Chapter-01

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(1) What does Artificial Intelligence mean?

Artificial → human made

Intelligence → Thinking power

It is a branch of computer science by which we can create intelligence machine which can behave like human, think like human and able to make decision.

(2) Goals of AI:

1) Replicate human intelligent.
2) Solve knowledge intensive task.
3) An intelligent connection of perception and action.
4) Creating some system which can exhibit intelligent behaviours - Learn new things by itself, demonstrate explain and advice to user.
5) Building a machine which can perform tasks that require human intelligent as -
   - → providing a theorem
   - → playing game (Chess)
   - → performing surgical operation
   - → Driving a car in traffic

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(3) What are the learning outcomes of AI?

Human AI-interaction:

Design user interface to improve human-ai and real time decition making.

Design and evaluate conversational interface

AI solution and Application

unstructured & unaccurat data automatic solution

AI Design and Development

Analysis data with supervised-unsupervised

AI & Organization

AI system to meet business

AI Bot to automate organizational process from end to end.

④ Define AI within 4 different paradigms with example.

2| Acting Humanly:

Turing Test approach

2| Thinking Humanly:

The cognitive modelling approach

3| Thinking Rationally:

Laws of thought → ~~agent give best performance by achieve~~

8| Acting Rationally:

Rational agent approach → Optimal decision to achieve specific goal.

(5)

agent:

An agent is anything that can be viewed as perceiving its environment through sensors and ~~environment~~ acting upon that environment through actuators.

Agent program:

~~The set ability~~

The set of rules or algorithms governing the agents behaviour and decision making.

Rationality:

The ability of an agent to choose actions that maximize the likelihood of achieving its goal.

Autonomy:

The degree to which an agent can operate independently without direct ~~direct~~ human intervention (হস্তক্ষেপ).

Sun | Mon | Tue | Wed | Thu | Fri

Date :...../...../.....

Deterministic:

A system or process where the outcome is entirely predictable and may vary.

Stochastic:

~~Involving~~

A system or process where the outcome is not entirely predictable and may vary.

(6) What is PEAS in specifying the task-environment?

PEAS is a type of model on which an AI agent works upon. When we define an AI agent or rational agent then we can group its properties under PEAS representation model. It is made up of four words.

P: performance measure

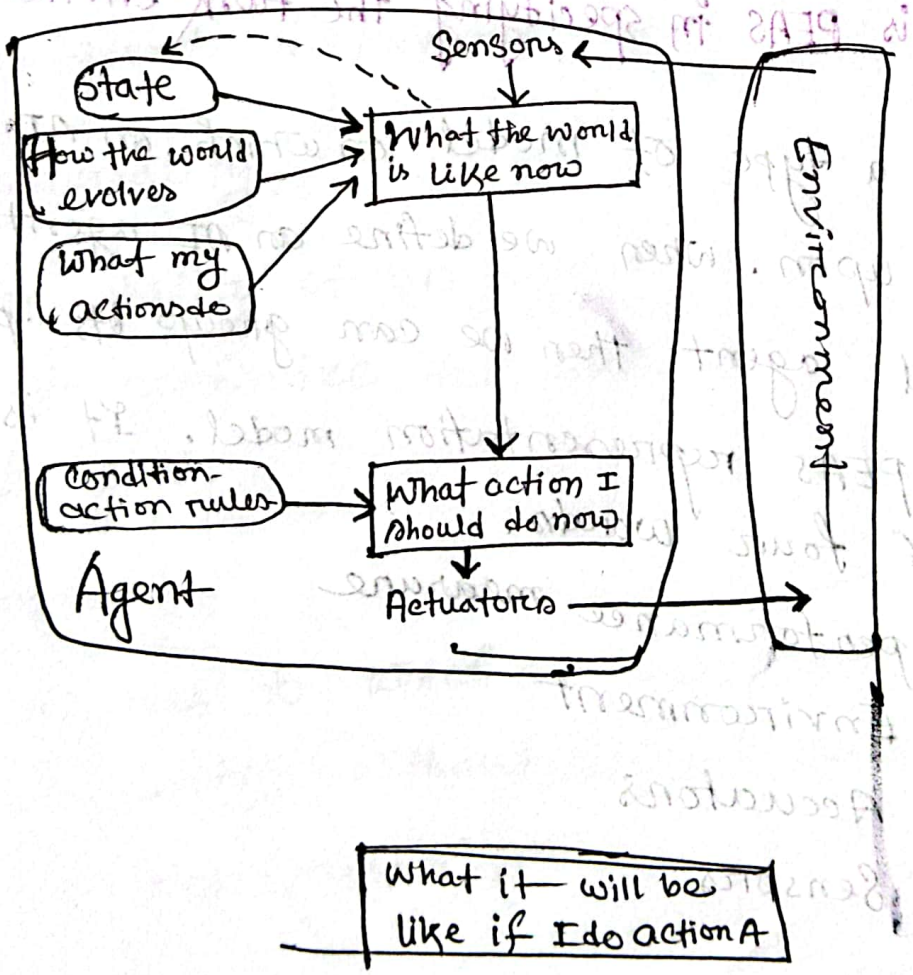
E: Environment

A: Accuators

S: Sensors

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi Driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, Other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, Speedmeter, GPS, accelerometer, engine sensors, keyboard

7. Illustrate and describe the structure of the model-based reflex agent.



Goal →

Goal base

Chapter-03

① Blind Search: (1)

Uninformed searches are search algorithms that explore a problem space without using any specific knowledge or heuristics about the problem domain.

② How to evaluate an algorithm's performance? (2)

We can evaluate an algorithms performance in four ways:

(I) Completeness:

A search algorithm is said to be complete if it guarantees to return a solution if at least any solution exists for any random input.

(II) Optimality:

If a solution found for an algorithm is guaranteed to be the best solution (lowest path cost) among all other solutions, then such a solution for is said to be an optimal solution.

(III) Time Complexity:

Time complexity is a measure of time for an algorithm to complete its task.

(IV) Space Complexity:

It is the maximum storage space required at any point during the search, as the complexity of the problem.

(3) Compare search strategies in terms of the four evaluation method. (3)

Criteria	Breadth First search	Uniform Cost	Depth First	Depth Limit	Iterative deepening	Bidirectional
Complete?	Yes	Yes	NO	NO	Yes	Yes
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$
optimal?	Yes	Yes	NO	NO	Yes	Yes

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(4) write the advantage of Informed search: (1)

i. use of heuristics → additional information

ii. More efficient

iii. Goal-directed → find a solution to a specific problem.

iv. Cost-based

v. prioritization

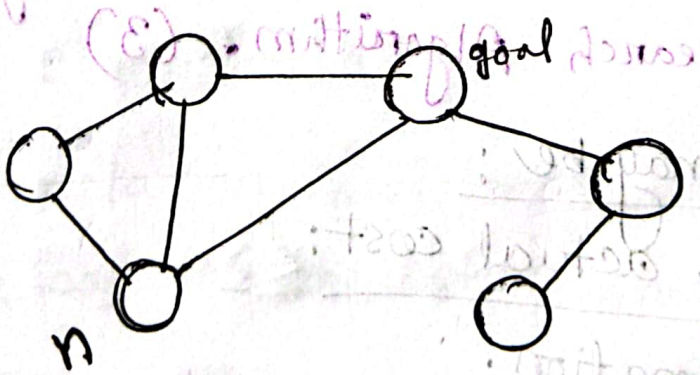
vi. optimality → admissible (never overestimating the actual cost)

vii. Quick solution.

viii. Less complexity. (Time, Space)

(5) How to make heuristic Function? (1)

h(n) = Estimate cost of the cheapest path from node n to goal node.



See question 10 also

7	2	4
5		6
8	3	1

Start state

—	1	2
3	4	5
6	7	8

Goal state

$h1$ = no. of displaced = 8

$h2 = 3+1+2+2+2+3+3+2 = 18$

$h(n) = h1 + h2$

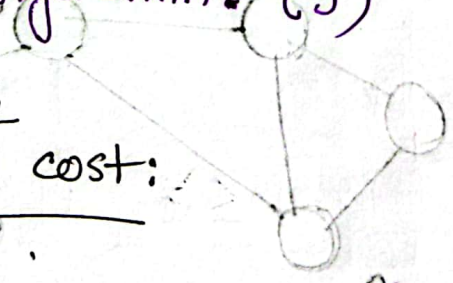
$= 8 + 18$

$= 26$

$f(n) = g(n) + h(n)$

misplaced

⑥ Define and illustrate admissible heuristic and consistent heuristic for estimating optimality of A* search Algorithm. (3)



Heuristic may be:

(1) $h(n) >$ actual cost:

overestimation.

optimal solution can be overlooked (উপেক্ষা করা)

optimal solution is not possible.

▶ show the heuristic must be consistent for the optimal solution in the A* search algorithm.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(2) $h(n)$ = actual cost:

Best case scenario, if $h(n)$ approximates actual cost, searching uses minimum of node to the goal.

(3) $h(n)$ < actual cost:

underestimation.

→ admissible heuristic: $h(n) \leq$ actual cost

→ Consistent heuristic: $h(n) \leq c(n,n') + h(n')$

▶ Admissible heuristic:

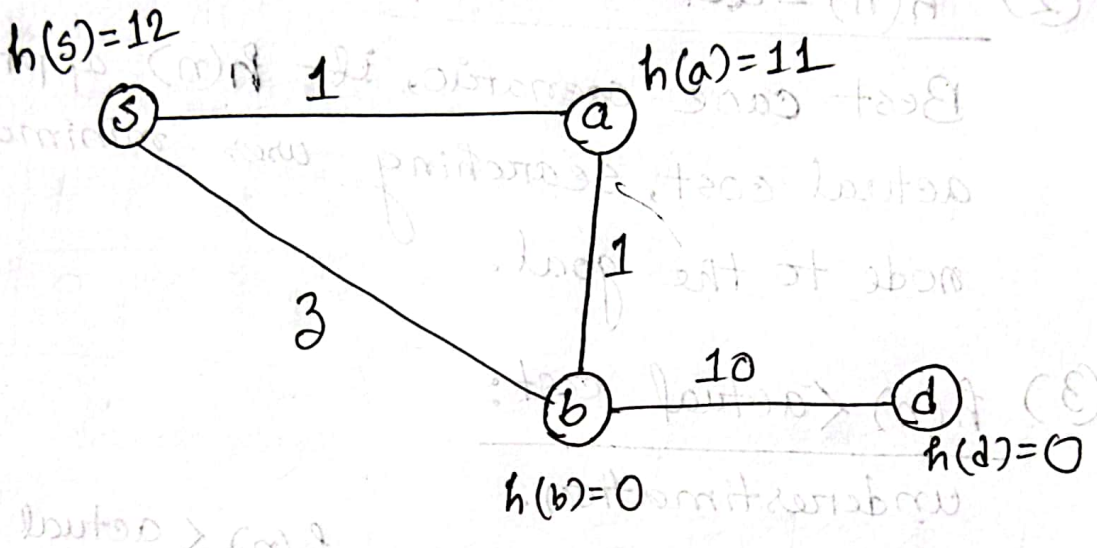
$h(n) \leq$ actual cost

If this relation maintains for every node, then we say this admissible heuristics.

▶ Consistent heuristic:

$h(n) \leq c(n,n') + h(n')$

if this relation maintains for every node then we say this consistent heuristic.



$h(n) \leq$ actual cost

$h(s) = 12 \leq 12$

$h(a) = 11 \leq 11$

$h(b) = 0 \leq 10$

~~So~~, condition satisfied for every node.

So, graph is admissible.

Again,

	$h(n)$	$c(n,n') + h(n')$
$s \to a$	12	$1+11=12$ $[h(n) \leq c(n,n')+h(n')]$
$s \to b$	12	$3+0=3$ $[h(n) \geq c(n,n')+h(n')]$

In the path $s \to b$ condition not satisfy.

So, graph is not consistent.

$f(n) = g(n) + h(n)$

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

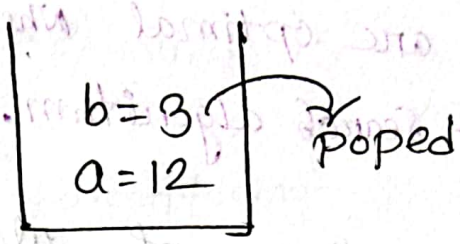
Applying A* search:

$a = 1 + 11$

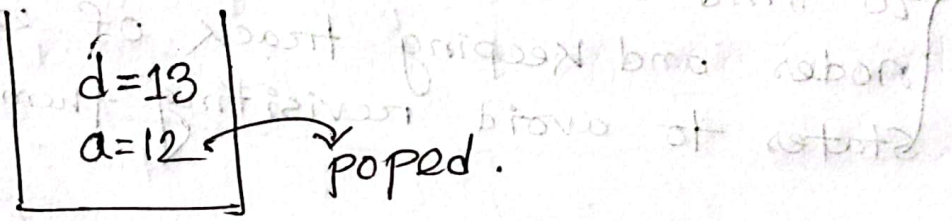
$= 12$

$b = 3 + 0$

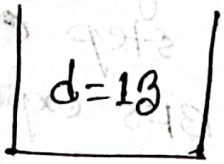
$= 3$



using priority queue we pop b.



Now, a is poped. but b already visited. So, b is in closed list because b poped before.



So, there only d in priority queue.

$d = 13$

but shortest distance is 12

If heuristic admissible but not consistent, it may not be provided optimum solution. For optimum solution, heuristic must be consistent. If it is consistent, it must be admissible.

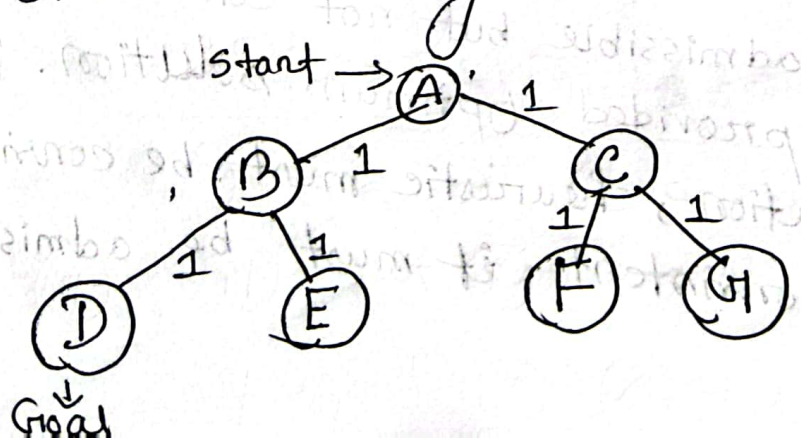
Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(7) prove that uniform cost search and breadth first search with constant step costs are optimal when used with the Graph-Search algorithm. (2)

The graph search algorithm is a general framework used for traversing a graph to find a solution. It involves expanding nodes and keeping track of explored states to avoid revisiting them.

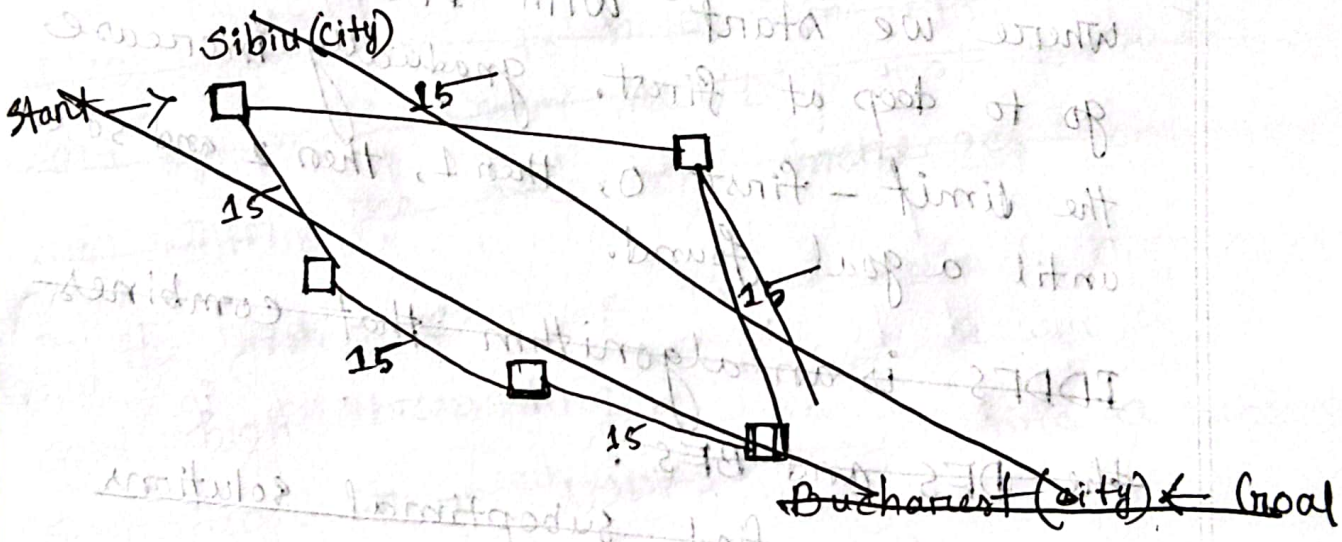
BFS (Breadth First Search) explores all nodes at the same depth level before moving deeper into the graph. When applied with constant step costs, BFS guarantees optimality. BFS explores nodes level by level, so if the step ~~cost~~ costs are constant, the solution found by BFS will always be shortest path.

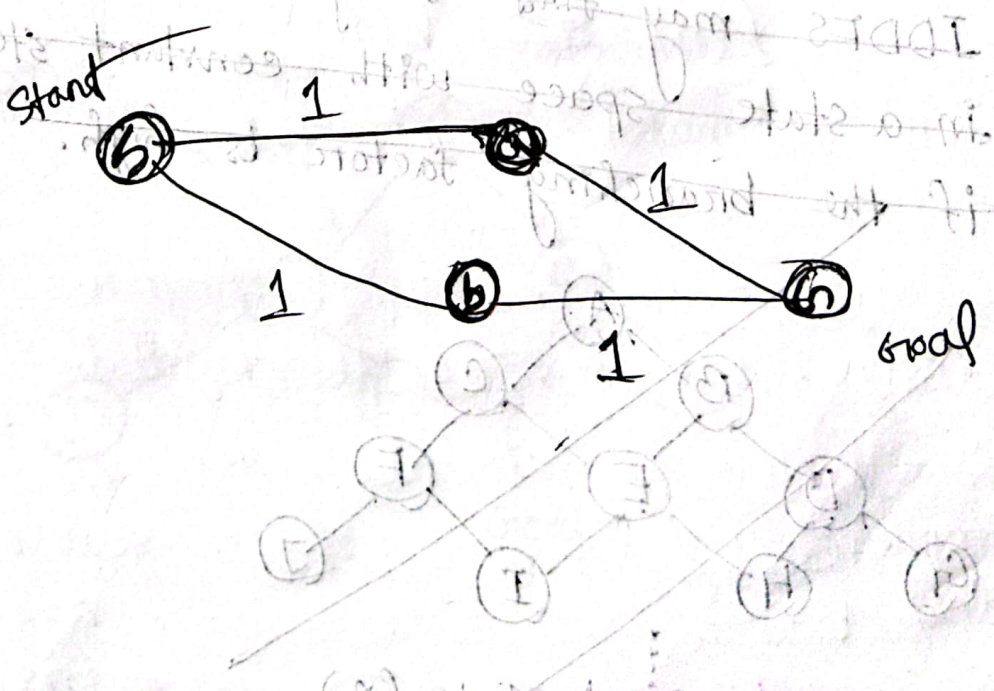


Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

Uniform cost search explores nodes with the lowest path cost from the start node. At each step it chooses the node with the lowest cost so far. So, if the step costs are same, guranteeing that the first solution it finds will have the lowest possible cost.





Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

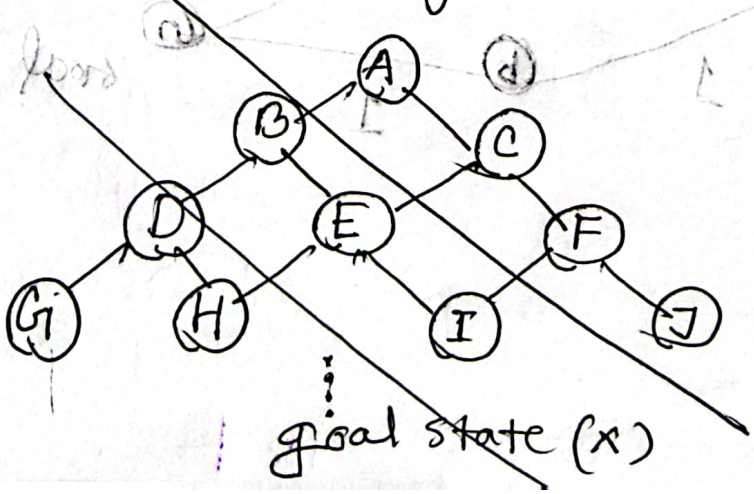
(8) Show a state space with constant step costs in which Graph-search using iterative deepening finds a suboptimal solution.

~~Iterative deepening depth first search (IDDFS) is an algorithm that combines the DFS and~~

Iterative deepening search is a strategy, where we start with DFS but don't go to deep at first. gradually increase the limit - first 0, then 1, then 2 and so on until a goal found.

~~IDDFS is an algorithm that combines the DFS and BFS.~~

~~IDDFS may find suboptimal solutions in a state space with constant step cost if the branching factor is high.~~



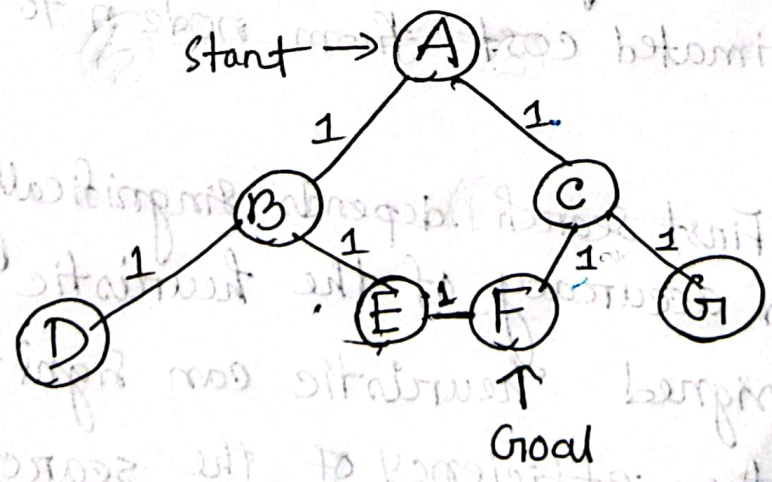
Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

The constant step cost and the high branching factore contribute to IDDFS finding suboptimal solution because it explores deeper path before exploring shallower path. IDDFS continues to deepen the search until it find a solution, potentially missing shorter paths along the way.

~~It's important to note that IDDFS is complete and optimal when the step cost is not constant or when the step cost increases with depth.~~

In the given example, the constant step cost and high branching factor create a scenario. where IDDFS may find a suboptimal solution.



Sat / Sun / Mon / Tue / Wed / Thu / Fri

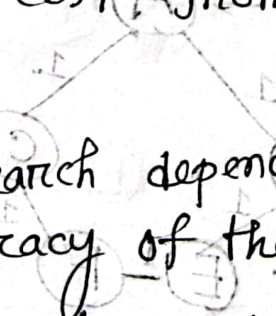
Date :...../...../.....

(9) How to minimize the total estimated solution cost using the best first search, A* search algorithm. Show the heuristic must be consistent for the optimal solution in the A* search algorithm.

Greedy Best first search always selects the path which appears best at that moment. It is the combination of BFS and DFS. With the help of best first search at each step we can choose the most promising node. In the best first search algorithm, we expand the node which is closest to the goal node and the closest cost is estimated by heuristic function,

$$f(n) = h(n)$$

$h(n)$ = estimated cost from node n to the goal.



Best-First search depends significally on the quality and accuracy of the heuristic function. A well designed heuristic can significantly improve the efficiency of the search process and lead to lower estimated solution costs. + বাকি অংশ (06) নং এর অনুরূপ।

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(10) Show that the 8 puzzle states are divided into two disjoint sets, such that no state in one set can be transformed into a state in the other set by any number of moves.

→ (5) নং এর অনুরূপ

Chapter-04

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

① Why do we use a local search strategy to address the optimization problem?

A local search algorithm is a type of optimization algorithm that is used to find an optimal solution for a particular problem with a small search space. Here are some reasons why local search strategies are commonly used for optimization problem.

① Large search space:

In many optimization problems, the solution space is vast and evaluating all possible solution is computationaly in feasible. Local search strategies explore a subset of the solution space, making them more scalable for large search space.

② Continous & Discrete optimization:

Local search can be applied to both continous and discrete optimization problems.

Date :...../...../.....

③ **Efficiency:**
Local search algorithms are often computationaly efficient.

④ Greedy nature.

⑤ Local optima

⑥ Adabtability

⑦ online & Real time optimization

⑧ Stochastic nature.

A local search algorithm starts from a conditional solution and then iteratively moves to a neighbour solution.

② What are the key advantages of local search algorithm?

There are two key advantages:

(1) they use very little memory - usually a constant amount.

(2) They can often find reasonable solutions in large or infinite (continuous) state spaces for which systematic algorithms are unsuitable.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

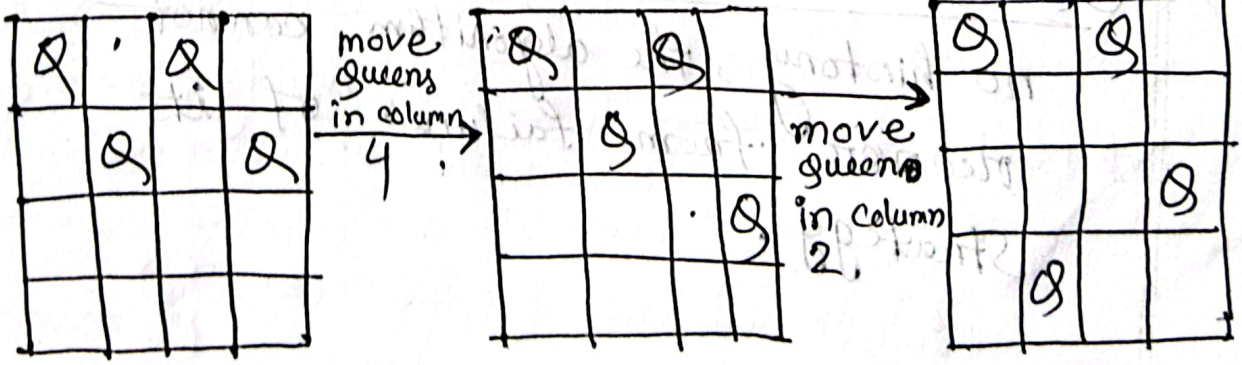
(3) Show how the last configuration of 4 queens on a 4x4 board has fewer conflicts than the first configuration using a local search strategy in where conflicts mean there are no two queens on the same row, column ~~and~~ or diagonal.

Goal: put 4 queens on an 4x4 board with no two queens on the same row, column or diagonal.

State space: All configurations with the queens in distinct columns.

State transition: Move a queen from its present place to some other square in the same column.

local search: Start with a configuration and repeatedly use the moves to reach the goal.



The last configuration has fewer conflicts than the first, but still not a solution.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(4) What are the reasons and problems of the hill climbing algorithm for getting stuck?

The standard version of hill climbing has some limitations and often gets stuck in the following scenario:

(I) Local maxima:

a local maxima is a peak that is higher than each of its neighboring states but lower than the global maximum. Hill climbing algorithms that reach the local maxima but will then be stuck with nowhere else to go.

(II) Cannot recover from failure:

no history, the algorithm cannot recover from failures of its strategy.

(III) Multiple local maxima:

Hill climbing functions can have multiple local maxima, which frustrates hill-climbing methods.

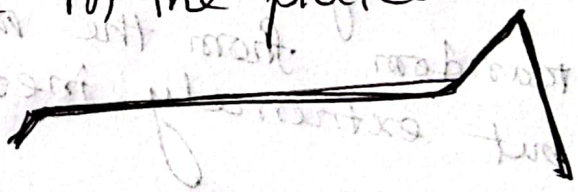
(IV) Stuck on plateaux & ridges:

Ridges:

There are sequence of local maxima making it difficult for algorithm to navigate.

Plateaux:

As there is no uphill to go algorithm often gets lost on the plateaux.



Local search algorithm:

A local search algorithm starts from a conditional solution and then iteratively moves to neighbour solution.

Date :...../...../.....

(5) How to escape these problems, using the simulated-annealing search algorithm?

A hill climbing algorithm that never makes downhill moves towards states with lower value is guranted to be incomplete because it can get stuck on a local maxima.

In a contrast a purely random walk that is moving to a successor chosen uniformly at random from the set of successors - is complete but extremely inefficient.

Therefore it seems reasonable to try to combine hill climbing with a random walk in some way that yields both efficiency and completeness. ~~Simulated annealing~~

- Escape ~~bad moves~~ local maxima by allowing some bad moves but gradually decrease their probability.

→ probability controlled by a parameter called Temperature

→ higher temp. allow more bad moves than lowertemp.

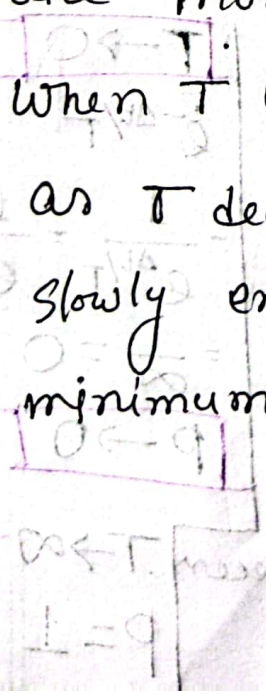
Simulated annealing is working process.

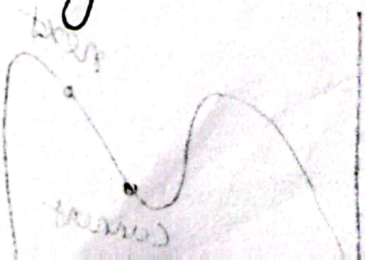
Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

The Simulated annealing solution is to start by shaking hard (high temperature) and then gradually reduce the intensity of shaking (lower the temperature).

Instead of picking the best move, however it pies a random moves. If the moves improves the situation. It is always ~~expected~~. accepted. Otherwise the algorithm accepts the move with some probability less than 1. The probability decreases exponentially with the badness of the move - the amount ΔE by which the evaluation is worsened. The probability also decreases as the temperature T goes down: bad moves are more likely to be allowed at the start when T is high and they become more unlikely as T decreases. If the schedule lowers T slowly enough, the algorithm will find a global minimum with probability approaching 1.





Date :...../...../.....

function Simulated-Annealing (problem, schedule)

return a solution state

input: problem, a problem

schedule, a mapping from time to temperature

current ← Make node (problem. initial-State)

for $t = 1$ to ∞ do

$T \leftarrow$ schedule (t)

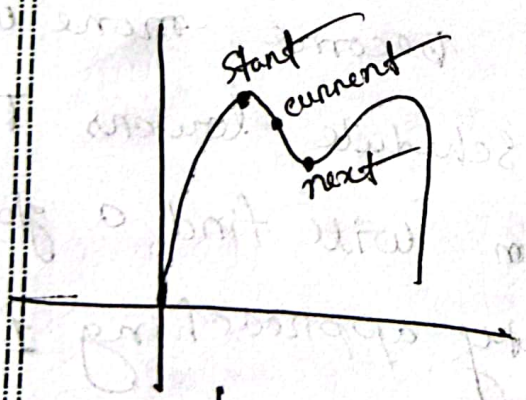
if $T = 0$ then return current

next ← a randomly selected successor of current

$\Delta E \leftarrow$ next.value − current.value

if $\Delta E > 0$ then current ← next

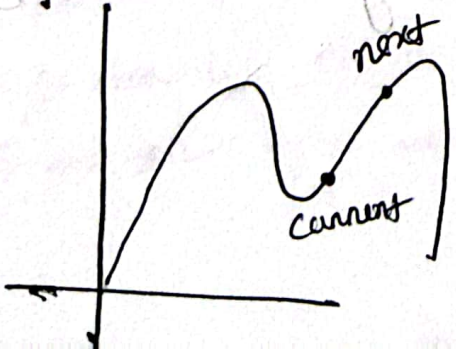
else current ← next only with probability $e^{\Delta E/T}$



current t1

next t2

t2 − t1 = annealing



current t1

next t2

t2 − t1 = Random access

$T \to 0$ — Temperature zero

$e^{-\Delta E/T}$

$= \frac{1}{e^{\Delta E/T}} = \frac{1}{e^{\infty}}$

$= \frac{1}{\infty} = 0$

$p \to 0$ → bad moves

$T \to \infty$ — Temp. so high

$p = 1$

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

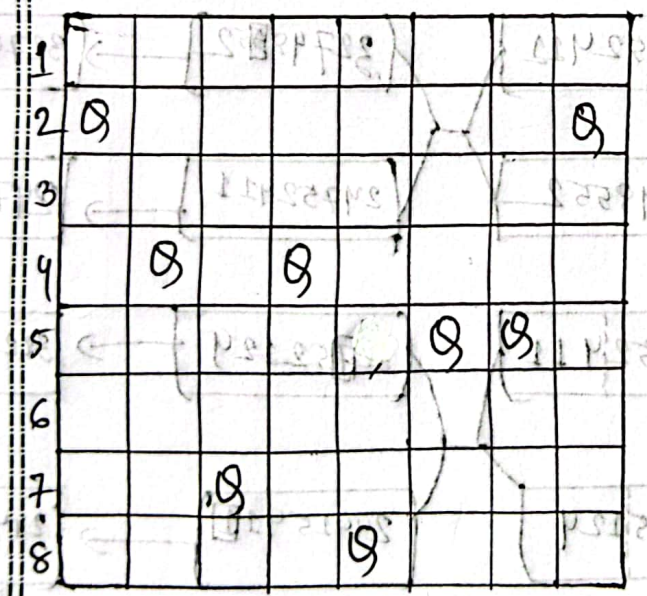
⑥ Illustrate and explain the genetic algorithm using digit string representation of 8 queen states.

Consider the following individuals states for 8 queen problem

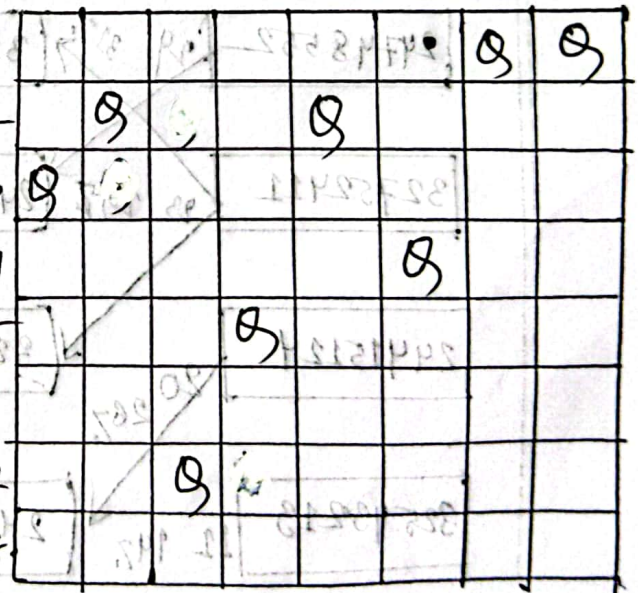
State 1	State 2	State 3	State 4
24748552	32752411	24415124	32543213

$\frac{n(n-1)}{2}$

$\frac{8\times7}{2} = 28$



24 - Non attacking pair



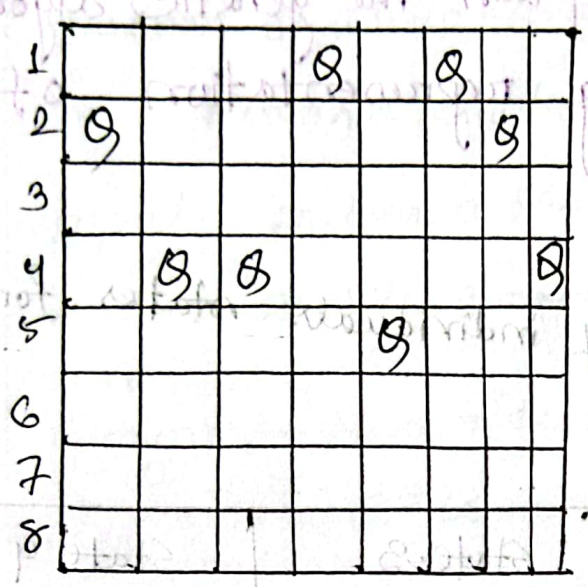
23 - non-attacking pair

• Population fitness = 24+23+20+11 = 78

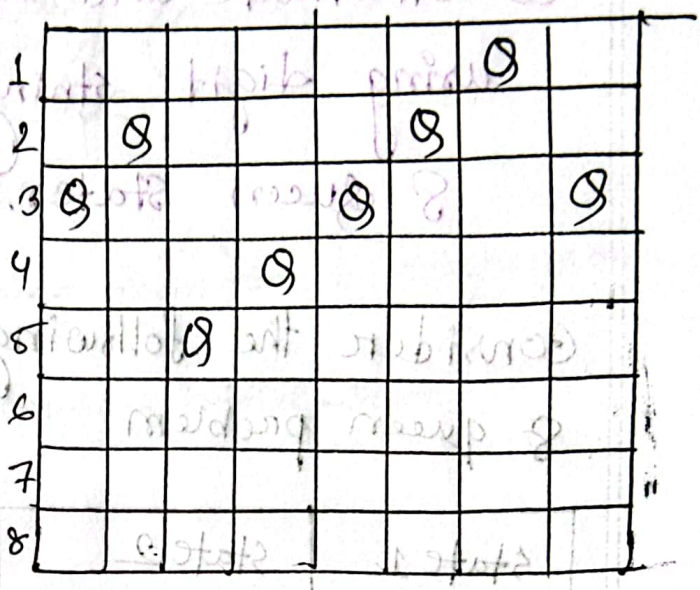
P(24/78) = 31%

Sat / Sun / Mon / Tue / Wed / Thu / Fri

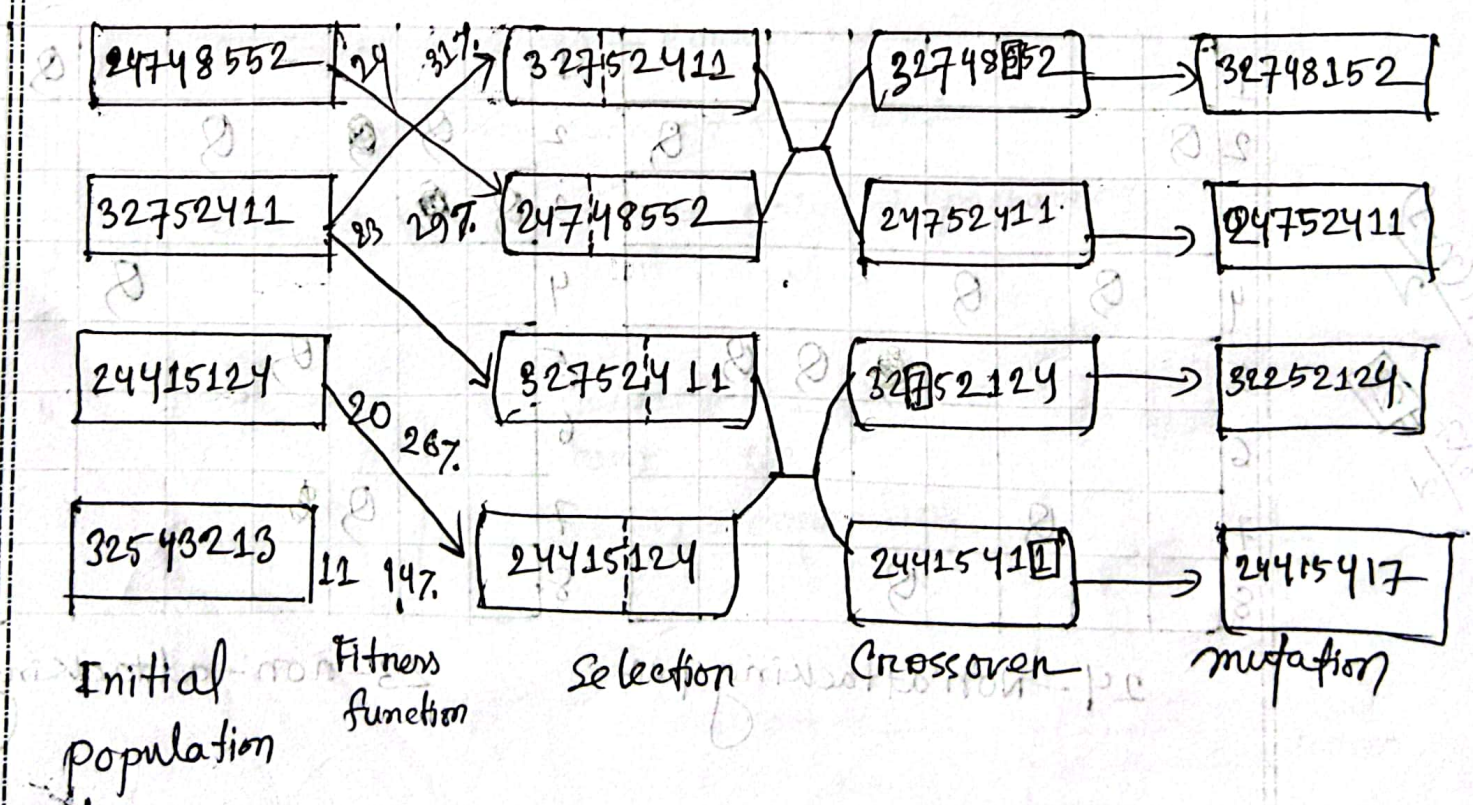
Date :...../...../.....



20 non attacking pair



11 non attacking pairs



Step 01: initialize population

Step 02: Selection according to fitness

Step 03: Cross over between selected Chromosoms

Step 04: perform mutation

Step 05: Repeat cycle till the condition of stop is true.

Chapter-06

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(1) Define constraint satisfaction problem.

A constraints satisfiction problem consists of three components X, D and C

X is a set of variables $\{x_1, \ldots, x_n\}$

D is a set of Domains $\{D_1, \ldots, D_n\}$ one for each variable

C is a set of constraints that specify allowable combinations of values.

Each Domain D_i consist of a set of allowable values $(v_1, \ldots, v_k)$ for variable x_i. Each Constraints C_i consists of a pair <scope, rel>

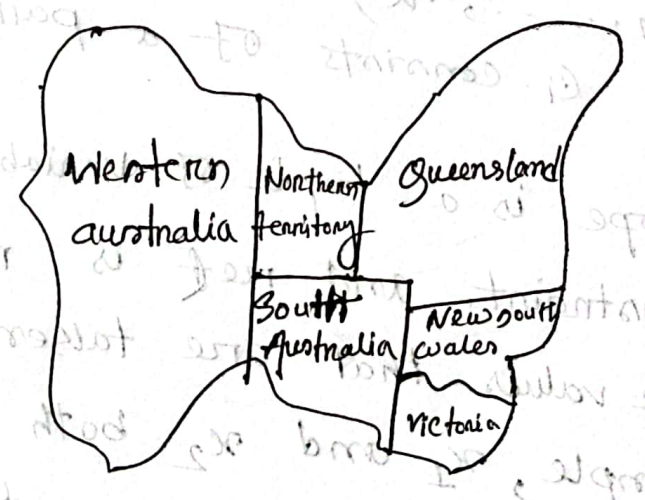
Where scope is a tuple of variable that participate in the constraint. and rel is relation that defines the values that are taken by variable. For example, x_1 and x_2 both have Domain $\{A, B\}$ The constraint saying the two variables must have different value can written as $\langle (x_1, x_2), [(A,B), (B,A)] \rangle$

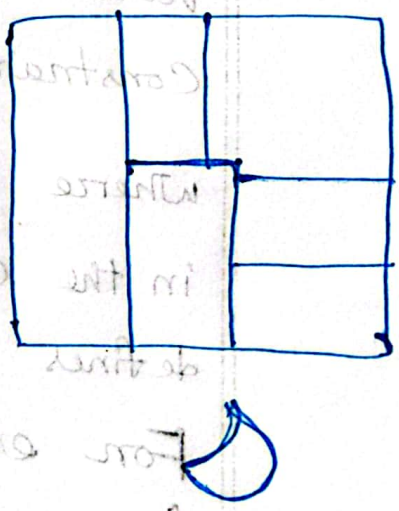
Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(2) Represent the map coloring problem with a constraint graph.

We are looking at a map of australia showing each of its state and territories. We are given the task of coloring each region either Red, green or blue in such way that no neighbouring region have the same colon. To formulate this as a csp.





We define the variables to be the region

$X = \{WA, NT, Q, NSW, SA, V, T\}$

The Domain of each variable set

$D_i = \{ red, green, blue \}$

The constraints requires neighboring region to have distinct color. Since there are nine places where rigions border there are nine constraints.

$C = \{ SA \neq WA, SA \neq NT, SA \neq Q, SA \neq NSW,$
$SA \neq , WA \neq NT, NT \neq Q, Q \neq NSW, NSW \neq V \}$

The possible combination are -

{(red, green) (red, blue) (green, red) (green, blue)
(blue, red) (blue, green)}

There are many possible solutions to this problem such as:

{ WA = red, NT = green, Q = Red, NSW = green,
V = Red, SA = blue, T = red }

It can be helpful to visualize a csp as a constraint graph.

Chapter-13

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

Conditional probability / Postercion:

posterciore probability is a type of conditional probability that results from updating the prior probability with information.

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

Marginal probability:

The probability of an event occuring $P(A)$.

It may be thought of as an unconditional probability.

Normalization:

A process by which it can possible probabilities sum to 1.

If B is a boolean random variables then

$$P(B) = \langle P(b), P(\neg b) \rangle$$

$$P(\text{cavity}) = \alpha \cdot \langle 0.01, 0.09 \rangle \quad \alpha = 10$$

here, α is a normalization constant.

Date :...../...../.....

Independent probability :

Independent probability refers to two events being unrelated where the occurence or non occurence of one event does not influence the probaby of the other event happening.

event A and B are independent if

$P(A \cap B) = P(A) \times P(B)$

Decision Theory :

A set of quantative method for reaching optimal solution.

(probability Theory + Utility theory)

Random variable :

Random variable is a variable that represents possible outcomes of a random uncertain process.

It could take values in discrete cases or continuous cases.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

Bayes' rule :

It provides a way to update the probability of a hypothesis based on new evidence.

The formula for the Baye's Rule is as follows:

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

$P(A|B)$ is the probability of A given the evidence B

$P(A)$ is the prior probability of A.

$P(B)$ is the prior probability of B.

Compute the patient's probability of having the liver disease if they are an alcoholic.

Being an alcoholic is the test for liver disease. past data tells you that 10% of patients entering your clinic have liver disease and 5% of the clinic's patients are alcoholics. You might also know that among these patients diagnosed with liver disease, 7% are alcoholic.

$P(a|ld) = 0.07$

$P(ld) = 0.1$

$P(a) = 0.05$

$$P(ld|a) = \frac{P(a|ld)\,P(ld)}{P(a)}$$

$$= \frac{0.7 \times 0.1}{0.05}$$

$$= 0.14$$

probability of liver disease is 14% if they are an alcoholic.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

Design a naive Bayes model, Bayesian classifier based on the dentistry example.

$$P(Cavity \mid toothache \wedge catche)$$

$$= \frac{P(toothache \wedge catch \mid Cavity)\, P(cavity)}{P(toothache \wedge catch)}$$

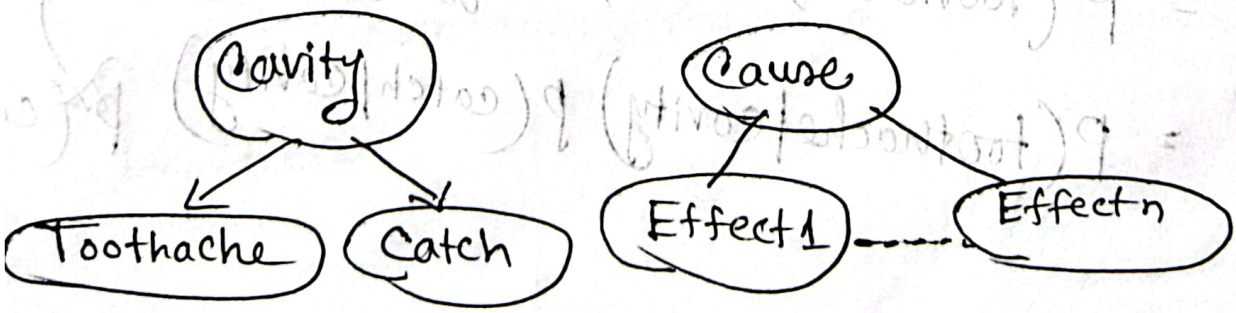
$$= \alpha\, P(toothache \wedge catch \mid Cavity)\, P(Cavity)$$

$$= \alpha\, P(toothache \mid cavity)\, P(catch \mid cavity)\, P(cavity)$$

A naive Bayes model is a mathematical model that assumes the effects are conditionally independent given the cause

$$P(cause, Effect_1 \ldots Effect_n)$$

$$= P(cause) \prod_i P(Effect_i \mid cause)$$



Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

Bayesian classifier Based on the dentistry example

The general defination of conditional independence of two variable x and y given a third variable z, is

$$P(x, y|z) = P(x|z)\, P(y|z)$$

In the dentist domain, for example it seems

$$P(\text{toothache}, \text{catch} \mid \text{cavity}) = P(\text{toothache} \mid \text{cavity}) \cdot P(\text{catch} \mid \text{cavity})$$

with absolute independence

$$P(x|y,z) = P(x|z), \; P(y|x,z) = P(y,z) \quad \text{--- (i)}$$

can also be used. For example, given the assertion in equation (i) we can drive a decomporition as follows -

$$P(\text{toothache}, \text{catch}, \text{cavity})$$

$$= P(\text{toothache}, \text{catch} \mid \text{cavity})\, P(\text{cavity}) \quad \text{[Product Rule]}$$

$$= P(\text{toothache} \mid \text{cavity})\, P(\text{catch} \mid \text{cavity})\, P(\text{cavity})$$

The dentistary example illustrates a commonly occurring pattern in which a single case directly influences a number of effect, all of which are conditionally independent given the case. The full joint distribution can be written as

$$P(\text{Cause}, \text{Effects} \ldots\ldots \text{Effects}_n)$$

$$= P(\text{cause}) \prod_i P(\text{Effects}_i \mid \text{cause})$$

Such a probability distribution is called a naive Bayes model. Naive because it is often used in cases where the effects variables are not actually conditionally independent given the cause variable. The naive bayes model is sometimes called a Bayesian classifier.

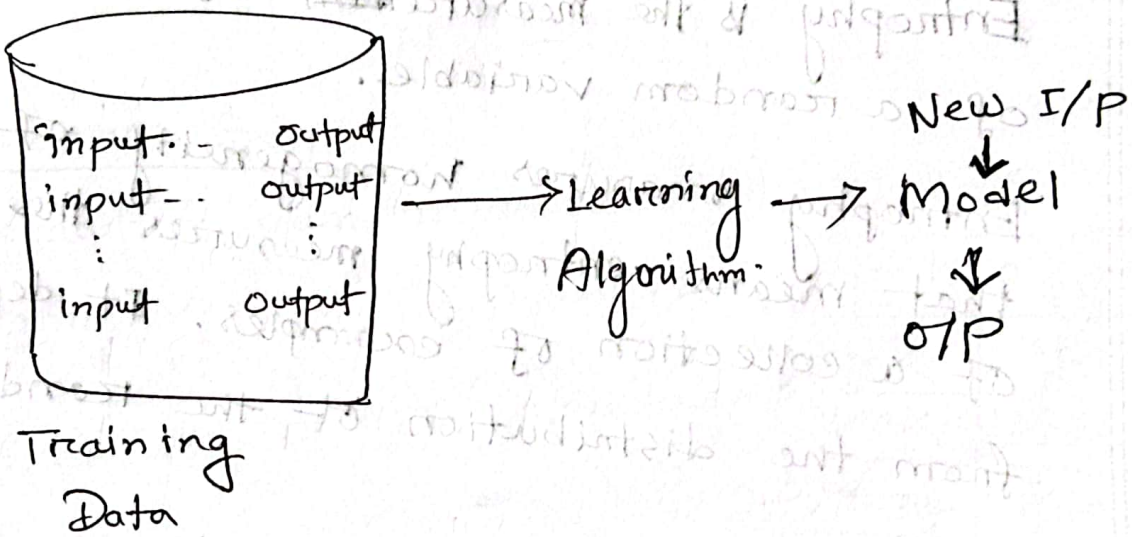
Chapter - 18

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(1) Supervised learning:

Supervised learning is the agent observes some example input-output pairs and learns a function that maps from input to output.



(2) How to learn decision trees using entropy and Information gain of attributes.

A decision tree represents a function that takes as input a vector of attribute values and return a decision - a single output value. The Input and output value can discrete or Continuous. For now we will concentrate

on problems where the inputs have discrete values and the output has exactly two possible values. Where each example input classified into true or false. A decision Tree reaches its decision by performing a sequence of tests.

Entrophy:

Entrophy is the measurement of the uncertainty of a random variable.

Entrophy measures homogeneity of examples that means entrophy measures the impurity of a collection of examples. It depends from the distribution of the random variable.

$$\text{Entrophy: } H(V) = -\sum_{k} P(v_k) \log_2 P(v_k)$$

entrophy of random variable V with values v_k.

- If all 14 variables positive

$$\text{Entrophy}([14+, 0-]) = -\left[\frac{14}{14}\log_2\left(\frac{14}{14}\right) + 0\log_2(0)\right] = 0$$

যদি সবগুলো positive হয় তাহলে entropy 0
অর্ধেক " / " " " entropy 1
বাকি ক্ষেত্রে calculation করতে হবে।

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

▶ If 9 variables positive and 5 variables negative.

$$\text{Entropy}([9+, 5-]) = -\left[\frac{9}{14}\log_2(9/14) + \frac{5}{14}\log_2(5/14)\right]$$

$$= 0.94$$

▶ If 7 variables positive and 7 variable negative

$$\text{Entropy}([7+, 7-]) = -\left[\frac{7}{14}\log_2(7/14) + \frac{7}{14}\log_2(7/14)\right]$$

$$= 1$$

Information Gain:

Information gain measures the expected reduction in entropy.

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

A = attribute

S = collection of examples

values (wind) = weak, strong

S = 9 positive, 5 negative

S_{weak} = 6 positive · 2 negative

S_{strong} = 3 positive, 3 negative

Wind	play tennis
Weak	Yes
Weak	NO
Strong	Yes
Strong	NO
Strong	Yes
Weak	Yes
Strong	NO
Strong	Yes
Weak	Yes
Weak	NO
Weak	Yes
Strong	NO
Weak	Yes
Yes	

$$Gain(S, wind) = Entropy(S) - \sum_{v \in \{weak, strong\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$= Entropy(S) - (8/14)\ Entropy(S_{weak}) - (6/14)\ Entropy(S_{strong})$$

$$= 0.940 - \left(\frac{8}{14}\right) 0.811 - \left(\frac{6}{14}\right) 1.00$$

$$= 0.048$$

(3) What is univariate linear regression?

Univariate linear regression refers to a linear regression model where we use one independent variable x to learn a linear function that maps our dependent variable.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(9) How to minimize the loss using gradient descent for fitting Linear regression?

Univariate linear function

$$y = w_1 x_0 + w_0$$

Where w_1 and w_0 are real valued coefficient to be learned.

$$h_w(x) = w_1 x + w_0$$

$$\text{Loss}(h_w) = \sum_{j=1}^{N} (y_i - (w_1 x_j + w_0))^2$$

w_1 and w_0 are obtained by minimizing the sum of the square of the errors.

$$\frac{\partial}{\partial w_0} = \sum_{j=1}^{N} (y_i - (w_1 x_j + w_0))^2 = 0$$

$$w_0 = \frac{(\Sigma y_j - w_1 \Sigma x_j)}{N}$$

$$\frac{\partial}{\partial w_1} = \sum_{j=1}^{N} (y_i - (w_1 x_j + w_0)) = 0$$

$$w_1 = \frac{N(\Sigma x_j y_j) - (\Sigma x_j)(\Sigma y_j)}{N(\Sigma x_j^2) - (\Sigma x_j)^2}$$

Will often have no closed form solution.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

We are trying to minimize the loss we will use gradient descent.

we choose only ~~about~~ starting point in weight-space here a point (w_0, w_1) plane and then we move to a neighbouring point that is downhill.

repeat until we converge on the minimum possible loss.

$w \leftarrow$ any point in the parameter space
loop until convergence do
  for each w_i in w do

$$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} \text{loss}(w)$$

The parameter α is called step size and usually it called learning rate.

We can find the minimum w_i.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

⑤ Distinguish generalization loss and empirical loss.

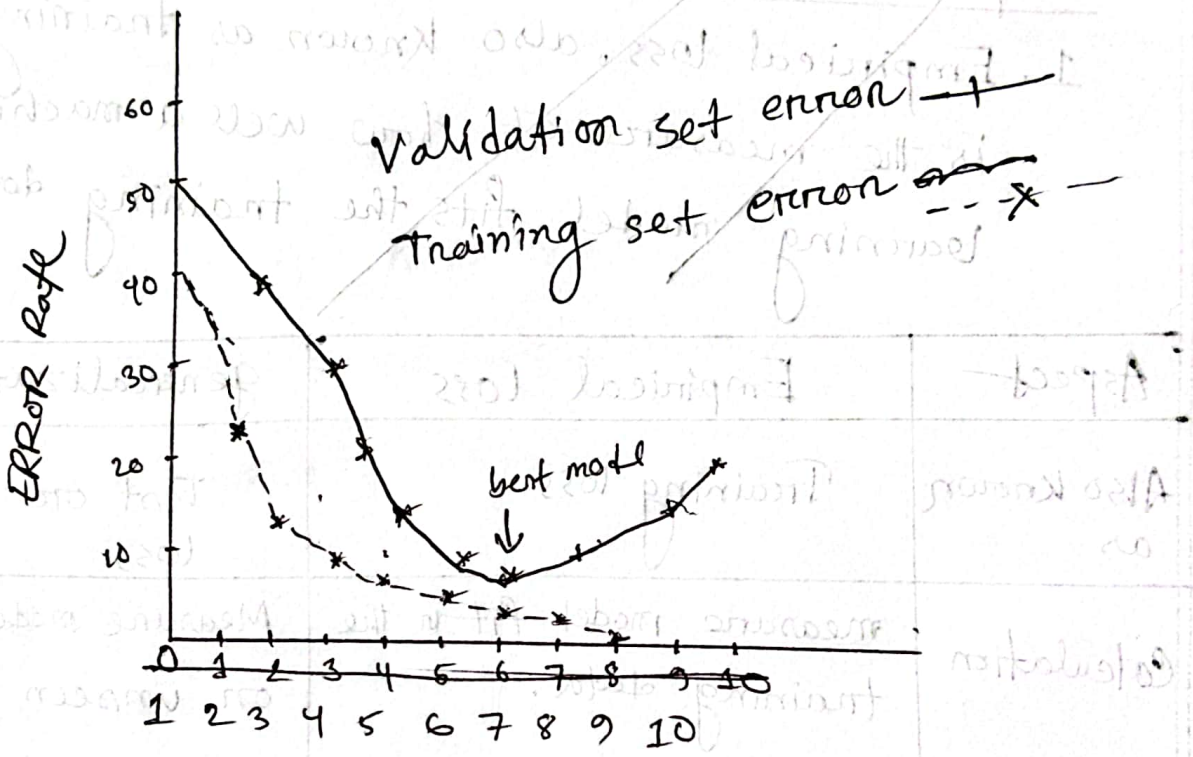
Empirical Loss:

1. Empirical loss, also known as training loss, is the measure of how well a machine learning model fits the training data.

Aspect	Empirical loss	generalization loss
Also known as	Training loss	Test or validation loss
Calculation	measure model fit in the training data.	Measure model performance on unseen data.
Data used	Uses the training data	uses a separate validation on test set
optimization goal	Minimize training data error	Minimize error on unseen data
overfitting risk	does not directly measure overfitting.	Higher values suggest potential overfitting
Goal	To fit the training data well	To generalize to new unseen instance.

Date :...../...../.....

(6) Show the model selection using error rates of training and validation data for different size decision trees.



Error rates on training data (dashed) and validation data (solid line) for different size decision trees. We stop when the training set error rate asymptotes, and choose the tree with minimal error on the validation set, in the tree of size 7 nodes.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :...../...../.....

(7) What is the necessary of the K-fold cross validation technique.?

We can squeeze more out of the data and still get an accurate estimate using a technique called K-fold cross validation.

The idea is that each example serves double duty as training data and test data. First we split the data into K equal subsets. We then perform K rounds of learning, on each round 1/K of the data is held out as a test set and the remaining examples are used as training data. The average test set score of the K rounds should then be a better estimate than a single score.

K=5.;