MONISH KUMAR . E.M

2023115076

IT-1

ML ASSIGNMENT-II

Unit - V Advanced learning

1) K-means clustering:-

K-means is a partitional clustering algorithm that divides a dataset into K-clusters based on similarity

Steps:-

1) choose no of clusters K
2) Randomly initialize K centroids
3) Assign each data point to nearest centroid
4) recompute centroids as mean of points in each clusters.
5) Repeat steps 3-4 until centroids do not change significantly.

Objective : Minimize $J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$

where, C_i = cluster i

μ_i = centroid of cluster i

Application:-

=) customer segmentation

=) Image compression.

2) principal component Analysis (PCA)

PCA is a dimensionality reduction technique used to transform high-dimensional data into fewer dimension while preserving maximum variance

steps:

1) standardise the data
2) compute covariance matrix
3) find eigenvalue, eigenvector
4) select top k eigenvector (principal components)
5) Transform data into new reduced feature space.

example:-

Dataset with 2 feature

x_1	x_2
2	0
0	2
3	1
1	3

step 1: standardise (sub mean = 1.5)

step 2: covariance matrix

$$\Sigma = \begin{bmatrix} 1.67 & -1.0 \\ -1.0 & 1.67 \end{bmatrix}$$

first principal component aligns along direction [1, -1];

data can be represented along one axis $(x_1, -x_2)$ capturing most of variance.

Advantages:-

=> Removes noise & redundancy.
=> reduces computation
=> Improves visualisation

Applications:-

=> Face and handwriting recognition
=> Image compression
=> Exploratory data analysis

3) Gaussian mixture model (GMM)

Introduction:-

A Gaussian mixture model (GMM) is a probablistic model for representing normally distributed subpopulation within an overall population unlike K-means, it allows clusters of different shapes.

model defination:-

$$P(X) = \sum_{i=1}^{K} \pi_i . N(X|\mu_i, \Sigma_i)$$

* π_i = mixing coefficient
* μ_i = mean vector
* Σ_i = covariance matrix

Learning parameter:-

expectation - maximization (EM) algo:-

1) Initialize means (μ), covariance (Σ) and weight (π).

2) E-step : compute probability each point belongs to each gaussian.

3) M-step : Update μ, Σ, π based on these probabilities

4) Repeat : Until log-likelihood converges.

examples:-

consider 1D data : [1.0, 1.2, 1.4, 5.0, 5.2, 5.4]

we assume 2 gaussians

* Initial means $\mu_1 = 1.0, \mu_2 = 5.0$
* EM will adjust $\mu_1 = 1.2, \mu_2 = 5.2$ and computes
* result: two overlapping normal curves fit two clusters better than k-means circles.

Advantages:-

=> handles overlapping clusters
=> provides probability of membership

Application:-

=> speaker recognition
=> Object detection.
=> Anomaly detection.

4. Q - learning Algorithm:-

reinforcement learning (RL) is learning through interaction. An agent learns to make sequence of decisions.

by receiving rewards from the environment. Q-learning is model - free off policy RL algorithm that learn optimal action - value function.

components:-

=) state: environments situation

=) Action: what agent can do

=) reward: numerical feedback

=) policy : mapping from state to actions

=) Q - value: expected feature rewards for (S, a)

Q - learning update equation.

$$Q(S,a) \leftarrow Q(S,a) + \alpha[r + \gamma \cdot \max_{a'} \cdot Q(S', a') - Q(S,a)]$$

example:-

grid world:

Agent starts at bottom - left and must reach top - right goal

Action: [up, down, left, right]

reward: +10 for goal, -1 per step.

Initially $Q(S,a) = 0$

when agent moves and gets rewards r, Q-table updates using formula.

After many episodes, agent learns optimal actions

Advantages:-

=> learns without environment model

=> works for stochastic tasks.

Application:-

=> Game AI (chess, Go)

=> robotic navigation

=> Traffic signal optimization.

Problem based on K-means clustering:-

cluster these points into $k=2$ clusters.

(2,3), (3,3), (6,6), (8,7)

step1: Initialize centroids.

$C1 = (2,3)$, $C2 = (6,6)$

step2: Assign points

point	d(C1)	d(C2)	assigned.
(2,3)	0	5	C1
(3,3)	1	4.24	C1
(6,6)	4.24	0	C2
(8,7)	6.4	2.24	C2

cluster: $C1 = [(2,3), (3,3)]$

$C2 = [(6,6), (8,7)]$

step 3: recompute centroids

$C1 = (2.5, 3.0)$

$C2 = (7.0, 6.5)$

step 4: reassign points → same cluster

$C1 = [(2,3), (3,3)]$

$C2 = [(6,6), (8,7)]$