L#1 **AI** 5-9

→ **What is Intelligence**

→ **What is AI?**

- Human like
- Create machines that can think and act like human
- ML: train the machine to make the decisions
- **Rational - Like**
- Make decisions according to situation
- Human Like decission making policy

- **Reflexive Action**
  - Actions that are performed suddenly on a special condition

e.g: Touching a very hot thing you immediately pull off your hand.

eye blinking

- **History of AI**
  - 1960s → start

- **AI Applications**
  - Robotics

Lecture #3 AI 12-9-2023

→ Agent
↳ Intelligence is measured by
① Situated in a environment
② Autonomous (Independent)
③ Proactive ④ Social (Interactive)

→ Task environment

Types / Structure of agents

1- Simple Reflex Agent → Have no history
2- Model based

Lecture#4 AI 14-4

AI Agent : AI Assistants

- Alexa and Ciri

Problem Solving by Searching

- AI used in games

AI

Problem Solving by Searching

Measuring problem-solving performance

→ Completeness: Always returns the ~~optimal~~ solution (Tell if solution exists or not)

→ Optimality: Optimal Solution

→ Time Complexity: How much time it takes

→ Space Complexity: How much space it takes

Branch factor: Maximum no. of nodes

Search strategies

→ No information about the path or the path cost and do not have additional information.

→ Often called blind search

→ e.g Breadth-first, Depth firsth, Uniform cost, Bidirectional search

Frontier

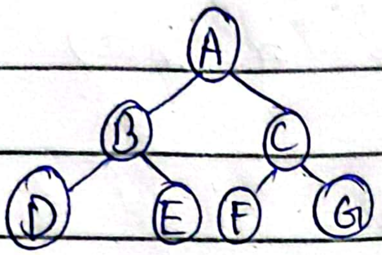
Node that may or may not have child node but they are not expanded or searched.

Breadth - first Search

Implementation:

assumes that every cost is 1

- Fringe is FIFO (Queue)
- Goal = M



Frontier | ~~A~~ | ~~B~~ | ~~C~~ | ~~D~~ | ~~E~~ | ~~F~~ | ~~G~~ |

Expand

Explore: | A | B | C | D | E | F | G |

-> if the frountier is empty

$\boxed{\text{Solution not Found}}$

is A == Goal

Frontier == Goal

BFS:

1. Complete : Yes
2. Time : O(b^d) -> exponential in d
3. Space : O(b^d) -> keeps every no in memory (problem)
4. Optimal : Yes (if cost = 1 per step)

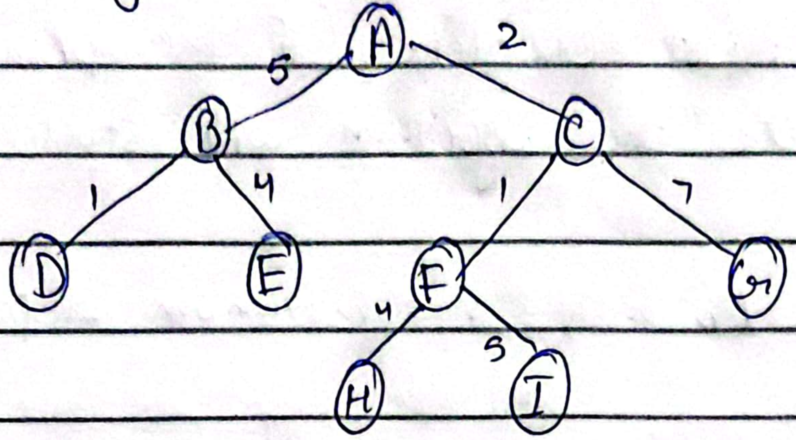
=> Time and Space complexity in exponential is not considered good.

Uniform - cost Search

- If the cost is equal of all nodes then BFS is optimal
- Extention of BFS, depends on costs

Implementation :

- Priority queue
- [X]= g(n)



- Elements are arranged ~~costs~~ according to costs of nodes (Assending order)
- If the cost of 2 nodes are same then we follow BFS (Queue)

UCS: :

① Complete: Yes (if b is finite)
② Time : $O(b^{c^*})$
③ Space: $O(b^{c^*})$
④ Optimal: Yes

Depth First Search

- Always expand one of the deepest level and then back track
- Backtracking search
- Implementation:

44

LIFO (~~Queue~~) (Stack)

DFS

① Complete : Not Complete (if the branch is infinite the algorithm will stuck in loop)

② Time: $O(b^m)$ m = maximum depth

③ Space:

④ Optimal : does't ~~greatee~~ gurentee best solution

AI

26-9

→ DFS is better in space complexity then BFS

Space Compalxity = bxm

→ Depth Limit Search

↳ The Solution ~~of~~ for the depth limit search, fix the limit of the depth.

→ Termination gurenteed

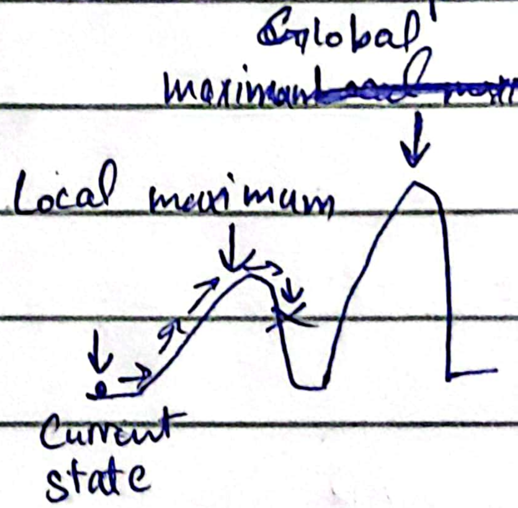
• Local Search

→ We use it because it improve a single
→ search iteratly

→ The goal itself is solution

→ Don't keep track or path but keeps a single "current" state

• Hill climbing algorithm



→ Stuck on local maximum

→ Local maximum get worst in higher dimension

→ Local maximum is ~~not~~ one solution but not optimal

→ Continue in same direction (one direction
forward

→ Plateaus

→ Restart or change direction

→ Random restart hill climbing

→ Meta-heuristic

- Optimization Problem : Reduce the Cost →
  Local minima, Global maxima
- np Hard Problems

⇒ $f(x_2) > f(x_1)$ → directly move

⇒ $f(x_2) < f(x_1)$ → Then use energy fn.

$$\Delta E = f(x_2) - f(x_1)$$

$$p = \frac{\Delta E}{T}$$

T ← assumed number (random)

Adversarial Search

- → Competition
- → Typical AI assumption
- → Zero-sum game (gain of one is lose for other)

Search vs Game

AI

Minimax Algorithm

→ 2 player game and each plays optimally.

→ Best of one is worst of other

Max starts from $-\infty$ $\max(-\infty, 0)$

Min starts from $+\infty$ $\min(+\infty, 0)$

→ Algorithm implementation

Multiplayer games

→ Two-sum cannot be follow

→ Player make alliance

Alpha beta pruning

→ based on min-max but it eliminate exponential space complexity

$$\overset{\alpha}{\left[-\infty\right.}, \left.\overset{\beta}{+\infty}\right]$$

→ Prune whenever $\boxed{\alpha \geq \beta \text{ for max}}$ $\boxed{\alpha \leq \beta \text{ for min}}$

if $\alpha \geq \beta$ or $\overset{3}{\cancel{2}} > 2$ then no need to explore the β branches

→ Alpha-beta is best for pruning trees and it complexity is $(b^{\frac{m}{2}})$ or b ^ (m/2)

→ For worst case it is b^m

AI

17-10

Genetic Algorithm

-> Implement real life genetric operations in AI

-> choose the survival one => better solution

-> Mutation - Swap any random bit

- Evaluation function

AI

19-10

➔ Cross over → there is a chance that The child also have the same properties as parent so mutation is done.

- **Survival of ~~fitness~~ fittest**
  - Select the surviving population
- **Termination**
  - Repeat until a condition is satisfied
  - Fixed number of generation reached
  - Allocation budget

AI

24-1

- ML

1. Supervised Learning
2. Unsupervised Learning
3. Reinforsment Learning
4. Semi - ~~sp~~ supervised Learning

7-11

AI

- Supervised Machine learning
  ↳ Labeled data
  → Data is split in
  ① ~~Test~~ Training data set
  ② Testing dataset

⇒ Sometimes Validation dataset (Held-out)

$$f(y) = \theta_1 x_1 + \theta_0$$

$x \rightarrow$ variable (independant)

$y \rightarrow$ independant

$\theta \rightarrow$ constant

Traditional Programming vs ML

Input, ~~Output~~, Program $\}\rightarrow$ Feed	Input, Output $\}\rightarrow$ Feed
Output → Result	Program → Result

AI

14-11

- Supervised learning

① Classification → when classes are given (Y/N)

② Regression → Number ~~production~~ / Detection

Prediction

Pros

① Previous Knowledge

② Labeled data

Cons

① Cannot predict accurately

② Highly depend on training / testing dataset

③ Can ~~out output~~ stuck on complex equations.

- Navier Bayesian Algorithm

- Only for Classification

- $P(C|F) = (P(F|C) \times P(C)) / P(F)$

AI

16-11

- Naive Baysesian Algorithm

Example #2

$$\frac{P(F/C)\,P(C)}{P(F)} = \frac{P(red/apple)P(round/apple)P(apple)}{P(red)P(round)}$$

$$= \frac{\frac{1}{2} \cdot \frac{2}{2} \cdot \frac{2}{8}}{\frac{4}{8} \cdot \frac{6}{8}}$$

Example #3

~~P(Sunny~~ $P(C) \Rightarrow$

$P(Yes) = \frac{9}{14}$ and $P(No) = \frac{5}{14}$

	Yes	No
Sunny	2/9	3/5
Cool	3/9	1/5
High	3/9	4/5
True	3/9	3/5

$$\frac{P(\frac{Sunny}{Yes})P(\frac{Cool}{Yes})P(\frac{High}{Yes})P(\frac{True}{Yes})P(Y}{P(Sunny)(Cool)(High)(True)}$$

$\cancel{P(\frac{5}{9})(\frac{4}{9})(\frac{9}{9})(\frac{8}{9})}$

~~P(~~

Data Preparation

- Data cleaning
- Feature selection
- Data transformation

→ Big values contribute more in predicture that may lead to wrong prediction

age has range

income has no range

age	income
15	$ 10,000
10	$ 50,000
20	$ 20,0000

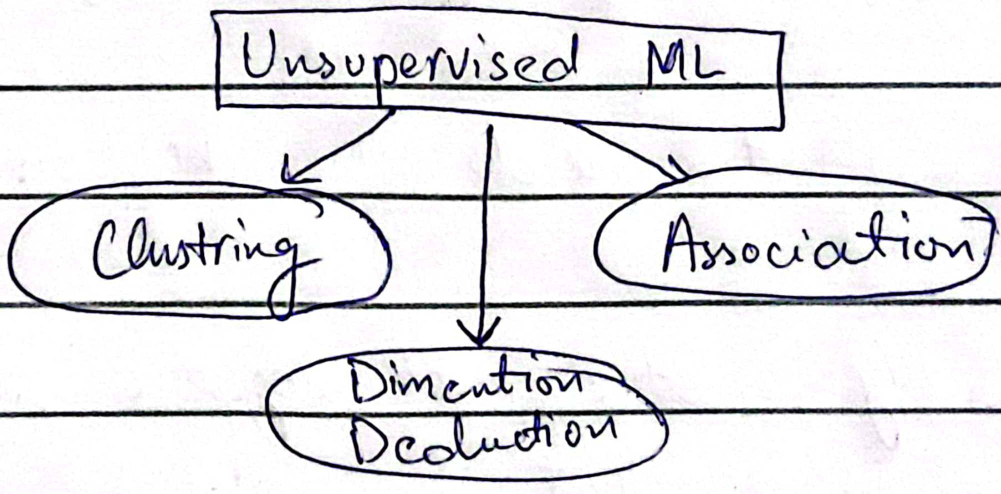
- **Decision Tree**
  - No hard and fast rule.
  - High time and space complexity
  - Not sure about efficiency.

- **Rule Based**
- **SVM** - Support vector machine
- **K - Nearest Neighbour**

AI (Lec # 18) (385) 22-11

Unsupervised Machine Learning

- Unlabeled data
- Clustring on based of ~~simil~~ similar charateristic



→ K-means clustring b/w $a = (x_1, y_1)$ & $b = (x_2, y_2)$

→ Manhattan function

$$P(a,b) = |x_2 - x_1| + |y_2 - y_1|$$

AI

20-11

→ K-means clustering Problem:

→ Elbow Method ⇒ Selecting The actual ~~k-val~~ k-value (optimal number of clustring)

→ distance canot be negative

- Triangle inequality
- KNN is different from k-Means
- Evaluation matrics for supervised and unsupervised machine learning algorithms.
- Confusion matrix
- Recall and Precesion
- F1 score

30-11

AI

- Image Processing
- Grayscale (28X28 default) only one channel
- RGB (28X28X3) 3 channels (Red, Green, Blue)

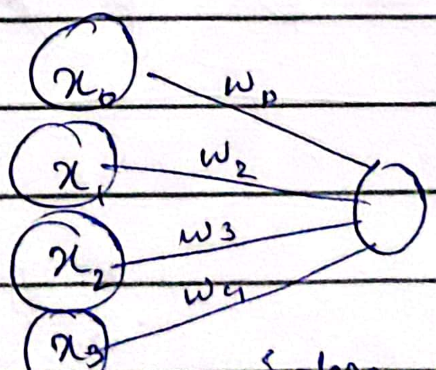
↳ Feature Selection

↳ Neural Networks (NN)

aka ANN or SNN

↳ Input layers, hidden layers, Output layer

$z = x_0w_0 + x_1w_1 + x_2w_2$
$+ x_3w_3 + x_4w_4$



$z \Rightarrow$ Value rang (0-1)

↓

$$g(z) = \frac{1}{1+e^{-z}}$$

$x \rightarrow$ input (never changing in learning) → independent

$w \rightarrow$ weights (changing but NN)

Input layer → Number of inputs
Output layer → Number of classes
→ Loss
$Loss = (Actual - Predicted)^2$ → for one instances
Types of ANN

AI

5-12

weights -> How much the features are important

NN adjusts the weights

$$w_1 = w_1 - \alpha \frac{\Delta \alpha}{w_1}$$

w_1 ↓ old value

$\frac{\Delta \alpha}{w_1}$ -> Difference

-> Knowledge Representation

What to represent

- Meta knowledge = knowledge about knowledge

① Tell ② Ask

- Heuristic knowledge = knowledge from experience

↳ Most of the time it performs good, but not good always.

Logical Representation

Sentence -> Atomic Sentence (True, False)

↳ Complex Sentence

AI

7-12

→ **Knowledge Representation**

↳ **Predicate ~~calculus~~ Logic** ↳ Propositional logic

→ **wumpus world game** – using logical rep.

$\forall$ → always used with ⇒ (with ~~Implies~~ implies)

$\exists$ → used with && (AND)

↳ Representing facts with First-Order Logic

↳ Propositional ~~knowledge~~ ~~calculus~~ logic

↳ Predicate calculus

AI Quiz 2

Unsupervised learning

- **Clustering**: Grouping the objects into clusters
- **Association**: finding the relationship between variables in the large database.

K-means clustering:

1. Choose the value of k to decide no. clusters.
2. Select random K points or centroids.
3. Assign each data point to their closest centroid, which will form the predefined K-
4. Calculate the variance and place a new centroide of each cluster.

- Stop when no points assignment changes

Distance

$$ED = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$ taxicab norm ~~or~~ or 1-norm

$$Mean = \left(\frac{x_1 + x_2 + x_3 + \dots x_n}{n}, \frac{y_1 + y_2 + y_3 + \dots y_n}{y} \right)$$

TP	FP
FN	TN

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{TP}{TP + FP} + \frac{TP}{TP + FN}$$

$$= \frac{2TP}{2TP + FP + FN}$$

$$D = |x_2 - x_1| + |y_2 - y_1|$$

	A1 (2,10)	A4 (5,8)	A7 (1,2)	Class
A1 (2,10)	~~(0,0)~~ 0	~~(3,2)~~ 5	~~(1~~,9	C1
A2 (2,5)	5	6	4	C3
A3 (8,4)	12	7	9	C2
A4 (5,8)	5	0	10	C2
A5 (7,5)	10	5	9	C2
A6 (6,4)	10	5	7	C2
A7 (1,2)	9	10	0	C3
A8 (4,9)	3	2	10	C2
~~A~~				

$WCSS = \sum P_{i \text{ in cluster1}} \text{distance}(P_i C_1$

Mean C1 = (2,10)

Not very consistant due to random initialization.

WCSS → Within cluster sum of square.

$$WCSS = \sum P_i \text{ distance } (P_i C_1)^2 + \sum P + \dots$$

$$WCSS = \sum P_{i \text{ in Cluster1}} \text{ distance } (P_i C_1)^2$$

(i) Symmetric (2) Positivity (3) Triangle inequality

	P(1)	N(0)	
	TP	FP	P(1)
	FN	FN	N(0)

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

ED | MD

$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ | $d = \sqrt{\sum (x_i - y_i)^2}$

$d = |x_2 - x_1| + |y_2 - y_1|$ | $d(x, y) = \sqrt{\sum (x_i - y_i)}$

Winthin Cluster Sum of Squaires

$$WCSS = \sum P_{i \text{ in cluster } 1} \text{distance}(P_i C_1)^2 +$$
$$\sum P_{i \text{ in cluster } 2} \text{distance}(P_i C_2)^2 +$$
$$\sum P_{i \text{ in cluster } 3} \text{distance}(P_i C_3)^2$$

Elbow method

① No consistancy because of random initilization of k-value

② Do not know k-value

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \cancel{w_i x} \text{ Loss}(x, y, \theta)$$

$$\min_{v, m} \text{TrainLoss}(V, w)$$

$$\text{Train Loss}(V, w) = \frac{1}{|D_{train}|} \sum \text{Loss}(x, y, V, w)$$

$$\text{Loss}(x, y, V, w) = (y - f_{v,w}(x))^2$$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Loss}(x, y, \theta)$$

Gradient Descent

$$\theta = \theta - \alpha \Delta_\theta J(x, y, \theta)$$

SGD

$$\theta = \theta = \alpha \Delta\theta \text{ Loss}(x, y, \theta)$$

$$w \leftarrow w - \eta \frac{\partial J(w)}{\partial w}$$

$$w_1 = w_1 - d\left(\frac{\partial L}{\partial w_1}\right)$$

AI

20-12

Prolog

2 major components: facts, rules

:- if

, and

; or

→ Inheritence (is - a) relationship

→ variables always written in capital words (letters) e.g X = John , Y = Mary

→ Query always have a Question mark.

e.g ?- likes (X,Y)

AI

21-12

Robotic System

-> Manual Robot

-> Fixed sequence Robot

-> Variable sequence Robot -> same as 2nd but reprogrammed

-> Playback Robot

-> Numerical Robot

-> Intelligent Robot

-> Law of Robotics

-> Robotic Path Planning

-> Descreet -> Continuous

↓ in memory ↓ Progressively in memory

↳ Dijkstra's Algorithm

↳ A* Algorithm

-> RRT Algorithm

-> Machin Learning Algorithm (Reinforsment lear.)

Naive Bayesian Algorithm → Bayesian Theorem

→ Conditional Probability

↳ Law of chonge . mange uncertainity

→ Assumes that feature of a measurement are . independent of each other

Pros

① No learning model , just extracts probabilitic

② Very easy and simple

③ Sometime provides high accuracy then other Supervised model

Cons

① Assumes That all features are independent

② Do not provide correct output when that features have relationship

① Decision Tree

→ Supervised LA → Tree structured

→ Nodes represents features , branches represents the rules and leaf node represent output

→ xor

② SVM (Support Vector Machine)

- finds a hypeplane to seperate two class of data, positive and negative

$f(x) = \langle w . w \rangle + b$

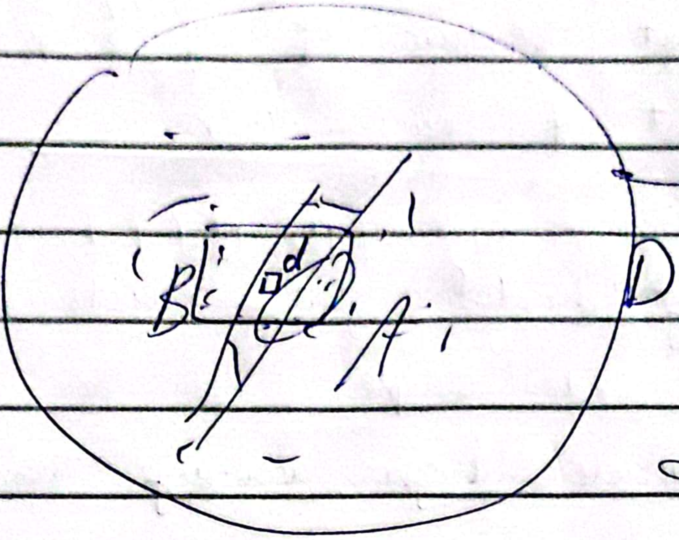
$$y_i = \begin{cases} 1 & \text{if } \langle w . x_i \rangle + b \geq 0 \\ -1 & \text{if } \langle w . x_i \rangle + b < 0 \end{cases}$$

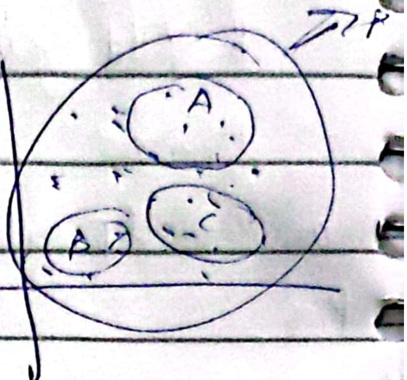
PV

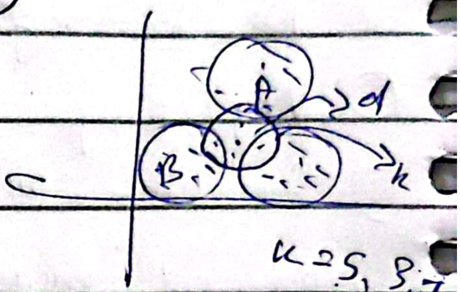
~~TP~~	~~TN~~
~~FP~~	~~FN~~

KNN(D, d, k) AV

① Compute the distance between d and every example in D







k = 5, 3, 7

① Compute the distance between d and every example in D
② Choose the k example in D that are nearest to d, denoted by $P \subseteq D$
③ Assign d the class that is most frequent class in P

F1 Score

$$\frac{2TP}{2TP + FP + FN}$$

	AV	
PV	TP	FP
	FN	TN

Unsupervised → Clustering / Association

① Clustering → Grouping the similar features (objects)
② Association → Find relationship between variable

K-Means

Step 1 - Choose the value of K to decide number of cluster

Step 2 - Select random K points or centroid.

Step 3 - Assign each data point to their closest centroid

Step 4 - Calculate the varience and place a new centroid of each cluster

Step 5 - Repeat the third step

Step 6 - If any reassignment occurs, then go to step 4 else end it.

Euclidian Distance

$$\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

Mean of ~~xs~~, point

$$\left(\frac{x_1+x_2+\ldots x_n}{n}, \frac{y_1+y_2+\ldots y_n}{n}\right)$$

Manhattan Distance

$$P(a,b) = |x_2 - x_1| + |y_2 - y_1|$$

Given Points	(2,10)	(5,8)	(1,2)	Cluster
(2,10)	0	~~7,7~~ 5	9	C1
(2,5)	5	6	4	C3
(8,4)	12	7	9	C2
(5,8)				
(7,5)				
(6,4)				
(1,2)				
(4,9)				

$C_1 = (2,10), ($

$C_2 = (8,4), (5,8), (7,5), (6,4), (4,9)$

$C_3 = (2,5), (1,2)$

$\boxed{C_1 = (2,10)}$

$C_2 = \left(\frac{8+5+7+6+4}{5}, \frac{4+8+5+4+9}{5}\right)$

$\boxed{C_2 = (6,6)}$

$C_3 = \left(\frac{2+1}{2}, \frac{5+2}{2}\right) = \left(\frac{3}{2}, \frac{7}{2}\right)$

$\boxed{C_3 = (1.5, 3.5)}$

Calulate again.

Cons

① Is not very consistant

② We dont always know k

↳ Elbow Method

$$WCSS = \sum_{P_i \text{ in Cluster } 1} distance(P_i C_1)^2 +$$

$$\sum_{P_i \text{ in Cluster } 2} distance(P_i C_2)^2 +$$

$$\sum_{P_i \text{ in cluster } 3} distance(P_i C_3)^2$$

9FIT SIFT

$$WCSS = \sum_{i \text{ in Cluster 1}} \text{distance}(P_i C_1)^2 +$$

$$\sum_{i \text{ in cluster 2}} \text{distance}(P_i C_2)^2 +$$
$$\sum_{i \text{ in Cluster 3}} \text{distance}(P_i C_3)^2 +$$
$$\sum_{i \text{ in Cluster 4}} \text{distance}(P_i C_4)^2$$

- Symmetric $D(A,B) = D(B,A)$
- Positivity and Self-Similarity
  $D(A,B) \geq 0$ and $A(A,B) = 0$ if $A = B$
- Triangular inequality
  $D(A,B) + D(B,C) \geq D(A,C)$

ANN or SNN

→ has a association weight, If the output of any individual node is above the specified threshold value, that node is activated, sending data to next layer of the network. → Feedforward

$$g(z) = \frac{1}{1+e^{-z}} \Rightarrow h(x) = g(\theta^T x)$$

$$\text{output} = f(x) = \begin{cases} 1 & \text{if } \sum w_1 x_1 + b \geq 0 \\ 0 & \text{if } \sum w_1 x_1 + b < 0 \end{cases}$$

$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + \ldots w_n x_n + bias$$

$$score = f(v, w)$$

$$\text{Train Loss}(V, w) = \frac{1}{|D_{train}|} \cancel{\Sigma(\text{Loss}}$$

$$\text{Train Loss}(V, w) = \frac{1}{|D_{train}|} \Sigma \text{Loss}(x, y, V, w)$$

$$\text{Loss}(x, y, V, w) = (y - f_{V,w}(x))^2$$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Loss}(x, y, \theta)$$

Gradient Decent

$$\theta = \theta - \alpha \nabla_\theta \underline{J(x, y\theta)}$$

$$\underline{\theta = \theta - \alpha \nabla_\theta \text{Loss}(x, y, \theta)}$$

$$\theta = \theta - \alpha \nabla_\theta \text{Loss}(x, y, \theta)$$

$$\theta = \theta - \alpha \nabla_\theta J(x, y, \theta)$$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Loss}(x, y, \theta)$$

$$\text{Loss}(x, y, \theta) = (y - f_{V,w}(x))^2$$

$$\text{Train Loss} = \frac{1}{|D_{train}|} \Sigma \text{Loss}(x, y, \cancel{\theta}) V, w)$$

Types of knowledge
1. Procedural
2. Declarative
3. Meta - Knowledge
4. Heuristic Knowledge
5. Structural

Robots Types
1. Fixed Sequence
2. Variable Sequence
3. Manual
4. Numerical
5.
6. Intelligent

Logical Representation (Propositional knowledge)

and	$\wedge$	True if both True (P, Q)
or	$\vee$	True if any one is True
not	$\neg$	True if False, False is True
if	$\Rightarrow$	True if unless P is true and Q is F
iff	$\Leftrightarrow$	True if both T or both F

Syntax
- construct ~~logic~~ legal sentence in logic
- which symbol we use in knowledge rep.
- Contruct legal sentence in logic

Semantic
- Rules by which we can interpret the sentence in the logic

Syntax
- Rule how to construct legal sentence in logic

Semantic
- Rule by which we interpret the sentence in the logic

Types
① Predicate Logic → Have Relationship
② Propositional Logic → No relationship → True or False
① simple ② Complex

Propositional Logic

$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow -\alpha) \equiv (\neg\alpha \vee \beta)$

$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$

$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$

→ Predicate calculus e.g $b(S, T)$

S, T are associated / connected by relationship b

Predicate calculus have Quantifiers

$\forall$ → For all + $(\Rightarrow)$

$\exists$ → There exists $(\wedge)$

$\forall x$ At (x, Lahore) $\Rightarrow$ Dumb

$\exists x$ At (x, Ksh) $\wedge$ Numb

Some at Standford is smart

$\exists x$ At (x, Standford) $\wedge$ Smart(x)

All at Standford are smart

$\forall x$ At (x, Standford) $\Rightarrow$ Smart(x)

$\forall x \forall y = \forall y \forall x$ and $\exists x \exists y = \exists y \exists x$

$\exists x \forall y \neq \forall y \exists x$

$\forall x$ Likes (x, Ice Cream) $\equiv \neg\exists x \neg$ Likes (x, Ice cream)

$\forall x, y \; Mother(x,y) \Rightarrow (Female(x) \land Parent(x,y))$

$\forall x, y \; FirstCousin(x,y) \Rightarrow \exists p, ps \; Parent(p, x)$
$\land \; sibling(ps, p) \land Parent(p\overset{s}{\cancel{x}}, y)$

$\forall x, y \; FirsCousin(x,y) \Rightarrow \exists p, ps \; Parent(p, x) \land Sibling(p, p)$
$parent(px, y)$

$d \rightarrow$ is a day

$p \rightarrow$ is a person

$mo \rightarrow$ is mugged on

$mi \rightarrow$ is mugged in

$s \rightarrow$ Soho $y \rightarrow$ unspecified person

$x \rightarrow$ unspecified day

$\forall x((dx) \rightarrow \exists x(mo(y,x) \land py) \land mi(y, s)))$

$\forall x((dx) \rightarrow \exists x(p(y) \land (mo(y,x) \land mi(y,s)))$

- ~~is Person (Ali)~~
- isProf (Ali)
- $\forall x(isProf(x) \rightarrow isPerson(x))$
- isDean (Salman)
- $\forall x(isDean(x) \rightarrow isProf(x))$

$\forall x \forall y(isProf(x) \land isDean(y) \rightarrow friendof(x,y)$
$\lor \rightarrow knows(x,y))$

- $\forall x \, \exists y \; friendOf(x, y)$
- $\forall x \, \forall y \; (isPerson(x) \wedge isPerson(y) \wedge$
  $criticise(x, y) \rightarrow \neg friendOf(x, y))$
- $\neg friendOf(Salman, Ali)$