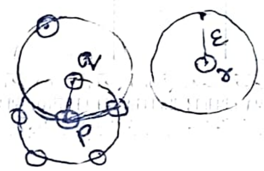
1) a)

i) Core object :- With respect to some radius Eps and Minimum no. of points req. MinPts ; if an ~~core object~~ object has more than MinPts objects in its $\varepsilon$ - neighborhood, then it is said to be a core object

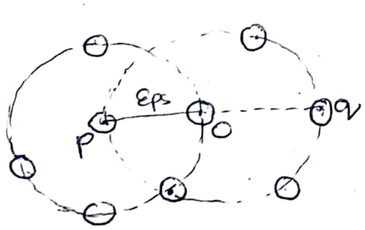
Border object :- wrt Eps & MinPts, an object that has less than MinPts objects in its $\varepsilon$-neighborhood, but itself lies in the $\varepsilon$-neighborhood of a core object is termed to be a border object

Noise object :- An object that is neither a core object nor a border object is said to be a noise object ; denoting errors or outliers in recording.




For MinPts = 5 and Eps = $\varepsilon$ ;
p is a core object
q is a border object
r is a noise object.

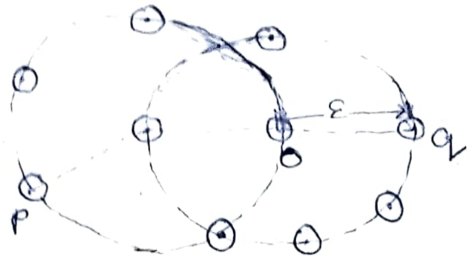
ii) Density reachability :- An object q is said to be density reachable from point p if there lies a chain of objects $p_1, p_2, \ldots, p_n$ such that $p_1 = p$ = a core object ; $p_i$ is directly density reachable from $p_{i-1}$ i.e $p_1, p_2, \ldots p_{n-1}$ are all core objects ; and $p_n = q$ w.r.t to some radius distance Eps & minimum no. of points MinPts




Here q is density reachable from p through o.

Density connectivity :- Two points p and q are density connected if they are density reachable to some common point o w.r.t to radius distance Eps & Minimum Number of Points MinPts

(5)

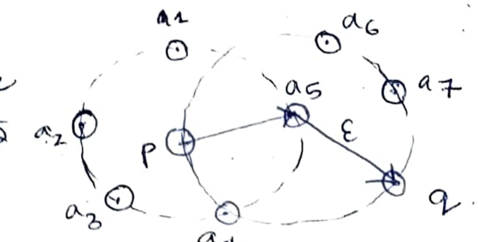



Here, for Eps = ε and MinPts = 5 ; p is density connected to q through o as both p & q are density reachable to o.

b) i) Maximality condition :- for some point p and cluster C such that p ∈ C, point q ∈ C if q is density reachable to p through some chain of core objects. q itself may be a core object or border object.

(2½)

Here Eps = ε
MinPts = 5

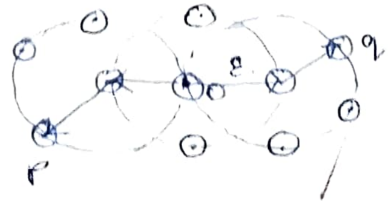



Here q is density reachable to p hence lies in the same cluster as that same cluster as p.

Also $a_1, a_2, a_3, a_4, a_5, a_6, a_7$ all lie in

This is wrt. some radius distance Eps & Minimum number of points MinPts.

ii) connectivity condition :- for two points p and q ; some radius distance Eps, and Minimum Number of Points MinPts, ~~p and~~ and cluster C, ~~p and~~ ~~p q~~ q ∈ C if p is density connected to q through some point o from which p and q are density reachable, and p, o ∈ C




Here, if p and o ∈ C, q ∈ C as q is density connected to p through o for Eps = ε & MinPts = 4.

c) ~~EP~~ · Eps can be chosen very easily for some value of MinPts using a graphical algorithm.

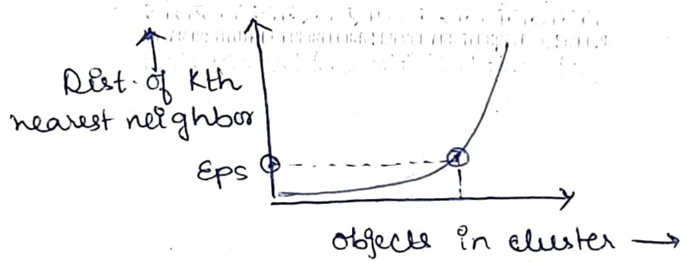
→ We use the concept of k-nearest neighbor, where, for each object we calculate the k nearest neighbors distance from it.

→ we then sort the points according to their k-dist (k-dist = distance of kth nearest neighbor)

→ for core objects and border points, this distance is fairly similar, based on the fact that for objects in a cluster, their kth nearest neighbors are approximately at the same distance. For noise points this value is fairly high

→ On plotting we may get a graph like:

4




→ we consider the point where the distance changes abruptly as the value of Eps for the chosen value of k.

Accordingly, points with dist of kth nearest neighbor < Eps are core objects; and rest are labelled as border or noise objects.

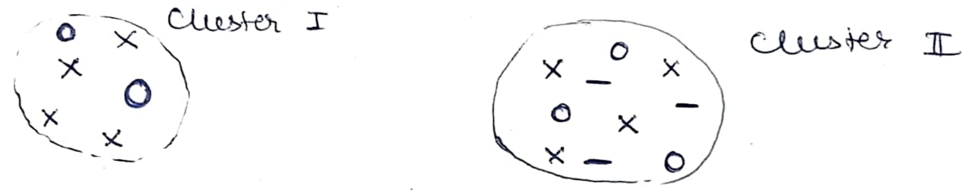
And how do we select k? k should not be

→ Too large as then small clusters with dist. of kth nearest neighbor <<< Eps will be labelled as noise.

→ Too small as then noise points or outliers will also be labelled as core or border points in a cluster.

2) i) Purity :- Purity of a cluster is defined as ratio of the frequency of items in most dominant class of a cluster to the total no. of items in the cluster. It is biased because ~~do~~ using this formula, clusters with only one object would have maximum purity of 1.

eg.




for cluster I ; $\text{purity} = \dfrac{\max(4, 2)}{4+2} = \dfrac{4}{6} = 2/3$

3

for cluster II ; $\text{purity} = \dfrac{\max(4, 3, 3)}{4+3+3} = \dfrac{4}{10} = 2/5$

ii) Rand Index :- It is ~~at~~ the ratio of items correctly classified to the total sum of all the cells in the confusion matrix.

| | Same class by classifier | Different classes by classifier |
|---|---|---|
| Same class in ground truth | A 50 | B 60 |
| Different classes in ground truth | C 60 | D 40 |

Rand index of this distribution is :- $\dfrac{A+D}{A+B+C+D}$

It is similar to measuring accuracy of a classification.

$= \dfrac{50+40}{50+40+60+60}$

$= \dfrac{90}{90+120} = \dfrac{90}{210}$

$= 3/7$

Both Rand index & Purity are external ~~vao~~ validation measures for measuring goodness of some clustering ~~classification~~ algorithm.

b) Intercluster distance $\delta(S, T)$ is defined as the distance of ~~ob~~ between objects $x$ and $y$ such that $x \in S$ and $y \in T$. High intercluster distance means high separation value and is indicative of good clustering.

Two intercluster distances are :-

i) Single linkage distance :- Is the distance between two nearest objects belonging to two different clusters.

Mathematically; $\delta_1(S,T) = \min_{\substack{x \in S \\ y \in T}} \{\delta(x,y)\}$

ii) Average linkage distance :- Is the average of the distance between all pairs of objects belonging to two different clusters. Mathematically; $\delta_2(S,T) = \frac{1}{|S| \times |T|} \sum_{\substack{x \in S \\ y \in T}} \{\delta(x,y)\}$

where $|S|$ and $|T|$ denote total number of objects in clusters S and T respectively.

5

Intracluster distance $\Delta(S)$ is defined as the distance between objects $x_1$ and $x_2$ such that $x_1 \in S$ and $x_2 \in S$. Low intracluster distance means high compactness and tightness and is indicative of a good clustering.

Two intracluster distances are :-

i) Complete diameter distance :- It is the distance between two farthest or most remote objects belonging to the same cluster. Mathematically; $\Delta_1(S) = \max_{x, y \in S} \{d(x,y)\}$

ii) Average diameter distance :- It is the average of distances between pairs of any two objects lying in the same cluster. Mathematically, $\Delta_2(S) = \frac{1}{|S| \times |S-1|} \sum_{\substack{x, y \in S \\ \forall x \neq y}} \{\delta(x, y)\}$

where $|S|$ denotes number of objects in cluster $S$.

Intercluster & Intracluster distance are two ~~interna~~ internal measures for ascertaining goodness of clusters obtained from some clustering algorithm.

c) Dunn's cluster validation index or DIndex is a way of validating where a clustering is good enough or not. Mathematically it is defined as:-

4

$$D\,Index(U) = \min_{i=1 \text{ to } c} \left\{ \min_{\substack{j=1 \text{ to } c \\ j \neq i}} \left\{ \frac{\delta(X_i, X_j)}{\max_{k=1 \text{ to } c} \{\Delta(X_k)\}} \right\} \right\}$$

where $c$ is the number of clusters in clustering $U$; $\delta(X_i, X_j)$ denotes intercluster distance between clusters $i$ and $j$; and $\Delta(X_k)$ denotes intracluster distance of cluster $k$.

Usefulness in cluster evaluation :-

→ Here we are trying to maximize the intercluster distances and minimize the intracluster distances.

→ Accordingly, a ~~to~~ high value of DIndex indicates a good clustering; and the clustering/number of clusters that lead to the biggest value of DIndex is chosen as the optimal clustering.

4) a) Here, the class variable is PlayTennis for which have two values P = Yes N = No.

Accordingly $I(p,n)$ or ~~Im~~ Information needed to classify an example = ~~$\log \frac{p}{p}$~~

$$-\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Here $p=6$; $n=4$; $p+n=10$.

$$\therefore I(p,n) = -\frac{6}{10}\log_2\frac{6}{10} - \frac{4}{10}\log_2\frac{4}{10}$$

$$= -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$$

$$= -\frac{3}{5}(\log_2 3 - \log_2 5) - \frac{2}{5}(\log_2 2 - \log_2 5)$$

$$= \frac{3}{5}(\log_2 5 - \log_2 3) + \frac{2}{5}(\log_2 5 - \log_2 2)$$

$$= \frac{3}{5}(2.3 - 1.6) + \frac{2}{5}(2.3 - 1)$$

$$= \frac{3}{5}\times 0.7 + \frac{2}{5}\times 1.3 = \frac{2.1 + 2.6}{5}$$

$$= \frac{4.7}{5} = 0.94$$

Let some attribute A divide data into subsets $S_1, S_2, \ldots S_n$

entropy of ~~some~~ A. $\text{entropy}(A) = \sum_{i=1}^{n} \frac{p_i + n_i}{p+n} I(p_i, n_i)$

where $p_i$; $n_i$ = no. of examples in subset $i$ classified as P and N respectively

if A = outlook; A divides data into 3 subsets → Sunny, → overcast, → Rain

for $i=1$; outlook = Sunny;
$p_i = 2$ $n_i = 2$

for $i=2$; outlook = Rain;
$p_i = 2$ $n_i = 2$

for $i=3$; outlook = overcast;
$p_i = 2$, $n_i = 0$.

$$I(2,0) = -\frac{2}{2}\log\frac{2}{2} - 0 = -2/2 \cdot 0 = 0.$$

$\therefore$ entropy (outlook) =

$$\frac{4}{10} I(2,2) + \frac{4}{10} I(2,2) + \frac{2}{10} I(2,0)$$

$$I(2,2) = \frac{2}{4}\log_2\frac{4}{2} + \frac{2}{4}\log_2 \ldots$$

$$= \frac{1}{2}\log_2 2 + \frac{1}{2}\log_2 2$$

$$= \log_2 2 = 1.$$

$\therefore$ entropy (outlook) $= \frac{4}{10} \cdot 1 + \frac{4}{10} \cdot 1 + \frac{2}{10} \cdot 0$

$= \frac{2}{5} + \frac{2}{5} = \frac{4}{5} = 0.8$

for A = Humidity; A divides data into 2 subsets $\nearrow$ Normal, $\rightarrow$ High

for i = 1; Humidity = Normal

$p_i = 4 \quad n_i = 1$

for i = 2; Humidity = High

$p_i = 2 \quad n_i = 3$

$\therefore$ entropy (Humidity)

$= \frac{5}{10} I(4,1) + \frac{5}{10} I(2,3)$

$I(4,1) = \cancel{\log} \ \frac{4}{5} \log 5/4 + \frac{1}{5} \log 5/1$

$= \frac{4}{5} \log_2 \frac{5}{4} - \frac{1}{5} \log_2 \frac{1}{5}$

$= \frac{4}{5}(\log_2 5 - \log_2 4) - \frac{1}{5}(\log_2 1 - \log_2 5)$

$= \frac{4}{5}(2.3 - 2) + \frac{1}{5}(\log_2 5 - 0)$

$= \frac{4}{5} \times 0.3 + \frac{1}{5} \times 2.3 = \frac{1.2 + 2.3}{5} = \frac{3.5}{5} = 0.7$

$I(2,3)$

$= \frac{2}{5}(\log 5 - \log 2) + \frac{3}{5}(\log 5 - \log 3)$

$= \frac{2}{5} \times 1.3 + \frac{3}{5} \times 0.7$

$= \frac{2.6 + 2.1}{5}$

$= 0.94$

$\therefore$ entropy(Humidity) $= \frac{1}{2} \times 0.7 + \frac{1}{2} \times 0.94 = \frac{1.64}{2} = 0.82$

for A = wind; A divides data into 2 subsets $\nearrow$ strong, $\rightarrow$ weak.

for i = 1; wind = strong

$p_i = 2 \quad n_i = 3$

for i = 2; wind = weak.

$p_i = 4 \quad n_i = 1$

entropy (wind) $= \frac{5}{10} I(2,3) + \frac{5}{10} (I, 4, 1)$

which is same as entropy (Humidity) = 0.82

$\therefore$ gain(outlook) = 0.94 − 0.8 = 0.14

gain (Humidity) = gain (wind) = 0.94 − 0.82 = 0.12

$\because$ gain (outlook) is maximum, so we choose attribute outlook as root of decision tree. (Ans)

5

b) The possible terminating criteria of a decision tree algorithm are :-

i) There are no more samples left to classify

ii) All the remaining samples belong to the same class so we ~~can~~ don't need to partition anymore.

3

iii) There are no more attributes remaining for making further splits — in which case we ~~to~~ are supposed to take majority voting into account in which all samples are classified as the class to which majority of the samples belong to.

c) Model over fitting can occur due to various reasons including :-

i) Presence of coincidental regularities in the data.

ii) Insufficient amount of data available.

iii) Inefficient classification of examples where multiple instances of the example are classified differently.

iv) Model is trained on test test (test set included in training set).

v) Using the same sample dataset for training multiple times.

In general a model is said to overfit if there exists some hypothesis that gives lower accuracy over training set but larger accuracy over test set or unseen data.

In decision tree we can solve it in two ways:-

i) Prepruning :- We halt generating new branches based on some split on some attribute if splitting leads to lowering of goodness of the decision tree below some threshold point. This threshold point is generally hard to determine.

4

ii) Post pruning :- we generate the entire tree and move from the bottom towards the top pruning off branches based on some rule; given that pruning leads to increase in accuracy of decision tree prediction.

6) a) Based on the decision tree model given; the examples ~~can~~ will be classified as follows;

1) Outlook = Sunny; Humidity = Normal; Wind = strong;
Play Tennis = Yes.

2) Outlook = Overcast; Humidity = Normal; Wind = Strong;
Play Tennis = Yes

3) Outlook = Rain; Humidity = High; Wind = Strong;
Play Tennis = No.

4) Outlook = Sunny; Humidity = High; Wind = weak;
Play Tennis = No.

5) Outlook = Rain; Humidity = High; Wind = Strong;
Play Tennis = No.

So the ground truth & classification values are

| | | |
|---|---|---|
| 1) | Yes | Yes |
| 2) | NO | Yes |
| 3) | Yes | No |
| 4) | No | No |
| 5) | No | No |

Confusion matrix :-

| Actual value \ Predicted value | Yes | No |
|---|---|---|
| Yes | TP = 1 | FN = 1 |
| NO | FP = 1 | TN = 2 |

4

i) Precision :- % of positives classified ~~recognized~~ that are actually positive $= \frac{TP}{TP+FP}$

$= \frac{1}{2} = 0.5$

ii) Recall :- % of positives recognized by classifier $= \frac{TP}{TP+FN}$

$= 0.5$

iii) F-score :- Harmonic mean of precision & recall $= 2 \times \text{precision} \times \text{recall} / (\text{precision} + \text{recall})$

$= \frac{2 \times 0.5 \times 0.5}{0.5 + 0.5} = \frac{0.5}{1} = 0.5$ (Ans)

b) i) Holdout method :- In this method the dataset is partitioned into 3 exclusive subsets; 2/3rd of which is used as training dataset and the remaining 1/3rd is used as test data set or validation data set. This is generally suitable for large dataset

ii) Cross-validation :- In Cross validation or K-fold cross validation ; we divide dataset D into K exclusive subsets $\{S_1, S_2, \ldots S_K\}$ and we use ~~sts~~ subset $S_i$ as the test set for the iteration i. generally the value of k is 10, & the method used is 10-fold cross validation method. This is generally suitable for medium sized datasets.

**Bootstrap**:- Or leave one out method is one in which every subset of the k subsets consists of only one sample. This method is generally used for small datasets. Various bootstrap ~~metho~~ method exists; in which one is .632 bootstrap where for a d sized dataset; we sample d times with replacement; where probability of element ending up in test set is $(1-1/d)^d \approx e^{-1} = 0.368$ & probability of element being in bootstrap is $1 - 0.368 = 0.632$

4

e> An ~~es~~ ensemble of classifiers helps to increase predictive accuracy of classification by using methods such as Majority voting, Bagging, Boosting etc. It requires that base learners misclassify different training examples i.e there is no overlap between the classifiers, instead of requiring highly accurate individual base learners.

The AdaBoost or Adaptive boost algorithm works in the following way:-

Model construction:-

i> for all items in a dataset D; weight of each item is the same i.e 1/d where d is no. of items in dataset

ii> ~~In the~~ ~~fi~~ For the first iteration the classifier classifies the training examples.

iii> Using ground truth, we measure error ~~rate~~. If $e=0$; or $e \geq 0.5$; terminate model building

iv> Else for all ~~eitm~~ examples classified correctly; multiply to the existing weight a factor of $\frac{e}{1-e}$

v) Normalize all the weights so that total sum of weights of all examples = 1

Repeat steps ii) to v) for subsequent iterations where adjusted weights are fed to the learner.

Model usage:-

i) Assign weight zero to all the class variables

ii) for each classifier, if classifier predicts the ~~training~~ test sample as class $C_i$; add weight $-\log(e/1-e)$ to class $C_i$; where e is the error rate for that classifier.

iii) Select the class with maximum net weight as the class of the test sample.

3½

– The = * = End –