AI/ML

Books:

- ‘Artificial Intelligence: A Modern Approach-2nd Edition’ by Stuart Russel, Peter Norvig, Pearson Education
- 'Introduction to Artificial Intelligence & Expert Systems' by Dan W. Patterson, Englewood Cliffs, NJ, 1990, Prentice Hall International
- 'Artificial Intelligence’ by Elaine Rich, Kevin Knight, Shivashankar B Nair, McGraw-Hill
- 'Artificial Intelligence’, 3rd Edition by Patrick Henry Winston, paperback, 1992

Other readings and relevant websites:

- http://www.cs.utexas.edu/~novak/cs381kcontents.html
- https://www.udacity.com/course/intro-to-artificial-intelligence--cs271
- http://nptel.ac.in/courses/106105077
- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010

Topics:

- Introduction – Definition, Future of Artificial Intelligence
- Characteristics of Intelligent Agents, Typical Intelligent Agents, Problem Solving Approach to Typical AI problems
- Problem-solving Methods, Search Strategies, A* algorithm, Uninformed, Informed, Heuristic
- search Techniques: Hill Climbing, Iterative deepening DFS, bidirectional search, Local Search Algorithms, and Optimization Problems
- Searching with Partial Observations, Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search.
- Game Playing, Optimal Decisions in Games, Alpha-Beta Pruning, Stochastic Games,
- First Order Predicate Logic, Unification, Forward Chaining, Backward Chaining

- The resolution, Knowledge Representation, Ontological Engineering, Categories and Objects, Events, Mental Events, and Mental Objects
- Reasoning Systems for Categories, Reasoning with Default Information
- Probability basics, Bayes Rule and its Applications, Bayesian Networks, Exact and Approximate Inference in Bayesian Networks, Hidden Markov Models
- Forms of Learning, Supervised Learning, Learning Decision Trees, Regression and Classification with Linear Models, Artificial Neural Networks
- Nonparametric Models, Support Vector Machines, Statistical Learning, Learning with Complete Data, Learning with Hidden Variables, Introduction to The EM Algorithm –Reinforcement Learning
- AI applications, Language Models, Information Retrieval, Information Extraction

----

Definition:

Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.

How does AI work?

As the hype around AI has accelerated, vendors have been scrambling to promote how their products and services use AI. Often what they refer to as AI is simply one component of AI, such as machine learning. AI requires a foundation of specialized hardware and software for writing and training machine learning algorithms. No one programming language is synonymous with AI, but a few, including Python, R and Java, are popular.

Future of Artificial Intelligence:

Undoubtedly, Artificial Intelligence (AI) is a revolutionary field of computer science, which is ready to become the main component of various emerging technologies like **big data, robotics, and IoT**. It will continue to act as a technological innovator in the coming years. In just a few years, AI has become a reality from fantasy. Machines that help humans with intelligence are not just in sci-fi movies but also in the real world. At this time, we live in a world of Artificial Intelligence that was just a story though for some years.

Javatpoint Link:

https://www.javatpoint.com/future-of-artificial-intelligence

• Characteristics of intelligent agents:

Intelligent agents have the following distinguishing characteristics:

- They have some level of autonomy that allows them to perform certain tasks on their own.
- They have a learning ability that enables them to learn even as tasks are carried out.
- They can interact with other entities such as agents, humans, and systems.
- New rules can be accommodated by intelligent agents incrementally.
- They exhibit goal-oriented habits.
- They are knowledge-based. They use knowledge regarding communications, processes, and entities.

• The structure of intelligent agents:

The IA structure consists of three main parts: architecture, agent function, and agent program.

- **Architecture:** This refers to machinery or devices that consists of actuators and sensors. The intelligent agent executes on this machinery. Examples include a personal computer, a car, or a camera.

- **Agent function:** This is a function in which actions are mapped from a certain percept sequence. Percept sequence refers to a history of what the intelligent agent has perceived.
- **Agent program:** This is an implementation or execution of the agent function. The agent function is produced through the agent program's execution on the physical architecture.

Categories of intelligent agents:

There are 5 main categories of intelligent agents. The grouping of these agents is based on their capabilities and level of perceived intelligence.

Simple reflex agents:

These agents perform actions using the current percept, rather than the percept history. The condition-action rule is used as the basis for the agent function. In this category, a fully observable environment is ideal for the success of the agent function.

Model-based reflex agents:

Unlike simple reflex agents, model-based reflex agents consider the percept history in their actions. The agent function can still work well even in an environment that is not fully observable. These agents use an internal model that determines the percept history and effect of actions. They reflect on certain aspects of the present state that have been unobserved.

Goal-based agents:

These agents have higher capabilities than model-based reflex agents. Goal-based agents use goal information to describe desirable capabilities. This allows them to choose among various possibilities. These agents select the best action that enhances the attainment of the goal.

Utility-based agents:

These agents make choices based on utility. They are more advanced than goal-based agents because of an extra component of utility measurement. Using a utility function, a state is mapped against a certain measure of utility. A rational agent selects the action that optimizes the expected utility of the outcome.

- **Learning agents:**

These are agents that have the capability of learning from their previous experience. Learning agents have the following elements.

- **The learning element:** This element enables learning agents to learn from previous experiences.
- **The critic:** It provides feedback on how the agent is doing.
- **The performance element:** This element decides on the external action that needs to be taken.
- **The problem generator:** This acts as a feedback agent that performs certain tasks such as making suggestions (new) and keeping history.

How intelligent agents work:

Intelligent agents work through three main components: sensors, actuators, and effectors. Getting an overview of these components can improve our understanding of how intelligent agents work.

- **Sensors:** These are devices that detect any changes in the environment. This information is sent to other devices. In artificial intelligence, the environment of the system is observed by intelligent agents through sensors.
- **Actuators:** These are components through which energy is converted into motion. They perform the role of controlling and moving a system. Examples include rails, motors, and gears.

- **Effectors:** The environment is affected by effectors. Examples include legs, fingers, wheels, display screen, and arms.

- ## Applications of intelligent agents:

Intelligent agents in artificial intelligence have been applied in many real-life situations.

- **Information search, retrieval, and navigation:**

Intelligent agents enhance the access and navigation of information. This is achieved through the search of information using search engines. The internet consists of many data objects that may take users a lot of time to search for a specific data object. Intelligent agents perform this task on behalf of users within a short time.

- **Repetitive office activities:**

Some companies have automated certain administrative tasks to reduce operating costs. Some of the functional areas that have been automated include customer support and sales. Intelligent agents have also been used to enhance office productivity.

- **Medical diagnosis:**

Intelligent agents have also been applied in healthcare services to improve the health of patients. In this case, the patient is considered as the environment. The computer keyboard is used as the sensor that receives data on the symptoms of the patient. The intelligent agent uses this information to decide the best course of action. Medical care is given through actuators such as tests and treatments.

- **Vacuum cleaning:**

AI agents are also used to enhance efficiency and cleanness in vacuum cleaning. In this case, the environment can be a room, table, or carpet. Some of the sensors employed in vacuum cleaning include cameras, bump sensors, and dirt detection sensors. Action is initiated by actuators such as brushes, wheels, and vacuum extractors.

- **Autonomous driving:**

Intelligent agents enhance the operation of self-driving cars. In autonomous driving, various sensors are employed to collect information from the environment. These include cameras, GPS, and radar. In this application, the environment can be pedestrians, other vehicles, roads, or road signs. Various actuators are used to initiate actions. For example, brakes are used to bring the car to a stop.

Section.io Link:

https://www.section.io/engineering-education/intelligent-agents-in-ai/

- **Problem Solving in Artificial Intelligence:**

The reflex agent of AI directly maps states into action. Whenever these agents fail to operate in an environment where the state of mapping is too large and not easily performed by the agent, then the stated problem dissolves and sent to a problem-solving domain which breaks the large stored problem into the smaller storage area and resolves one by one. The final integrated action will be the desired outcomes.

- **Steps problem-solving in AI:**

The problem of AI is directly associated with the nature of humans and their activities. So we need a number of finite steps to solve a problem which makes human easy works. These are the following steps which require to solve a problem :

- **Goal Formulation:** This one is the first and simple step in problem-solving. It organizes finite steps to formulate a target/goals which require some action to achieve the goal. Today the formulation of the goal is based on AI agents.
- **Problem formulation:** It is one of the core steps of problem-solving which decides what action should be taken to achieve the formulated goal. In AI this core part is dependent upon software agent which consisted of the following components to formulate the associated problem. Components to formulate the associated problem:

- **Initial State:** This state requires an initial state for the problem which starts the AI agent towards a specified goal. In this state new methods also initialize problem domain solving by a specific class.
    1. **Action:** This stage of problem formulation works with function with a specific class taken from the initial state and all possible actions done in this stage.
    2. **Transition:** This stage of problem formulation integrates the actual action done by the previous action stage and collects the final stage to forward it to their next stage.
    3. **Goal test:** This stage determines that the specified goal achieved by the integrated transition model or not, whenever the goal achieves stop the action and forward into the next stage to determines the cost to achieve the goal.
    4. **Path costing:** This component of problem-solving numerical assigned what will be the cost to achieve the goal. It requires all hardware software and human working cost.

GeeksforGeeks Link:

https://www.geeksforgeeks.org/problem-solving-in-artificial-intelligence/

• Search Algorithms in Artificial Intelligence:

Search algorithms are one of the most important areas of Artificial Intelligence. This topic will explain all about the search algorithms in AI.

• Problem-solving agents:

In Artificial Intelligence, Search techniques are universal problem-solving methods. **Rational agents** or **Problem-solving agents** in AI mostly used these search strategies or algorithms to solve a specific problem and provide the best result. Problem-solving agents are the goal-based agents and use atomic representation. In this topic, we will learn various problem-solving search algorithms.

• Search Algorithm Terminologies:

- **Search:** Searching is a step by step procedure to solve a search-problem in a given search space. A search problem can have three main factors:

  - **Search Space:** Search space represents a set of possible solutions, which a system may have.
  - **Start State:** It is a state from where agent begins **the search**.
  - **Goal test:** It is a function which observe the current state and returns whether the goal state is achieved or not.

1. **Search tree:** A tree representation of search problem is called Search tree. The root of the search tree is the root node which is corresponding to the initial state.
2. **Actions:** It gives the description of all the available actions to the agent.
3. **Transition model:** A description of what each action do, can be represented as a transition model.
4. **Path Cost:** It is a function which assigns a numeric cost to each path.
5. **Solution:** It is an action sequence which leads from the start node to the goal node.
6. **Optimal Solution:** If a solution has the lowest cost among all solutions.

Javatpoint Link:

https://www.javatpoint.com/search-algorithms-in-ai

- ## A* Search Algorithm:

To approximate the shortest path in real-life situations, like- in maps, games where there can be many hindrances.

- **What is A* search Algorithm?**

A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.

GeeksforGeeks Link:

https://www.geeksforgeeks.org/a-search-algorithm/

- ## **Uninformed, Informed, Heuristic:**

Baeldung Link:

https://www.baeldung.com/cs/informed-vs-uninformed-search

javatpoint Link:

https://www.javatpoint.com/ai-uninformed-search-algorithms

- ## **Hill Climbing Algorithm in Artificial Intelligence:**

- Hill climbing algorithm is a local search algorithm which continuously moves in the direction of increasing elevation/value to find the peak of the mountain or best solution to the problem. It terminates when it reaches a peak value where no neighbor has a higher value.
- Hill climbing algorithm is a technique which is used for optimizing the mathematical problems. One of the widely discussed examples of Hill climbing algorithm is Traveling-salesman Problem in which we need to minimize the distance traveled by the salesman.
- It is also called greedy local search as it only looks to its good immediate neighbor state and not beyond that.
- A node of hill climbing algorithm has two components which are state and value.
- Hill Climbing is mostly used when a good heuristic is available.
- In this algorithm, we don't need to maintain and handle the search tree or graph as it only keeps a single current state.

- ## **Features of Hill Climbing:**

Following are some main features of Hill Climbing Algorithm:

- **Generate and Test variant:** Hill Climbing is the variant of Generate and Test method. The Generate and Test method produce feedback which helps to decide which direction to move in the search space.
- **Greedy approach:** Hill-climbing algorithm search moves in the direction which optimizes the cost.
- **No backtracking:** It does not backtrack the search space, as it does not remember the previous states.

Baeldung Link:

https://www.javatpoint.com/hill-climbing-algorithm-in-ai

- ## Iterative deepening DFS:

Iterative deepening depth-first search (IDDFS) is an algorithm that is an important part of an Uninformed search strategy just like BFS and DFS. We can define IDDFS as an algorithm of an amalgam of BFS and DFS searching techniques. In IDDFS, We have found certain limitations in BFS and DFS so we have done hybridization of both the procedures for eliminating the demerits lying in them individually. We do a limited depth-first search up to a fixed "limited depth". Then we keep on incrementing the depth limit by iterating the procedure unless we have found the goal node or have traversed the whole tree whichever is earlier.

In order to implement the iterative deepening search we have to mark differences among:

- Breakdown as the depth limit bound was attained.
- A breakdown where depth bound was not attained.

While in the case once we try the search method multiple times by increasing the depth limit each time and in the second case even if we keep on searching multiple times since no solution exists then it means simply the waste of time. Thus we come to the conclusion that in the first case failure is found to be failing unnaturally, and in the second case, the failure is failing naturally.

- **Advantages:**
- IDDFS gives us the hope to find the solution if it exists in the tree.

- When the solutions are found at the lower depths say n, then the algorithm proves to be efficient and in time.
- The great advantage of IDDFS is found in-game tree searching where the IDDFS search operation tries to improve the depth definition, heuristics, and scores of searching nodes so as to enable efficiency in the search algorithm.

Educba.com Link:

https://www.educba.com/iterative-deepening-depth-first-search/

- ## Bidirectional Search:

Searching a graph is quite famous problem and have a lot of practical use. We have already discussed here how to search for a goal vertex starting from a source vertex using BFS. In normal graph search using BFS/DFS we begin our search in one direction usually from source vertex toward the goal vertex, but what if we start search from both direction simultaneously.

Bidirectional search is a graph search algorithm which find smallest path from source to goal vertex. It runs two simultaneous search –

- Forward search from source/initial vertex toward goal vertex
- Backward search from goal/target vertex toward source vertex

Bidirectional search replaces single search graph(which is likely to grow exponentially) with two smaller sub graphs – one starting from initial vertex and other starting from goal vertex. The search terminates when two graphs intersect.

- **Performance measures:**
- **Completeness :** Bidirectional search is complete if BFS is used in both searches.
- **Optimality :** It is optimal if BFS is used for search and paths have uniform cost.
- **Time and Space Complexity :** Time and space complexity is O(bd/2).

GeeksforGeeks Link:

https://www.geeksforgeeks.org/bidirectional-search/

- ## Backtracking Search:

A depth-first search that chooses values for one variable at a time and backtracks when a variable has no legal values left to assign. Backtracking algorithm repeatedly chooses an unassigned variable, and then tries all values in the domain of that variable in turn, trying to find a solution.

- ## Optimal Decision Making:

The optimal solution becomes a contingent strategy when specifies MAX(the player on our side)'s move in the initial state, then Max move to the states resulting for every possible response by MIN. Then MAX's moves in the states resulting from every possible response by MIN to those moves, and so on.

The minimax algorithm explores the game tree from top to bottom in depth-first. The temporal complexity of the minimax method is O if the maximum depth of the tree is m and there are b legal moves at each point (bm). For an algorithm that creates all actions at once, the space complexity is O(bm), while for an algorithm that generates actions one at a time, the space complexity is O(m) The time cost is obviously impractical for real games, but this technique serves as a foundation for game mathematics analysis and more practical algorithms.

- ## Alpha-Beta Pruning:

- Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.
- As we have seen in the minimax search algorithm that the number of game states it has to examine are exponential in depth of the tree. Since we cannot eliminate the exponent, but we can cut it to half. Hence there is a technique by which without checking each node of the game tree we can compute the correct minimax decision, and this technique is called **pruning**. This involves two threshold parameter Alpha and beta for future expansion, so it is called **alpha-beta pruning**. It is also called as **Alpha-Beta Algorithm**.
- Alpha-beta pruning can be applied at any depth of a tree, and sometimes it not only prune the tree leaves but also entire sub-tree.
- The two-parameter can be defined as:

   - **Alpha:** The best (highest-value) choice we have found so far at any point along the path of Maximiser. The initial value of alpha is **-∞**.
   - **Beta:** The best (lowest-value) choice we have found so far at any point along the path of Minimizer. The initial value of beta is **+∞**.
- The Alpha-beta pruning to a standard minimax algorithm returns the same move as the standard algorithm does, but it removes all the nodes which are not really affecting the final decision but making algorithm slow. Hence by pruning these nodes, it makes the algorithm fast.

Javatpoint Link:

https://www.javatpoint.com/ai-alpha-beta-pruning

• Stochastic Games in Artificial Intelligence:

Many unforeseeable external occurrences can place us in unforeseen circumstances in real life. Many games, such as dice tossing, have a random element to reflect this unpredictability. These are known as stochastic games. Backgammon is a classic game that mixes skill and luck. The legal moves are determined by rolling dice at the start of each player's turn white, for example, has rolled a 6–5 and has four alternative moves in the backgammon scenario.

GeeksforGeeks Link:

https://www.geeksforgeeks.org/stochastic-games-in-artificial-intelligence/

• First-Order Logic in Artificial intelligence:

In the topic of Propositional logic, we have seen that how to represent statements using propositional logic. But unfortunately, in propositional logic, we can only represent the facts, which are either true or false. PL is not sufficient to represent the complex sentences or natural language statements. The propositional logic has very limited expressive power. Consider the following sentence, which we cannot represent using PL logic.

- **"Some humans are intelligent", or**
- **"Sachin likes cricket."**

To represent the above statements, PL logic is not sufficient, so we required some more powerful logic, such as first-order logic.

• Basic Elements of First-order logic:

Following are the basic elements of FOL syntax:

Constant	1, 2, A, John, Mumbai, cat
Variables	x, y, z, a, b
Predicates	Brother, Father
Function	sqrt, LeftLegOf
Connectives	∧, ∨, ¬, ⇒, ⇔
Equality	==
Quantifier	∀, ∃

Javatpoint Link:

https://www.javatpoint.com/first-order-logic-in-artificial-intelligence

• Unification:

Unification is a process of making two different logical atomic expressions identical by finding a substitution. Unification depends on the substitution process. It takes two literals as input and makes them identical using substitution
In logic and computer science, unification is an algorithmic process of solving equations between symbolic expressions. ... If higher-order variables, that is, variables representing functions, are allowed in an expression, the process is called higher-order unification, otherwise first-order unification.
In computer science and logic, unification is the algorithmic procedure used in solving equations involving symbolic expressions. In other words, by replacing certain sub-expression variables with other expressions, unification tries to identify two symbolic expressions.
The Relational Resolution Principle is analogous to that of propositional resolution. The main difference is the use of unification to unify literals before applying the rule.

Kcpelearning Link:

https://kcpelearning.com/showcourse/AI/unification-in-ai

- ## Forward Chaining and backward chaining in AI:

In artificial intelligence, forward and backward chaining is one of the important topics, but before understanding forward and backward chaining lets first understand that from where these two terms came.

- ## Inference engine:

The inference engine is the component of the intelligent system in artificial intelligence, which applies logical rules to the knowledge base to infer new information from known facts. The first inference engine was part of the expert system. Inference engine commonly proceeds in two modes, which are:

- **Forward chaining**
- **Backward chaining**

- ### Horn Clause and Definite clause:

Horn clause and definite clause are the forms of sentences, which enables knowledge base to use a more restricted and efficient inference algorithm. Logical inference algorithms use forward and backward chaining approaches, which require KB in the form of the **first-order definite clause**.

- **Definite clause:** A clause which is a disjunction of literals with **exactly one positive literal** is known as a definite clause or strict horn clause.
- **Horn clause:** A clause which is a disjunction of literals with **at most one positive literal** is known as horn clause. Hence all the definite clauses are horn clauses.
- **Example: (¬ p V ¬ q V k)**. It has only one positive literal k.
- It is equivalent to p ∧ q → k.

Javatpoint Link:

https://www.javatpoint.com/forward-chaining-and-backward-chaining-in-ai

- ## Resolution Method in AI:

Resolution method is an inference rule which is used in both Propositional as well as First-order Predicate Logic in different ways. This method is basically used for proving the satisfiability of a sentence. In resolution method, we use Proof by Refutation technique to prove the given statement.

- ## Resolution Method in Propositional Logic:

In propositional logic, resolution method is the only inference rule which gives a new clause when two or more clauses are coupled together.
Using propositional resolution, it becomes easy to make a theorem prover sound and complete for all.
The process followed to convert the propositional logic into resolution method contains the below steps:

- Convert the given axiom into clausal form, i.e., disjunction form.
- Apply and proof the given goal using negation rule.
- Use those literals which are needed to prove.
- Solve the clauses together and achieve the goal.

But, before solving problems using Resolution method, let's understand two normal forms

Tutorialandexample Link:

https://www.tutorialandexample.com/resolution-method-in-ai

- ## Introduction to Ontologies:

OWL is built on RDFS which helps us to define ontologies.
Ontologies are formal definitions of vocabularies that allow us to define difficult or complex structures and new relationships between vocabulary terms and members of classes that we define. Ontologies generally describe specific domains such as scientific research areas.

- ### Components:

- **Individuals –**
  Individuals are also known as instances of objects or concepts. It may or

may not be present in an ontology. It represents the atomic level of an ontology. For example, in the above ontology of movie, individuals can be a film (Titanic), a director (James Cameron), an actor (Leonardo DiCaprio).

- **Classes –**
  Sets of collections of various objects are termed as classes. For example, in the above ontology representing movie, movie genre (e.g. Thriller, Drama), types of person (Actor or Director) are classes.
- **Attributes –**
  Properties that objects may possess. For example, a movie is described by the set of 'parts' it contains like Script, Director, Actors.
- **Relations –**
  Ways in which concepts are related to one another. For example, as shown above in the diagram a movie has to have a script and actors in it.

• Mental Events & Mental Objects:

- **mental event** is any event that happens within the mind of a conscious individual. Examples include thoughts, feelings, decisions, dreams, and realizations. Some believe that mental events are not limited to human thought but can be associated with animals and artificial intelligence as well.
- **mental object** - the sum or range of what has been perceived, discovered, or learned. cognitive content, content. cognition, knowledge, noesis - the psychological result of perception and learning and reasoning. tradition - an inherited pattern of thought or action

• Reasoning in Artificial intelligence:

In previous topics, we have learned various ways of knowledge representation in artificial intelligence. Now we will learn the various ways to reason on this knowledge using different logical schemes.

• Reasoning:

The reasoning is the mental process of deriving logical conclusion and making predictions from available knowledge, facts, and beliefs. Or we can say, "Reasoning is a way to infer facts from existing data." It is a general process of thinking rationally, to find valid conclusions.

In artificial intelligence, the reasoning is essential so that the machine can also think rationally as a human brain, and can perform like a human.

- **Types of Reasoning:**

In artificial intelligence, reasoning can be divided into the following categories:

- Deductive reasoning
- Inductive reasoning
- Abductive reasoning
- Common Sense Reasoning
- Monotonic Reasoning
- Non-monotonic Reasoning

Javatpoint Link:

https://www.javatpoint.com/reasoning-in-artificial-intelligence

- **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.

- Information occurred from unreliable sources.
- Experimental Errors
- Equipment fault
- Temperature variation

- 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.

Javatpoint Link:

https://www.javatpoint.com/probabilistic-reasoning-in-artifical-intelligence#:~:text=Probability%3A%20Probability%20can%20be%20defined,probability%20of%20an%20event%20A.

- **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:

- $P(A \land B)= P(A|B) P(B)$ or

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

- $P(A \land B)= P(B|A) P(A)$

Javatpoint Link:

https://www.javatpoint.com/bayes-theorem-in-artifical-intelligence

• 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**.

Javatpoint Link:

https://www.javatpoint.com/bayesian-belief-network-in-artificial-intelligence#:~:text=%22A%20Bayesian%20network%20is%20a,decision%20network%2C%20or%20Bayesian%20model.

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.

GeeksforGeeks Link:

https://www.geeksforgeeks.org/markov-decision-process/

Forms of Learning:

Machine Learning (ML) is an automated learning with little or no human intervention. It involves programming computers so that they learn from the available inputs. The main purpose of machine learning is to explore and construct algorithms that can learn from the previous data and make predictions on new input data.

The input to a learning algorithm is training data, representing experience, and the output is any expertise, which usually takes the form of another algorithm that can perform a task. The input data to a machine learning system can be numerical, textual, audio, visual, or multimedia. The corresponding output data of the system can be a floating-point number, for instance, the velocity of a rocket, an integer representing a category or a class, for example, a pigeon or a sunflower from image recognition.

In this chapter, we will learn about the training data our programs will access and how learning process is automated and how the success and performance of such machine learning algorithms is evaluated.

- **<u>Concepts of Learning:</u>**

Learning is the process of converting experience into expertise or knowledge.

Learning can be broadly classified into three categories, as mentioned below, based on the nature of the learning data and interaction between the learner and the environment.

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning

Similarly, there are four categories of machine learning algorithms as shown below −

- Supervised learning algorithm
- Unsupervised learning algorithm
- Semi-supervised learning algorithm
- Reinforcement learning algorithm

However, the most commonly used ones are supervised and unsupervised learning.

- **<u>Supervised Learning:</u>**

Supervised learning is commonly used in real world applications, such as face and speech recognition, products or movie recommendations, and sales forecasting. Supervised learning can be further classified into two types - Regression and Classification.

- **<u>Unsupervised Learning:</u>**

Unsupervised learning is used to detect anomalies, outliers, such as fraud or defective equipment, or to group customers with similar behaviors for a sales campaign. It is the opposite of supervised learning. There is no labelled data here.

When learning data contains only some indications without any description or labels, it is up to the coder or to the algorithm to find the structure of the underlying data, to discover hidden patterns, or to determine how to describe the data. This kind of learning data is called unlabelled data.

- **Semi-supervised Learning:**

If some learning samples are labelled, but some other are not labelled, then it is semi-supervised learning. It makes use of a large amount of unlabelled data for training and a small amount of labelled data for testing.

- **Reinforcement Learning:**

Here learning data gives feedback so that the system adjusts to dynamic conditions in order to achieve a certain objective. The system evaluates its performance based on the feedback responses and reacts accordingly. The best known instances include self-driving cars and chess master algorithm AlphaGo.

Tutorialspoint Link:

https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_types_of_learning.htm

- # Parametric algorithms:

Parametric algorithms are based on a mathematical model that defines the relationship between inputs and outputs. This makes them more restrictive than nonparametric algorithms, but it also makes them faster and easier to train. Parametric algorithms are most appropriate for problems where the input data is well-defined and predictable.

Some examples of parametric algorithms include:

- **Linear regression** — Linear regression is used to predict the value of a target variable based on a set of input variables. It is often used for predictive modelling tasks, such as predicting the sales volume of a product based on historical sales data.
- **Logistic regression** — Logistic regression is used to predict the value of a target variable based on a set of input variables. It is often used for predictive modelling tasks, such as predicting the likelihood that a customer will purchase a product.
- **Neural network** — Neural networks are a type of machine learning algorithm that are used to model complex patterns in data. Neural networks are inspired by the workings of the human brain, and they can be used to solve a wide variety of problems, including regression and classification tasks.

• Nonparametric algorithms:

Nonparametric algorithms are not based on a mathematical model; instead, they learn from the data itself. This makes them more flexible than parametric algorithms but also more computationally expensive. Nonparametric algorithms are most appropriate for problems where the input data is not well-defined or is too complex to be modelled using a parametric algorithm.

Some examples of nonparametric algorithms include:

- **Decision trees** — Decision trees are a type of nonparametric machine learning algorithm that are used to model complex patterns in data. Decision trees are based on a hierarchical structure, and they can be used to solve a wide variety of problems, including regression and classification tasks.
- **Support vector machine** — A support vector machine is a type of machine learning algorithm that is used for data classification tasks. SVMs are particularly well-suited for tasks where the input data is linearly separable. This means that the data can be separated into distinct classes or groups using a linear boundary.

• Application of AI:

Artificial Intelligence has various applications in today's society. It is becoming essential for today's time because it can solve complex problems with an efficient way in multiple industries, such as Healthcare, entertainment, finance, education, etc. AI is making our daily life more comfortable and fast.

- **AI in Astronomy:**
- Artificial Intelligence can be very useful to solve complex universe problems. AI technology can be helpful for understanding the universe such as how it works, origin, etc.
- **AI in Healthcare:**
- In the last, five to ten years, AI becoming more advantageous for the healthcare industry and going to have a significant impact on this industry.
- Healthcare Industries are applying AI to make a better and faster diagnosis than humans. AI can help doctors with diagnoses and can inform when patients are worsening so that medical help can reach to the patient before hospitalization.

Javatpoint Link:

https://www.javatpoint.com/application-of-ai

- ## Natural Language Processing:

Natural Language Processing (NLP) refers to AI method of communicating with an intelligent systems using a natural language such as English.

Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc.

The field of NLP involves making computers to perform useful tasks with the natural languages humans use. The input and output of an NLP system can be –

- Speech
- Written Text

- **Components of NLP:**

There are two components of NLP as given −

- **Natural Language Understanding (NLU):**

Understanding involves the following tasks −

- Mapping the given input in natural language into useful representations.
- Analysing different aspects of the language.

- **Natural Language Generation (NLG):**

It is the process of producing meaningful phrases and sentences in the form of natural language from some internal representation.

It involves −

- **Text planning −** It includes retrieving the relevant content from knowledge base.
- **Sentence planning −** It includes choosing required words, forming meaningful phrases, setting tone of the sentence.
- **Text Realization −** It is mapping sentence plan into sentence structure.

The NLU is harder than NLG.

−−−−−−−−−−−