UNIT V - DISTANCE AND RULE BASED MODELS

Distance Metrics - measures similarity / dissimilarity bet 2/more data points.

Types of Distance Metrices.

① **Euclidean Distance** - straight line distance between 2 points in Euclidean plane. It is calculated using Pythagorus thm.

$P = (x_1, x_2, \ldots, x_n)$

$Q = (y_1, y_2, \ldots y_n)$

$$D(P,Q) = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + \cdots + (x_n-y_n)^2}$$

But, normally for $A(x_1, y_1)$ & $B(x_2, y_2)$

$$D(A,B) = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

Appln
KNN
Clustering methods

② **Manhattan Distance** - also known as "taxicab" or "city block" distance. This measures the distance betn 2 points based on sum of absolute differences of their coordinates.

For $A(x_1, y_1)$ & $B(x_2, y_2)$

$$D(A,B) = |x_2 - x_1| + |y_2 - y_1|$$

→ similar to Euclidean but emphasis grid like paths

→ useful for high-dimensional spaces

③ **Hamming Distance** - measures no. of positions at which 2 strings of equal length differ. Primarily used for categorical data

For 2 strings A & B:

$$D(A,B) = \sum_{i=1}^{n} [A_i \neq B_i]$$

→ applicable only to strings of equal length

→ non-negative & symmetric

Appln
error detection / correction

④ **Minkowski Distance** - generalizes both Euclidean & Manhattan distances. It is defined by a parameter 'p', which determines the type of distance being calculated.

for 2 points P & Q:

$$D(P,Q) = \left(\sum |x_i - y_i|^p\right)^{1/p}$$

$p = 1$, it becomes Manhattan distance

$p = 2$, it becomes Euclidean distance

$p = \infty$, it approaches Chebyshev distance

Neighbours & Examples

↳ refers to data points that are closest to a given point based on specific distance metrics

KNN for Classification & Regression

KNN is ML algo. used to make predictions. It looks at the closest data points (neighbours) to a new data point & makes decisions based on those neighbours.

For Classification.

- Select optimal value of k - not high not low
- Calculate distances - all distances by metrics.
- Identify nearest neighbour - determine smallest distance
- Voting mechanism - assign class label that has max votes

For Regression
Select k
Calculate distances
Identify nearest neighbours
Average values = calculate predicted value by taking avg of target values of neighbours.

Applications

For classification
- Image ~~classificat~~ recognition
- document classification
- medical diagnosis

For Regression
- Price prediction
- forecasting.

Advantages
- simplicity
- Non-parametric ie no assumptions
- Versatile - both Regr^n & Class$^{f^n}$

Disadvantages
- Requires lot of calculations
- sensitive to irrelevant features
- Curse of dimensionality.

Clustering as a Learning Task
Clustering is an unsupervised learning task that involves grouping similar data points into ~~clust~~ clusters based on their distance from one another.
Algorithms used for clustering - K-means, K-medoids & heirarchical clustering.

K-means clustering algorithm.
- is a way to group similar data points together into clusters.

Steps :-
Step 1 - Choose the number of Clusters (K)
Step 2 - Pick initial centers
Randomly select k points from your data as the starting centers (called centroids) for each cluster.
Step 3 - Assign Data points to clusters
For each data point, find the closest centroid using Euclidean distance
Assign the data point to the cluster, with nearest centroid
Step 4 - Update centroids.
Find avg, and this becomes new centroid for the cluster.
Step 5 - Repeat (step 3 - 4) until
No points change their cluster assignment.
The centroids dont move much anymore
You reach max. iterations.

K-Medoids with Example

Step 1 - Initialize: 'K' random points as initial medoids.
Step 2 - Assignment: Assign each data point to the cluster of the nearest medoid based on Manhattan distance.
Step 3 - Update: find the medoid that min. the total distance to all other points in that cluster.
Step 4 - Repeat. Swapping of non-medoids with medoid can be done if ≤ cost

Centroid (K-means)

→ avg position of all points in a cluster.
→ Calculated by taking the mean of the coordinates of all points within the cluster.

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

→ It represents center of the ~~centroid~~ cluster.
→ Eg (1,2) (3,4) (5,6) then

$$C = \left(\frac{1+3+5}{3}, \frac{2+4+6}{3}\right) = (3,4)$$

Medoid (K-Medoids)

→ centrally located point in a cluster
→ medoid is always one of the actual data points in dataset.
→ calculate total distance of each point to all other point & select the point with lowest distance.
→ medoid minimizes sum of distance to all other points in its cluster
→ more robust to outliers
→ Eg. (1,2) (3,4) (5,6)

for (1,2) = d(3,4) + d(5,6)
for (3,4) = d(1,2) + d(5,6)
for (5,6) = d(1,2) + d(3,4)

smallest total distance is medoid

Hierarchical Clustering

Group similar items into clusters based on their characteristics.
It creates a tree-like structure called dendrogram.

Types:

Agglomerative

→ most common type
→ start with each item as its own cluster.
→ clusters are merged iteratively based on approx. until single cluster remains / desired no. of clusters is formed.

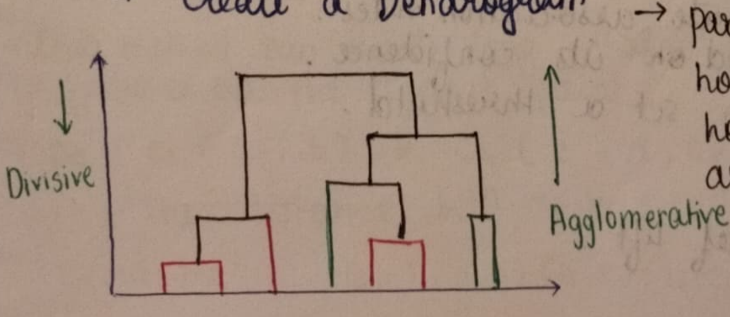
Divisive

→ starts with all items in one cluster.
→ It then splits the cluster into smaller ones until each item is its own cluster.
→ less commonly used method.

Divisive Dendrogram for Heirarchical clustering

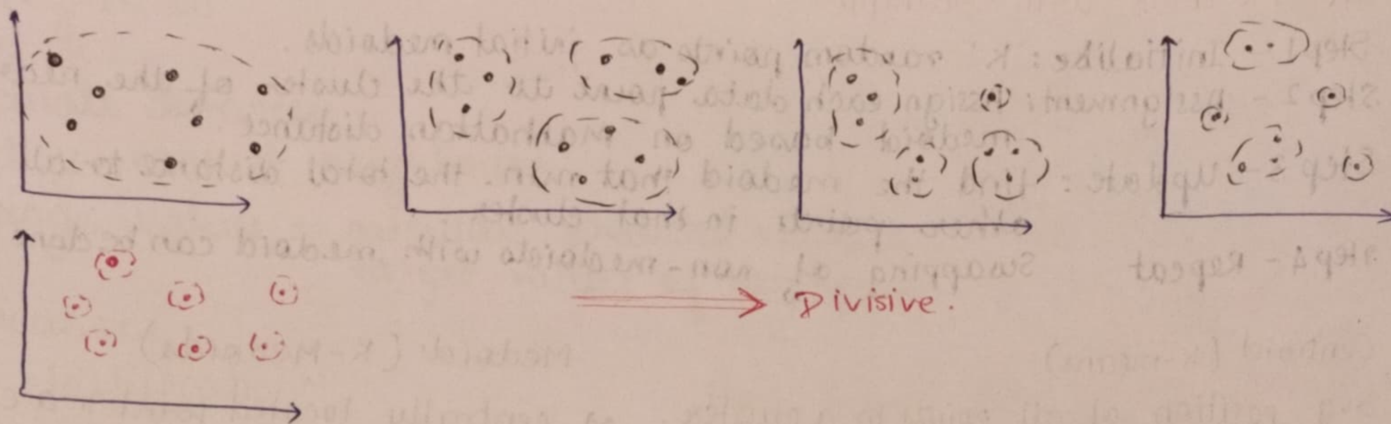
top-down approach.

Step 1 - Start with one cluster
2 - Split the cluster
3 - Recursive Splitting
4 - Create a Dendrogram



Dendrogram

→ Tree like diagram that visually represents the heirarchical relationships betn obj
→ particularly used for illustrating how clusters are formed & how similar / dissimilar are the objects are from each other.



Association Rule Mining

Aims to observe frequently occuring patterns, correlations or associations from dataset.

Association Rules :—

Typically expressed in the form of "if-then" statements, where "if" is called antecedent & "then" is called consequent.

eg. "If a customer buys bread, then they are likely to buy butter".

Rule learning for subgroup discovery.

Involves identifying subsets of data that exhibit specific characteristics or behaviour.

This helps in making predictions based on observed data.

Subgroup discovery aims to find rules that characterize certain groups within the dataset, providing insights into relationships & trends.

Apriori Algorithm

most widely used for mining frequent itemsets & generating association rules.

Step 1 - Determine Support.

calculate support for all itemsets & select the minimum support.

Step 2 - select frequent Itemsets.

select itemsets that has support value > min. support.

Step 3 - Generate Rules.

for selected freq. itemsets, generate association rules.

Each rule is evaluated based on its confidence.

Select minimum confidence to set a threshold.

step 4 - Sort Rules by lift

sort in decreasing order of lift.

Performance Measures

① Support - proportion of transaction that contain specific itemsets ie measures the frequency of occurence of itemset in a dataset

High support = more significant freq in dataset.

$$\text{support}(X) = \frac{\text{No. of transactions containing } X}{\text{Total no. of Transactions}}$$

② Confidence - the likelihood that the consequent occurs given that the antecedent is present.

High confidence = stronger rules
Ranges between 0 to 1

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

③ Lift - how much likely the consequent occurs when the antecedent is present compared to when they are independent.
ie measures the strength of a rule compared to the expected frequency of the consequent occuring independently of the antecedent.

Value > 1 = positive correlation betⁿ X & Y.

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

④ Rule - An implication of the form $X \Rightarrow Y$, where X(antecedent) is set of items & Y (consequent) is another set of items. This means if X occurs, then Y is likely to occur as well.

Q) State and explain with appropriate example different types of linkage use in clustering.

In agglomerative hierarchical approach, we start by defining each data point to be a cluster and combine existing cluster at each step. Here are different methods to do it:

① Single Linkage -
→ measures distance between 2 clusters as the minimum distance betⁿ any single pair of points from each cluster.
→ This method can create long, chain like clusters because it focuses on the closest point.

$$L(A,B) = \min d(a,b)$$

→ Eg $C_1(a, b)$ & $C_2(c, d, e)$
∴ The distance betⁿ cluster 1 & 2 would be smallest distance found betⁿ (a,c) (a,d) (a,e) (b,c) (b,d) (b,e)

② Complete Linkage :

→ distance betⁿ 2 clusters as the maximum distance betⁿ any single pair of points from each cluster.

→ tends to produce more compact & spherical clusters since it considers farthest points.

$$L(A,B) = \max d(a,b)$$

③ Average Linkage :

→ calculates distance as avg of all pairwise distances betⁿ points in both clusters.

→ provides balance betⁿ single & complete linkage.

$$L(A,B) = \frac{1}{|A| \cdot |B|} \sum_{a \in A} \sum_{b \in B} d(a,b)$$

no. of points in clusters A & B

④ Centroid Linkage

→ measures distance based on distance betⁿ their centroids

→ this method can also create spherical clusters but may be influenced by outliers

$$L(A,B) = d(\mu_A, \mu_B)$$

centroids of CA & CB

⑤ Ward's Method

→ minimizes total within-cluster variance

→ produce compact & well separated outliers