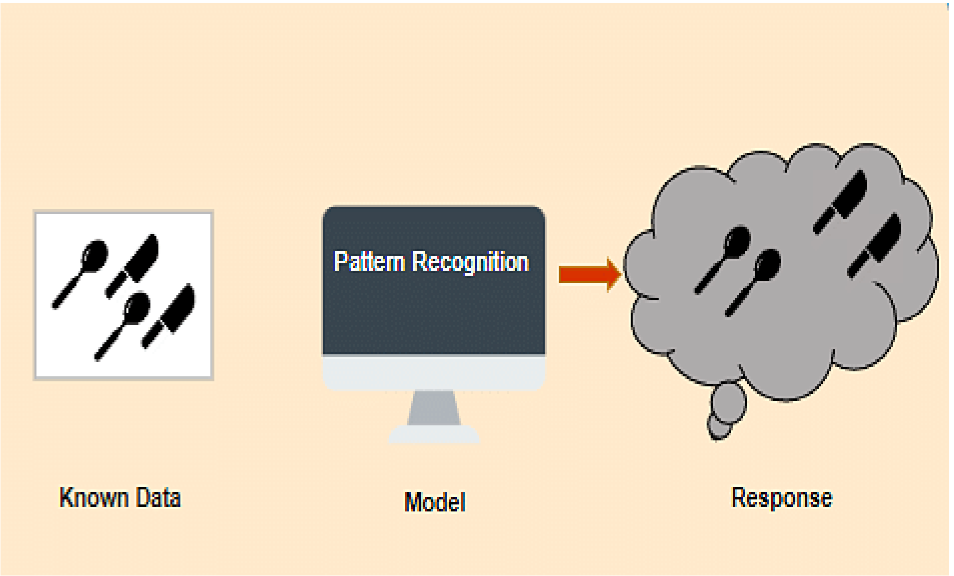
3. Supervised learning

- In Supervised Learning, the machine learns under supervision.
- It contains a model that is able to predict with the help of a **labeled dataset.**
- A labeled dataset is one where you already know the target answer.



4. Unsupervised Learning

- Use unlabeled dataset.
- In Unsupervised Learning, the machine uses unlabeled data and learns on itself without any supervision.
- The machine tries to find a pattern in the unlabeled data and gives a response.
- Grouping and categorization of the objects is based on the understanding of similarities and visualization of their relations
- Unsupervised learning performs hierarchical clustering.



5. Semi - Supervised Learning

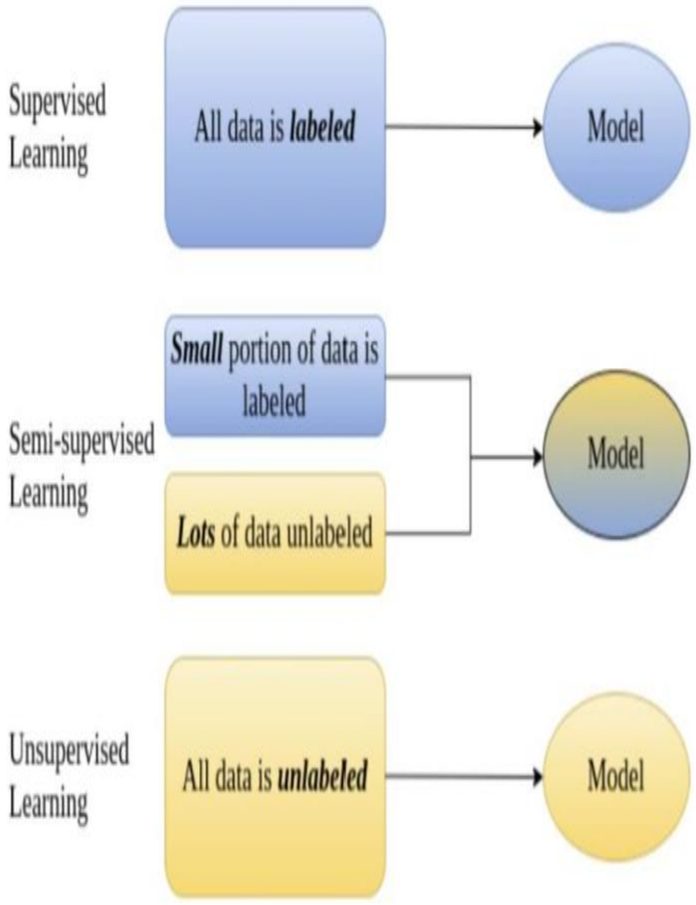
- Semi-supervised learning tries to learn from the labelled as well as unlabeled data.

- Let U be a set of unlabeled data and L be a set of labelled data.

- As the learning process the learning approach identifies the unlabeled data U with reference to a labelled data L and <u>keeps on labelling the unlabeled data.</u>

- This method is also called as self training in semi supervised learning.

- With semi-supervised learning, you train an initial model on a few labeled samples and then iteratively apply the model to a larger dataset to label them.



6. Ensemble Learning

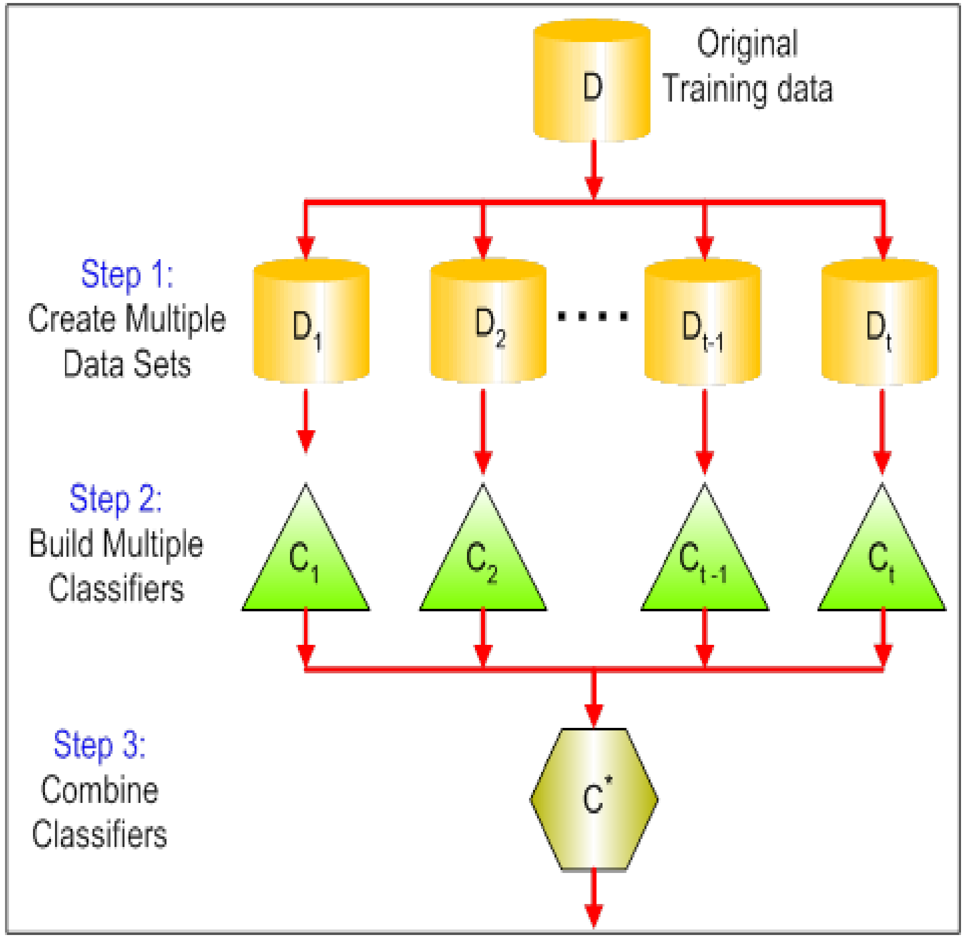
Ensemble learning is an approach in which two or more models are fitted to the same data, and the predictions of each model are combined.

In this approach, the learners or referred to as base learners.

- The most commonly used ensemble learning methods are

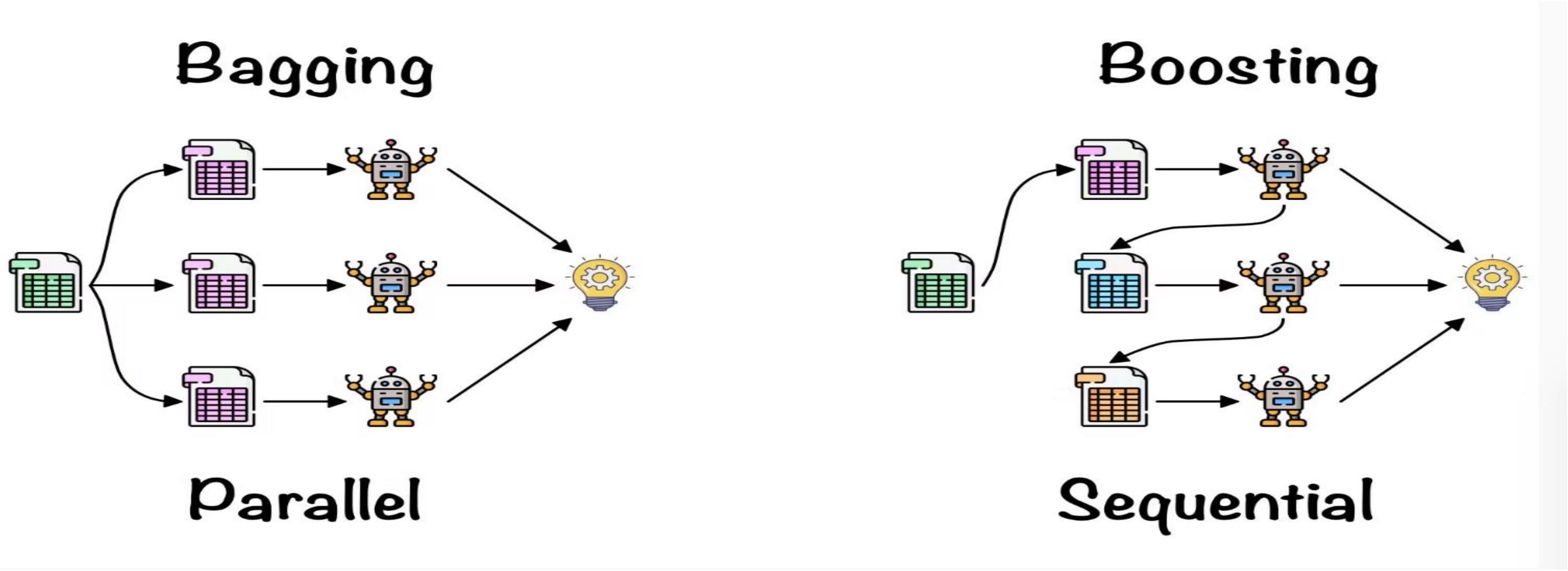
1) Boosting

2) Bagging



6. Ensemble Learning

1) Boosting - Boosting adjust the weight of an observation based on the last classification. If an observation was classified incorrectly, it tries to increase the weight of this observation and vice versa. Boosting in general decreases the bias error and builds strong predictive models
2) Bagging - Bootstrap aggregation, or bagging, is a technique that improves prediction accuracy by combining predictions from multiple models.



7. Discovery based learning

- The learning is based on the basis of past experience and knowledge
- Tries to discover the outcomes based on these knowledge

8. Learning by Problem Solving

- Various parameters related to solution and problem are considered.
- These parameters are used and effectively desirability of a particular outcome or decision is determined

Learning Methods

1. Bayesian Learning
2. Learning with hidden variables – EM Algorithm
3. Artificial neural network based learning-Back propagation
4. Support vector machines
5. Reinforcement learning
6. Adaptive learning
7. Multi_agent based learning
8. Ensemble learning
9. Learning for decision making
10. Distributed learning
11. Speedup learning
12. Generalized learning

1. Bayesian Learning

- Learning based on Bayes' theorem
- Bayes' theorem allows us to infer our belief in a hypothesis based on new data.
- We start with a prior belief in the hypothesis, represented by P(A), and then update this belief based on how likely the data are to be observed under the hypothesis, represented by P(B|A).
- The posterior probability P(A|B) represents our updated belief in the hypothesis after considering the data.

2. Learning with hidden variables – EM Algorithm

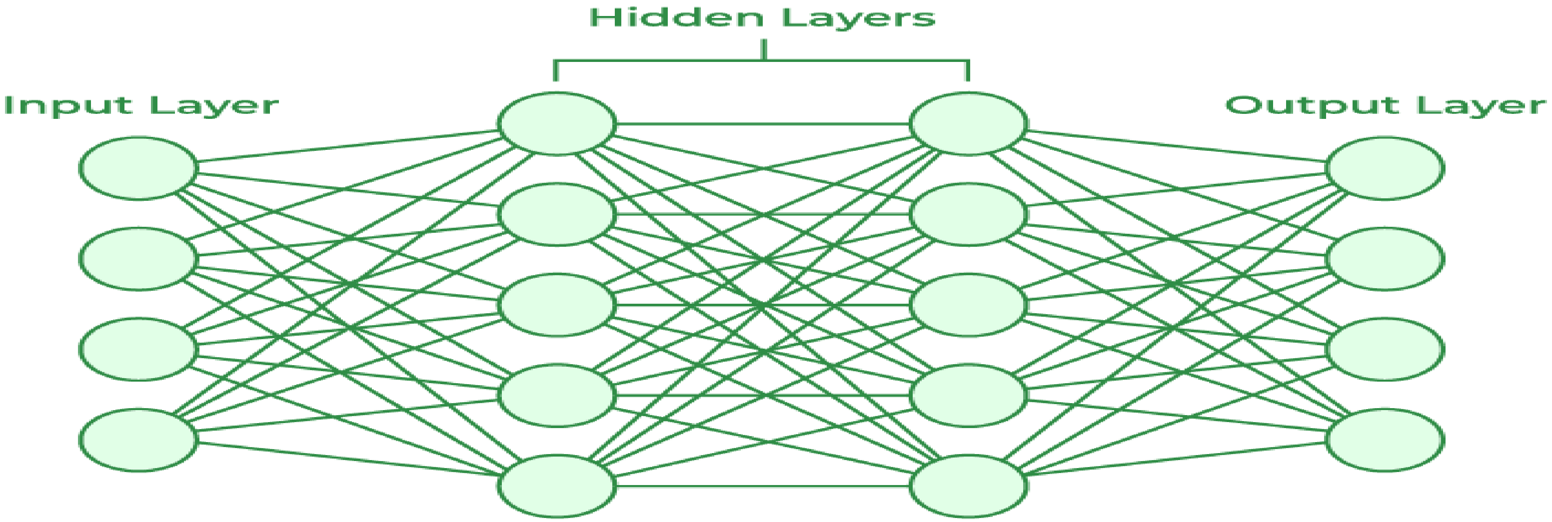
- The Expectation-Maximization (EM) algorithm is defined as the combination of various unsupervised machine learning algorithms, which is used to determine the local maximum likelihood estimates (MLE) or maximum a posteriori estimates (MAP) for unobservable variables in statistical models.
- Further, it is a technique to find maximum likelihood estimation when the latent variables (**hidden variable)** are present.
- It is also referred to as the latent variable model.
- A latent variable model consists of both observable and unobservable variables where observable can be predicted while unobserved are inferred from the observed variable. These unobservable variables are known as latent variables.

3. Artificial neural network based learning

- ANN is a computational model that performs simulation of the human biological neurons.
- Artificial Neural Networks contain artificial neurons which are called units. These units are arranged in a series of layers that together.
- Artificial Neural Network has an input layer, an output layer as well as hidden layers.
- The input layer receives data from the outside world which the neural network needs to analyze or learn about.
- Then this data passes through one or multiple hidden layers that transform the input into data that is valuable for the output layer.
- Finally, the output layer provides an output in the form of a response.

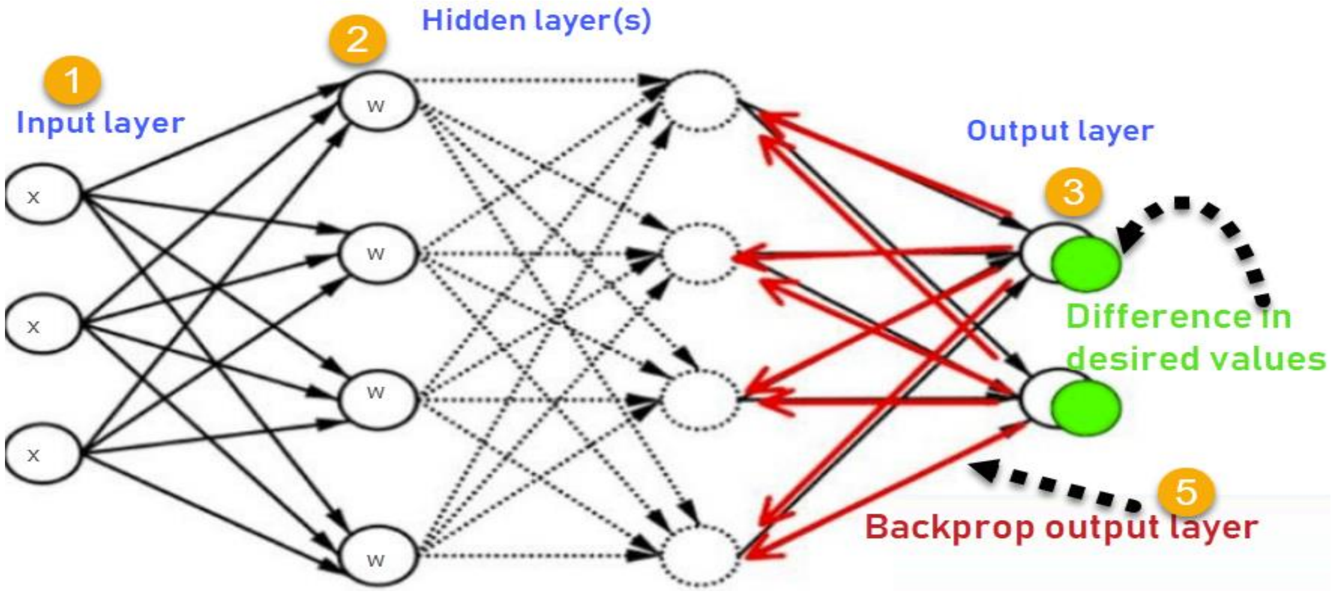
3. Artificial neural network based learning

- Units are interconnected from one layer to another.
- Each of these connections has weights that determine the influence of one unit on another unit.
- As the data transfers from one unit to another, the neural network learns more and more about the data which eventually results in an output from the output layer.



Example: Back- propagation Algorithm

- Backpropagation is an effective algorithm used to train artificial neural networks, especially in feed-forward neural networks.
- Back-propagation is the essence of neural net training.
- It is the method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch (i.e., iteration).
- Proper tuning of the weights allows you to reduce error rates and to make the model reliable by increasing its generalization.



Backpropagation Algorithm

Forward pass:

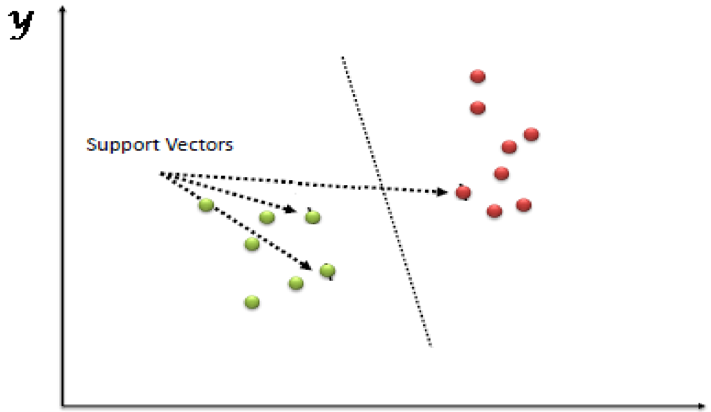
computes ‘functional signal’, feed forward propagation of input pattern signals through network

Backward pass phase:

computes ‘error signal’, propagates the error backwards through network starting at output units (where the error is the difference between actual and desired output values)

4. Support Vector Machines (SVM)

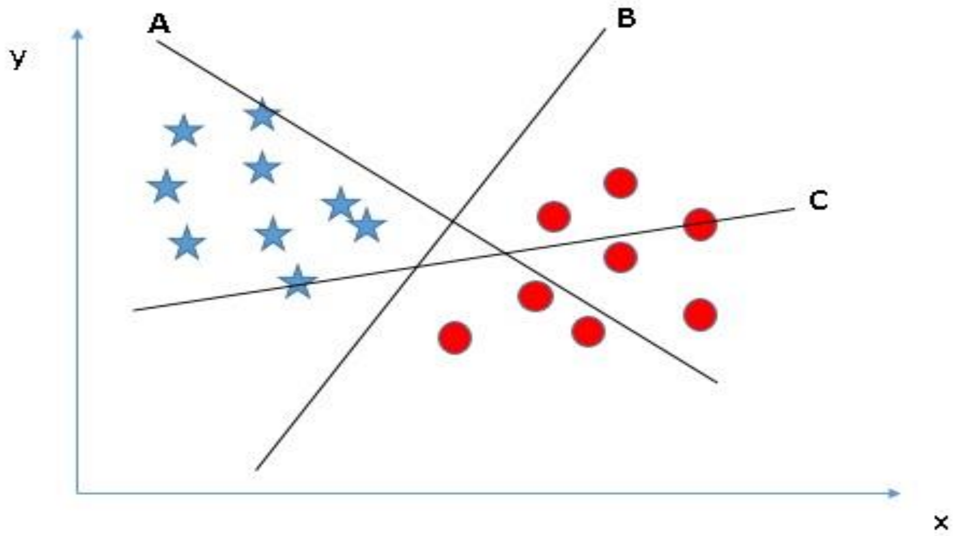
- "Support Vector Machine" (SVM) is a supervised learning machine learning algorithm that can be used for both classification or regression challenges.
- However, it is mostly used in classification problems, such as text classification.
- In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a particular coordinate.
- Then, we perform classification by finding the optimal hyper-plane that differentiates the two classes very well



4. Support Vector Machines (SVM)

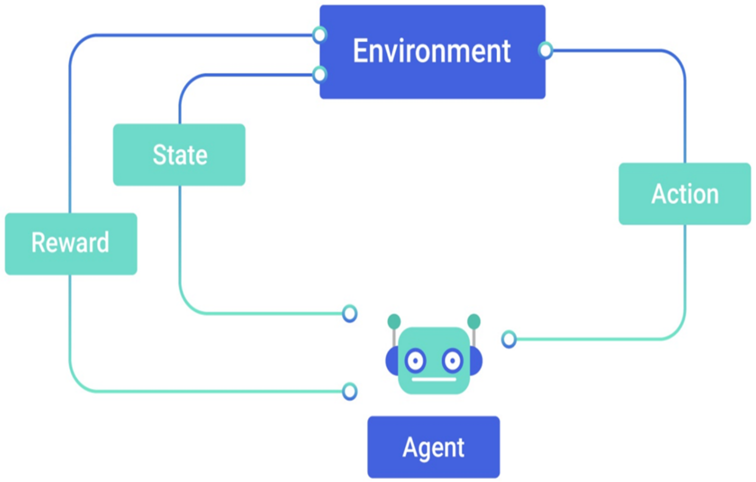
What is a hyperplane?

- A hyperplane is a line that linearly separates and classifies a set of data.
- So when new testing data are added, whatever side of the hyperplane it lands will decide the class that we assign to it.
- remember a thumb rule to identify the right hyper-plane: "Select the hyper-plane which segregates the two classes better." In this scenario, hyper-plane "B" has excellently performed this job.



5. Reinforcement Learning

- Learning from interaction with an environment to achieve some long-term goal that is related to the state of the environment
- The goal is defined by reward signal, which must be maximized.
- Agent must be able to partially/fully sense the environment state and take actions to influence the environment state
- The state is typically described with a feature-vector



6. Adaptive learning

- No learning method is complete in itself. So
- Need to select the learning method based on the requirements.
- Need to develop a combination of some of the existing methods based on requirements.
- **Adaptive machine learning algorithms are the machine learning models, where the changes in the environment help in selecting the algorithm or learning method.**

Adaptive learning (Contd..)

- As per the scenario, most suitable algorithm is selected.
- Moreover the development of especially fast adapting algorithms poses many different issues like selection of choices, handling equilibrium states and so on.
- The adaptive learning solves some of the complex problems for which a single learning method is not enough.
- This method is even more appropriate when the environment is continuously changing and real time response is expected.

7. What is a Multi-Agent System?

- A multiagent system is one that consists of a number of agents, which *interact* with one-another
- In the most general case, agents will be acting on behalf of users with different goals and motivations
- To successfully interact, they will require the ability to *cooperate*, *coordinate*, and *negotiate* with each other, much as people do

Need for Multi agent Learning

- A single agent cannot handle learning in case of applications
- A team or group of agents possesses the potential to overcome the limitation of single agent and work in in coordination to accomplish a task.
- This can be two cases in multi agent based learning:

Multi agent based learning

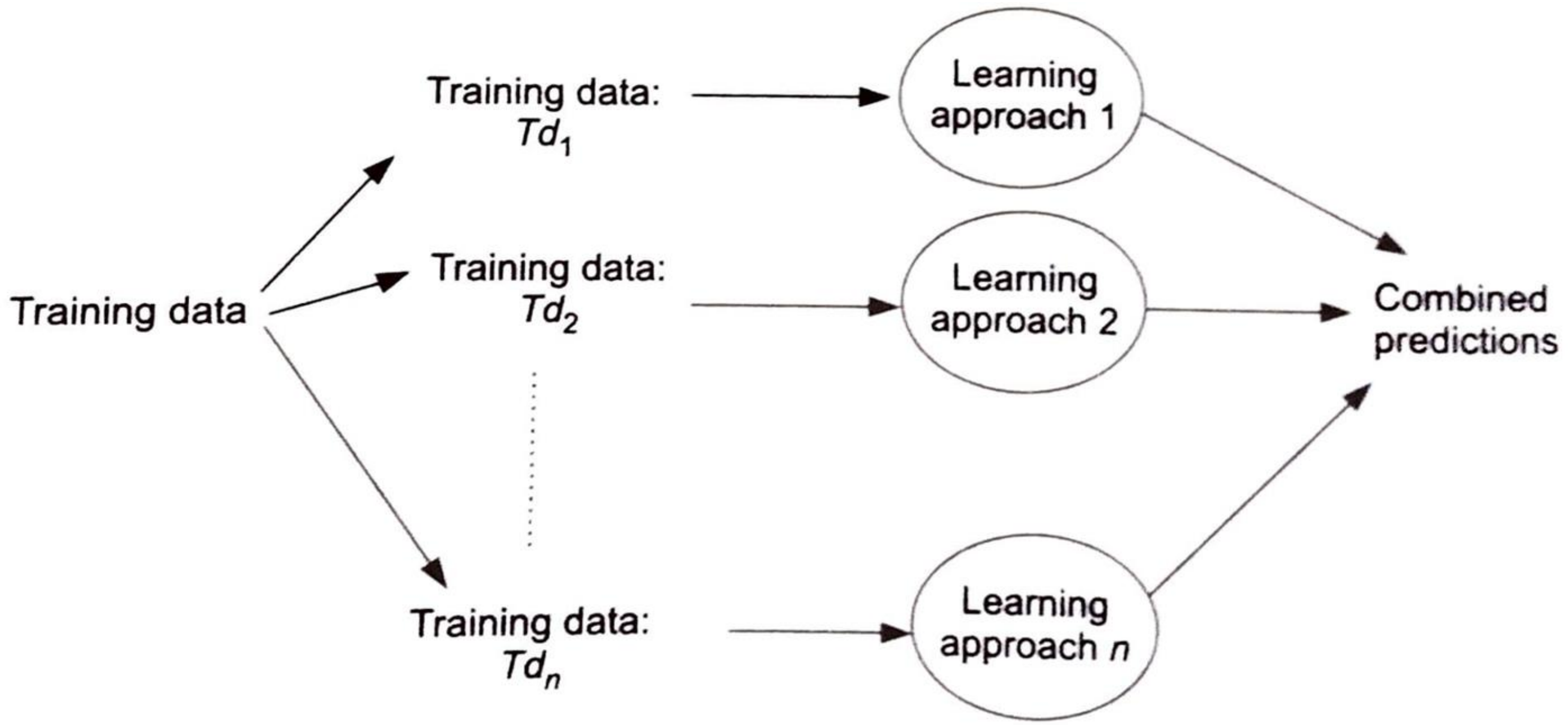
1) Where the agent tries to maximize its own utility

- Eg: Consider a manufacturing industry domain. The task are built and assigned where each agent works in co-operation to build the end product. This is the case to achieve a common goal.

2) Where they worked in collaboration to achieve some common goals.

- Eg: Game playing. In that gaming environment, multiple agents are in Operation to select the best strategy.
- Can be related with the reinforcement learning, Where for each strategy of the agent some reward is achieved this is where each agent tries to maximize his own utility function.

8. Ensemble Learning



Ensemble method: The concept.

8. Ensemble Learning (Contd…)

Ensemble learning method is the one where multiple learners or learning algorithms are trained.

- In most of the learning algorithms a single hypothesis drives the learning.
- In ensemble learning method the whole collection or ensemble of hypothesis is selected from the hypothesis space and their predictions are combined.
- In this approach, the learners or referred to as base learners.
- The most commonly used ensemble learning methods are

1) Boosting
2) Bagging.

Ensemble Learning (Contd..)

Boosting:

- Boosting can probably be defined as the method for generating accurate predictions by combining the rules that are comparatively inaccurate.
- Boosting works on the weighted training sets. The weights of the training example reflects the importance of training examples.

Bagging:

- In Bagging, the training data is resampled. This is referred to as ***bootstrap sampling***, where the training data with replacement is taken in the learning approaches.

9. Learning for decision making

- It is observed from different learning mechanisms that Capability to take decisions is increased.
- speaking about the supervisor and unsupervised methodologies the decisions taken are not sequential in nature.
- That is, if the system make a mistake on one decision, this has no bearing on the subsequent decisions.
- To cope up with this dynamic situation there is a need to understand the perspective of decision making.
- Another aspect is environment and system Learning, which also needs to be looked upon during decision making. While taking decisions **one specific learning approach may not be suitable**.
- The learning approach is dependent on decision scenario.

10. Distributed learning

- In distributed learning the task of learning is distributed.
- **Need for distributed learning** - Arises due to large data sets and time constraints.
- More than one agent in different parts of the data set. There will be **distributed learning algorithms** taken part in each partition to get the desired outcome, which would then be combined.
- **Efficiency** of distributed learning is affected to look at. it is extremely important that outcome of distributed learning matches with the ones acheived under the absence of distributed environment.
- Multi agent systems can be thought of as a **subset of distributed learning.**

11. Speedup learning

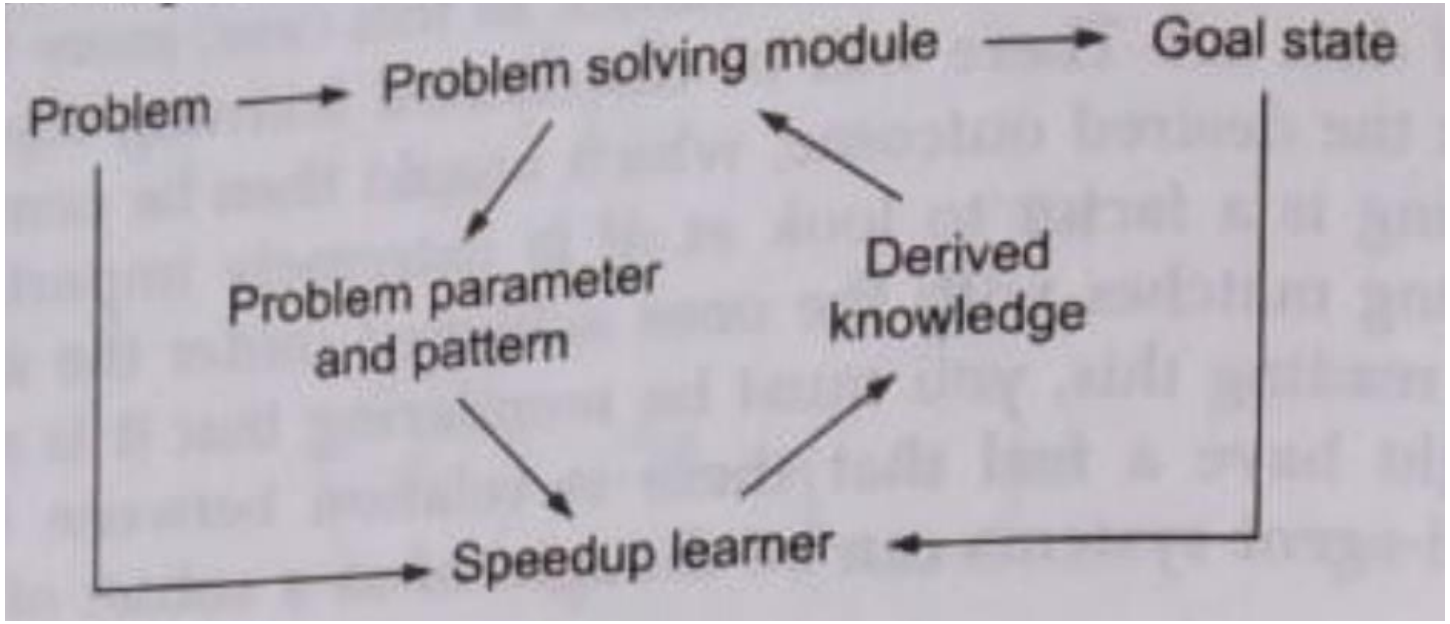
- Speed learning typically deals with speeding up problem solving by effective use of problem solving experience. Hence, Prayer problem solving experience is an input for speed of learning.
- In this learning,

1) There is no option with the environment.

2) New problems cannot be solved.

So, speed up learning accelerates the process experiences
And prior observations.

11. Speedup learning (Contd..)



Speedup learning modules.

12. Generalized learning

- Another dimension to the speed up learning is generalized learning. this is also known as explanation based learning.
- There are a number of issues with explanation based learning, as it is implemented and embedded in system to solve real life problems and is suitable for a particular set of problems where the sequential processes once developed can be used again and again.
- But this is not the case in many real life problems, where dynamic change in environment demands the improvement in the established scenarios and even there is a need to keep on learning based on the new findings
-

➢ INTRODUCTION TO EXPERT SYSTEM:

1. Is a Computer based system.

2. Uses Artificial Intelligence techniques.

3. Is built to **replicate** the knowledge, reasoning and decision-making processes of **a human expert**.

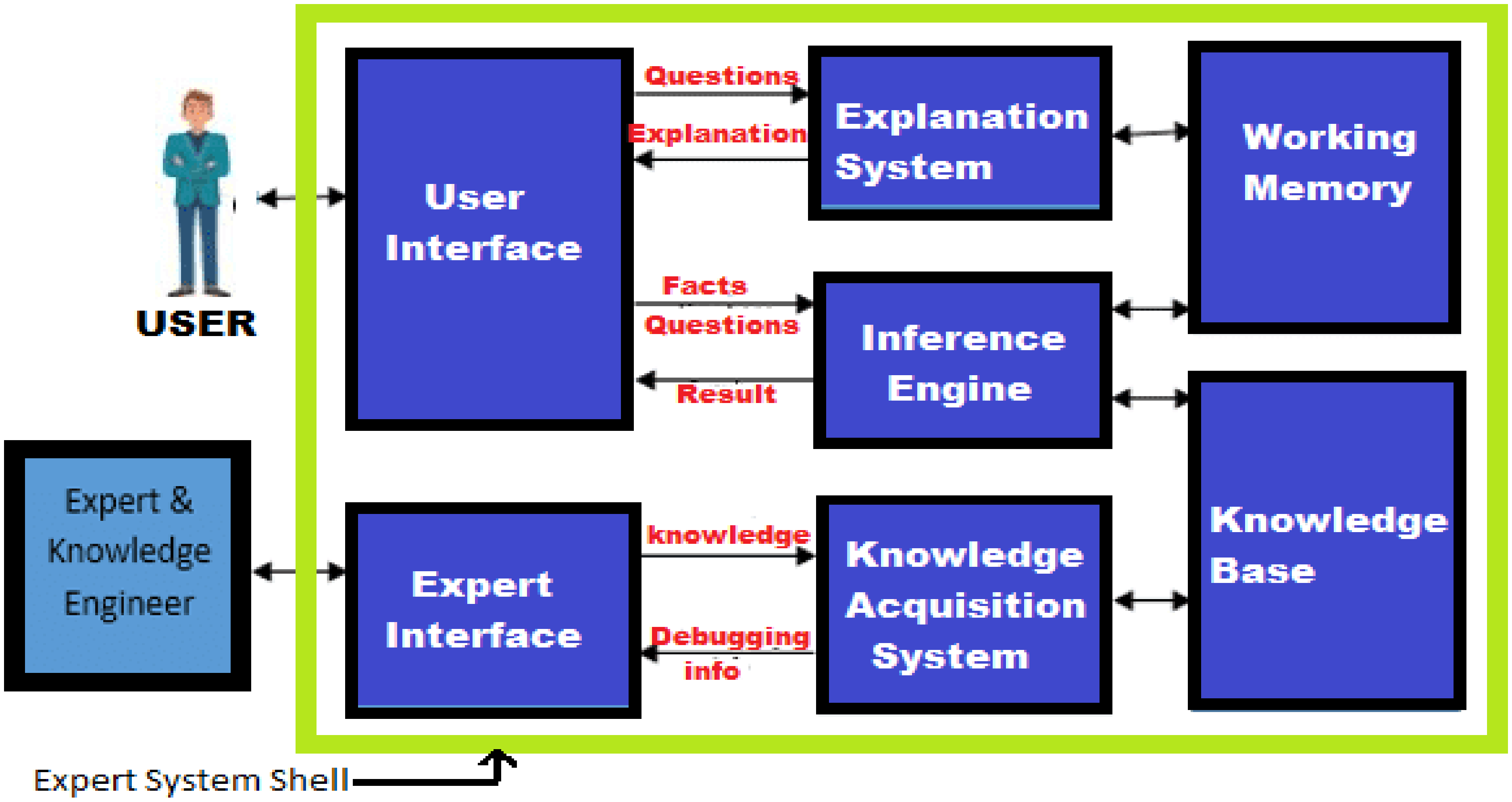
4. To provide solutions, make recommendations or assist in solving complex problems within that domain.

EXPERT SYSTEM

- **COMPONENTS IN AN EXPERT SYSTEM**

  1. User
  2. User Interface
  3. Knowledge Base
  4. Inference Engine
  5. Working Memory
  6. Knowledge Acquisition System
  7. Knowledge Engineer
  8. Expert Interface
  9. Human Expert
  10. Explanation System

➢ ARCHITECTURE OF AN EXPERT SYSTEM



1. User:

- A computer user is an individual who interacts with a computer system or device to perform tasks.

2. User Interface:

- A system that allows a non-expert user to query the expert system.
- Is a point of interaction between a human user and an artificial intelligence system.
- Enabling users to input queries, receive responses, interact with features, and perform tasks within the AI application or system.
- Ex: Graphical User Interface (GUI)

  Voice User Interface (VUI)

  Chatbots or Conversational UI

  Gesture-Based Interface

3. Knowledge Base:

- Is a **repository or database** that stores structured information, facts and rules about a specific domain or problem.
- Serves as the foundation for decision-making, problem-solving, or reasoning within the AI system.
- It is divided into two main parts:

  a. **Facts base -** which contains specific information about the domain.

  b. **Rules base** - which holds the rules or logic used for reasoning and decision-making.

Facts Base:

Scenario: The system has a facts base containing specific data about patient symptoms.

Example Facts:

Symptom 1: "Patient A has a fever."

Symptom 2: "Patient A experiences fatigue."

Symptom 3: "Patient A has a cough."

These facts represent the symptoms reported by a patient.

Rules Base:

- Refers to a collection of rules or logical statements used by a system.
- Each rule typically follows an "IF-THEN" structure.
- Rules are stored in a knowledge base and are used by the inference engine to make decisions or draw conclusions.
- In general:

IF (condition A AND condition B are true), THEN perform action X.

IF condition C is true, THEN perform action Y.

Example Rules:

Rule 1: "If a patient has a fever and fatigue, then suspect a viral infection."

Rule 2: "If a patient has a cough and fatigue, then suspect a respiratory illness."

4. Inference Engine:

- Acts like a **search engine applies inference rules** for information that matches the user's query.
- It evaluates, interprets and derives new information from existing knowledge to make decisions or provide solutions to problems using Rule base.
- Categorized based on their approach to reasoning and the methods:
    a. Forward Chaining
    b. Backward Chaining

Forward Chaining: start state to goal state.
Approach: Starts with known symptoms and iteratively applies rules to deduce potential diagnoses.

Knowledge Base:
Rule 1: IF (fever AND cough AND fatigue) THEN diagnose as flu.
Rule 2: IF (fever AND rash) THEN diagnose as allergic reaction.
Rule 3: IF (cough AND sore throat) THEN diagnose as common cold.

Process:
User Input: "fever, cough, fatigue"
The forward chaining engine starts with the symptoms provided.
It matches the symptoms against the rules in the knowledge base.
Rule 1 matches the input symptoms ("fever, cough, fatigue").

Conclusion: Forward chaining deduces the potential diagnosis of "flu" based on Rule 1.

Backward Chaining: goal to start state
Approach: Starts with a goal or diagnosis and works backward to find symptoms or conditions supporting that diagnosis.

Knowledge Base:
Rule 1: IF (fever AND cough AND fatigue) THEN diagnose as flu.
Rule 2: IF (fever AND rash) THEN diagnose as allergic reaction.
Rule 3: IF (cough AND sore throat) THEN diagnose as common cold.

Process:
User Query: "What could cause flu?"
The backward chaining engine starts with the goal ("flu") and checks which rules lead to that diagnosis.
It identifies Rule 1 as the one leading to the diagnosis of "flu."
Working backward, it identifies the symptoms in Rule 1: "fever, cough, fatigue."

Conclusion: Backward chaining determines that the symptoms associated with the diagnosis of "flu" are "fever, cough, and fatigue."

5. Working memory:

- Is a short-term memory component that stores data, facts, or information **temporarily while the system is processing and reasoning.**

- This information can include recent input data, intermediate results of computations, or relevant facts retrieved from the knowledge base.

- The contents of the working memory can change rapidly as the AI system performs various tasks.

- It's a dynamic storage area that gets updated or replaced as new information is processed.

6. Knowledge Acquisition System :

- A Knowledge Acquisition System (KAS) in AI refers to a **set of processes, tools, methodologies and techniques**.
- It is used to gather, extract and formalize knowledge from human experts or external sources to be utilized within an AI system.

7. Knowledge engineer:

- Responsible for designing and building knowledge-based systems.
- Works on creating algorithms and structures that allow computers to utilize and process information in a way that mimics human knowledge.
- Develops the rules and logic systems necessary for AI to reason and make decisions based on the available knowledge.

8. Expert Interface:

- **Is a point of interaction between a system and an expert person**.
- Enabling experts to receive queries, provide responses to expert system and perform other tasks.

9. Human expert:

- Plays a vital role in **transferring their knowledge** into the AI system.
- Signifies the individuals whose expertise and knowledge are leveraged to train, guide or inform AI systems.
- Ensuring that these systems can replicate or simulate their decision-making processes or problem-solving abilities within a specific domain.
- Might collaborate with AI engineers, data scientists or knowledge engineers to define rules, create algorithms, annotate data or design the knowledge base necessary for the AI to function effectively within their area of expertise.

10. Explanation System:

- An Explanation System refers to a component or capability within an AI system.
- It provides **explanations or reasoning behind the system's outputs, decisions, predictions, or recommendations in a human-understandable manner**.
- It aims to bridge the gap between AI decision-making processes and human understanding.

- **HOW EXPERT SYSTEM WOULD BE CREATED?**

For Medical Diagnosis:

- Doctors are interviewed for requirements.

- Data is collected from experts.

- Knowledge base is designed and created.

- Rule base is designed and created.

- Inference engine is designed and created.

- Input screen is designed and created.

- Output format is designed and created.

- The system is tested against known diagnoses and results are evaluated.

- Interviewing doctors about effectiveness of the new system and how does it match original system requirements.

- **EXAMPLES OF EXPERT SYSTEMS:**

1. Medical Diagnosis:

- To assist in diagnosing diseases.

- Through interactive user screen, questions are asked with symptoms.

- Expert system compares symptoms with those in knowledge base using inference engine and applying rule base.

- Matches are found and advice are output.

- Recommends treatment based on the symptoms, history and knowledge from medical experts.

2. Troubleshooting and Maintenance:

- Provides guidance in troubleshooting for technical issues.
- Ex: Car engine faults diagnosis.
- Interactive user screen appears, details of car type are entered.
- On-board computer connected to expert system.
- Engine problems are typed in.
- Inference engine searches the knowledge base using the Rule base.
- Suggested probabilities of faults are output in the form of a report to the mechanic and report is displayed.
- Performs maintenance task for complex machinery/system.
- Helps to reduce downtime and improves efficiency.

3. Customer Support:

- To provide automated responses.

- Troubleshoots common issues.

- Guides user through problem-solving steps.

4. Financial Decision making:

- Helps for financial planning, risk assessment and investment advice.

- Based on market trends, historical data and expert knowledge.

➢ **LEADING EXPERT SYSTEM COMPANIES:**

1. IBM Watson – Includes healthcare, finance and customer service.

2. Oracle Corporation – Various industries includes manufacturing, finance and supply chain management.

3. Wolters Kluwer – Includes healthcare and legal sectors.

Advantages of Expert Systems

1. A consistent output
2. Quick and fast response
3. Location/date/day/time independent
4. Can be made generalised. (A change in application would result in looking out for a human expert related to that field if we are not relying on expert systems. But with expert systems, having generalisation, the process gets less complicated.)
5. Efficient utilisation of the knowledge (Human experts may forget some aspects, whereas expert systems consider all the rules and scenarios.)
6. Simple future enhancements with additional information of the knowledge and the rules (but with humans, it is difficult)
7. Easy maintenance of system.

Disadvantages of Expert Systems

The disadvantages of expert systems are mentioned below:

1. They cannot handle new dynamic situation.
2. The systems cannot be adaptive based on the decisions taken earlier.
3. Limited set of knowledge will leave them with limited set of decision outcomes.
4. Development cost could be high depending on the purpose they are used for.

MYCIN

MYCIN - Expert system

- MYCIN was an early expert system that used artificial intelligence to identify bacteria causing severe infections.
- recommend antibiotics, with the dosage adjusted for patient's body weight
- The MYCIN system was also used for the diagnosis of blood clotting diseases.
- MYCIN was developed over five or six years in the early 1970s at Stanford University.
- It was written in Lisp

MYCIN

•MYCIN was a stand alone system that required a user to enter all relevant information about a patient by typing in responses to questions MYCIN posed.

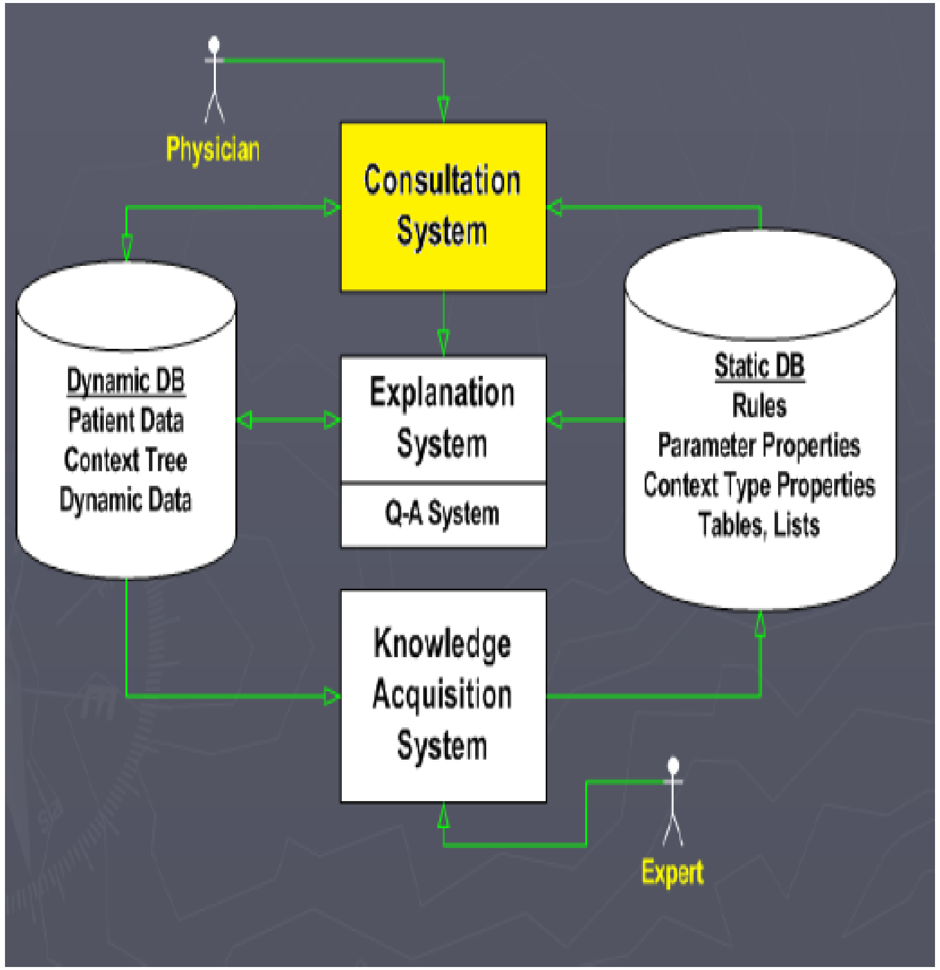
•MYCIN operated using a fairly simple inference engine, and a knowledge base of ~600 rules.

•It would query the physician running the program via along series of simple yes/no or textual questions.

Tasks and Domain

- Disease DIAGNOSIS and Therapy SELECTION
- Advice for non-expert physicians with time considerations and incomplete evidence on:
  - Bacterial infections of the blood
  - Expanded to meningitis and other ailments
  - Meet time constraints of the medical field

Consultation System

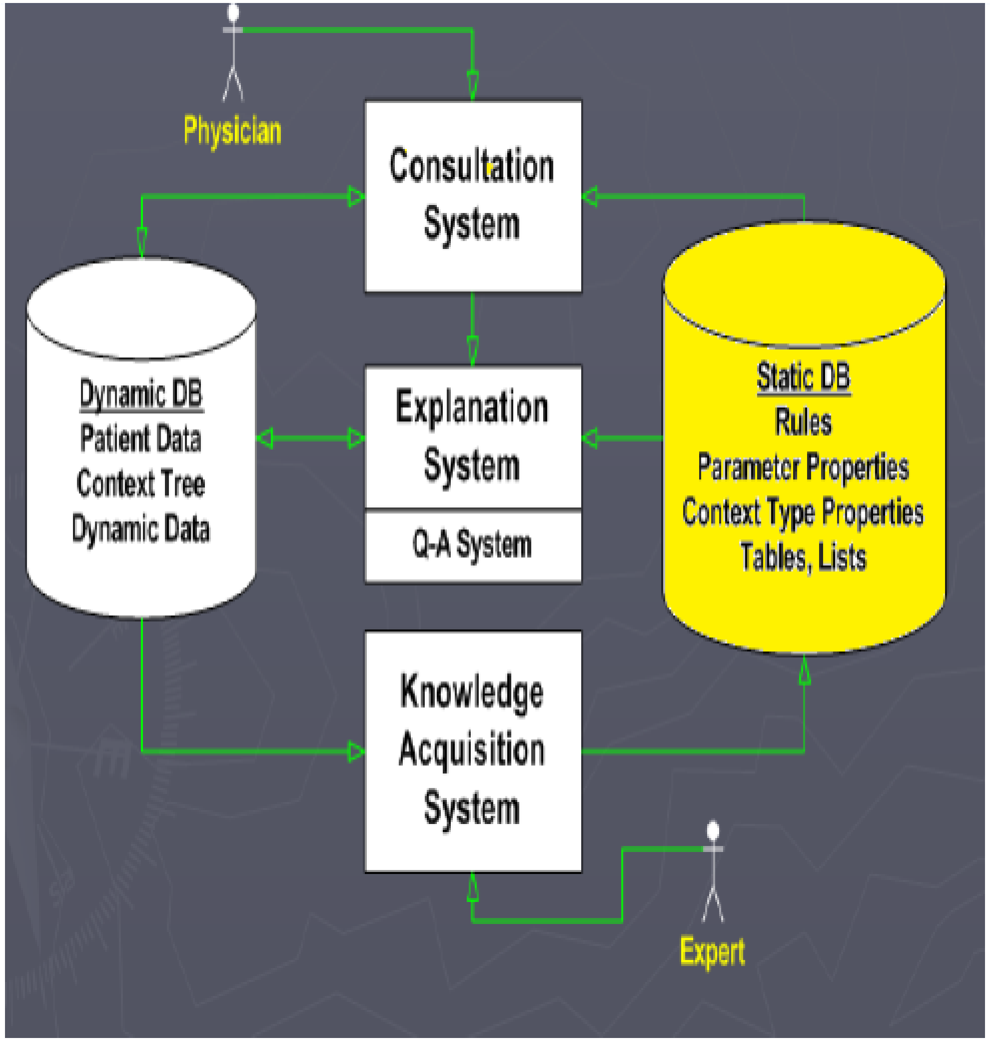


- Performs Diagnosis and Therapy Selection
- Control Structure reads Static DB (rules) and read/writes to Dynamic DB (patient, context)
- Linked to Explanations
- Terminal interface to Physician

Consultation "Control Structure"

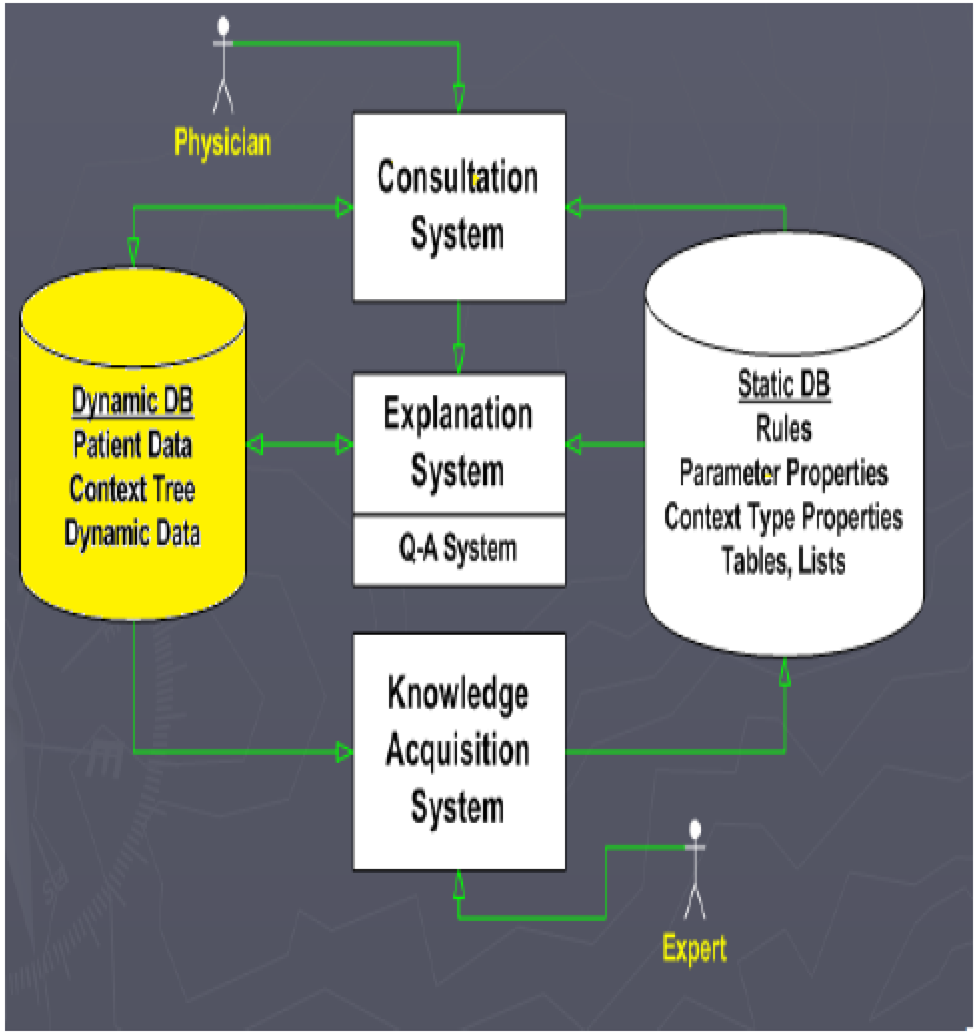
- Goal-directed Backward-chaining Depth-first Tree Search
- High-level Algorithm:
    1. Determine if Patient has significant infection
    2. Determine likely identity of significant organisms
    3. Decide which drugs are potentially useful
    4. Select best drug or coverage of drugs

Static Database



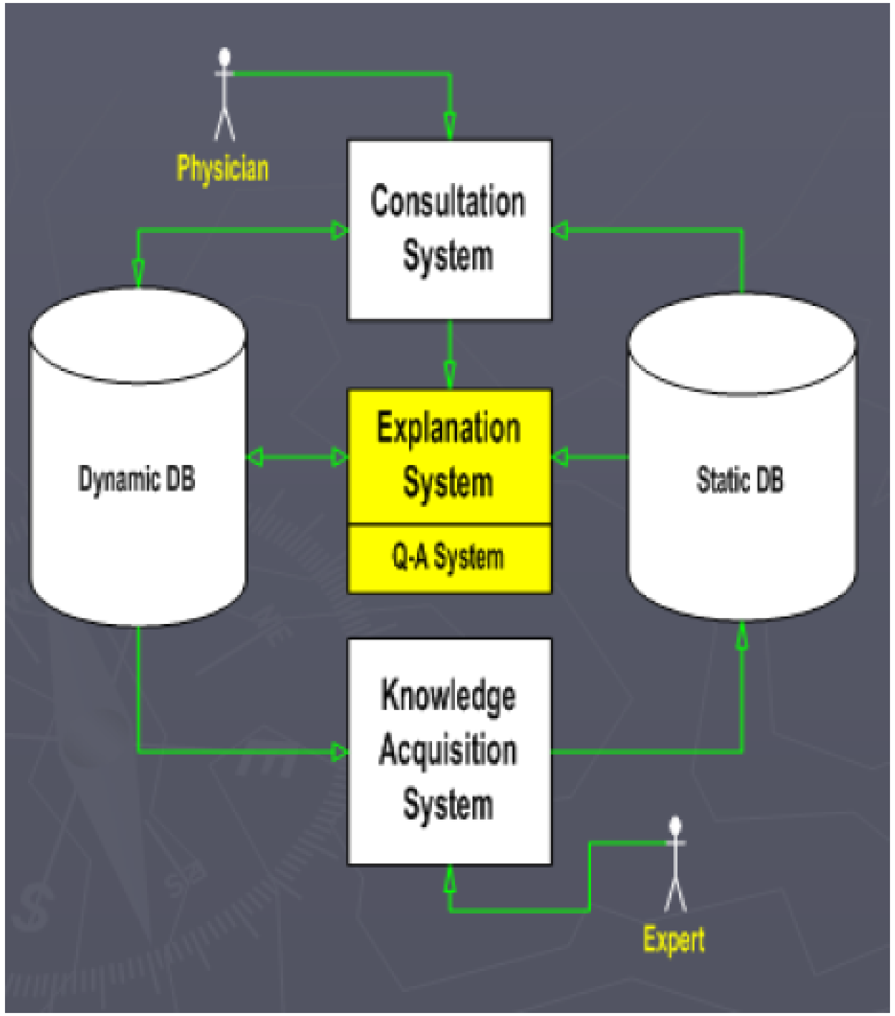
- Rules
- Meta-Rules
- Templates
- Rule Properties
- Context Properties
- Fed from Knowledge Acquisition System

Dynamic Database



- Patient Data
- Laboratory Data
- Context Tree
- Built by Consultation System
- Used by Explanation System

Explanation System



- Provides reasoning why a conclusion has been made, or why a question is being asked
- Q-A Module
- Reasoning Status Checker