3) a) Let I be an itemset, $I = \{i_1, i_2, \ldots i_m\}$.

Let t be a transat^n $t \subseteq I$ and T be the dataset of transactions.

Assuming, X and Y to be subsets of transaction dataset such that $X, Y \subset I$ and $X \cap Y = \{\phi\}$

An association rule is given by : $X \rightarrow Y$

which denotes if transact^n X happens then transat^n Y will happen with some probability.

Support : The rule has a support which can be given by prob. of $X \cup Y$.

i.e $\quad \text{Support} = \dfrac{\{X \cup Y\}_{count}}{n}$; n = no. of transact^n

4

Confidence : For the rule above, confidence is the probability of occurrence of Y if transaction X has already been done

$$\text{Confidence} = \frac{\{X \cup Y\}_{count}}{\{X\}_{count}}$$

→ An item set is referred to as frequent item set if its support is greater than or equal to the minimum support.

→ An association rule is referred to as an important rule if it has both support and confidence more than their minimum values.

b) Given

Min-support = 30 %

Min- confidence = 80 %

Size-1 itemset	Frequency
Bread	5
Butter	3
Milk	3
Jelly	1
Coke	2

~~Xou.~~

Support ≥ 0.3

$\Rightarrow \frac{X_{count}}{6} \geq 0.3$

$\Rightarrow (X_{count} \geq 1.8 \approx 2)$

$\therefore F_1 = \{Bread\}, \{Butter\}, \{Milk\}, \cancel{\{Jelly\}} \{Coke\}$

{ F_i : Frequent itemset of size 'i'

Size-2 itemset	Frequency
Bread, Butter	3
Bread, milk	2
Butter, milk	1
Bread, Jelly	1
Butter, Jelly	1
Bread, Coke	1
Milk, Coke	1

$\therefore F_2 = \{Bread, Butter\}, \{Bread, milk\}$

Size-3 itemset	Frequency
Bread, Milk, Butter	1
Bread, Jelly, Butter	1

$F_3 = \{\}$

{ None of the itemset satifies min. support criteria

Subsets of $\cancel{F_2}$ = {Bread, Butter} are

{Bread}, {Butter}

Rule 1: Bread $\longrightarrow$ Butter

Support $= \frac{3}{6} = 0.5$, Confidence $= \frac{3}{5} = 0.6 < 0.8$

(Not important)

Rule 2 : Butter $\rightarrow$ Bread

Support $= \frac{3}{6} = 0.5$, Confidence $= \frac{3}{3} = 1 > 0.8$

(important).

From the set {Bread, milk}

Rule 3 : Bread $\rightarrow$ milk

Support $= \frac{2}{6} = \frac{1}{3} = 0.33$ Confidence $= \frac{2}{5} = 0.4 < 0.8$

(Not important)

Reile 4 : milk $\rightarrow$ Bread

Support $= \frac{2}{6} = 0.33$ Confidence $= \frac{2}{3} = 0.66 < 0.8$

(Not important).

6

$\therefore$ The important rule obtained by Apriori algorithm is :

Butter $\rightarrow$ Bread (Confidence = 100% Support = 50%)

c) ~~Following are the major drawbacks of a-priori algorithm~~

~~It~~ xelres

4) a) If decision is taken on Outlook.

Outlook

Sunny	Rain	Overcast
Yes = 2, No = 2	Yes = 2, No = 2	Yes = 2, No = 0

Play Tennis?

Expected info / Entropy $= \sum_{i=1}^{k} \frac{p_i + n_i}{p+n} I(p_i, n_i)$

$$E(\text{outlook}) = \frac{4}{10} I(2,2) + \frac{4}{10} I(2,2) + \frac{2}{10} I(2,0)$$

$$I(p,n) = -\left(\frac{p}{p+n}\log_2 \frac{p}{p+n} + \frac{n}{p+n}\log_2 \frac{n}{p+n}\right)$$

$$\therefore E(\text{outlook}) = 0.4 \times 1 + 0.4 \times 1 + 0.2 \times 0 = 0.8$$

Humidity

Normal	High
Yes = 4, No = 1	Yes = 2, No = 3

Play Tennis?

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

$$I(4,1) = -\left(\frac{4}{5}\log_2\left(\frac{4}{5}\right) + \frac{1}{5}\log_2\left(\frac{1}{5}\right)\right)$$

$$= 0.8 \times (\log_2(5) - \log_2(4)) + 0.2 \log_2(5)$$

$$= \log_2(5) - 1.6 \log_2(2)$$

$$= 2.3 - (1.6 \times 1) = 0.7$$

Min = Humidity

$$I(2,3) = $$

$$= -\left[\frac{2}{5}\log_2\left(\frac{2}{5}\right) + \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right]$$

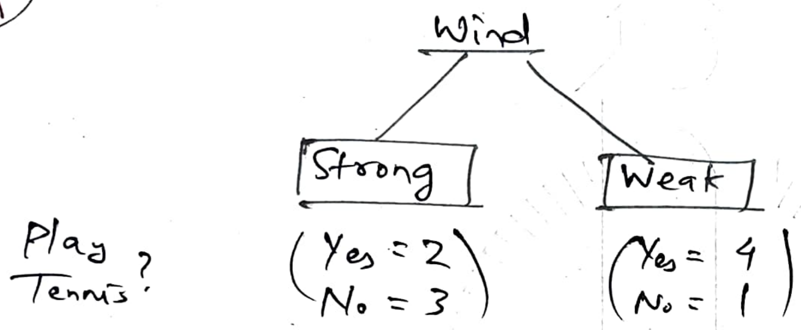
$$= 0.4\left[\log_2 5 - \log_2(2)\right] + 0.6\left[\log_2(5) - \log_2(3)\right]$$

$$= \log_2(5) - 0.4\log_2(2) - 0.6\log_2(3)$$

$$= 2.3 - 0.4 - (0.6\times 1.6) = 1.9 - 0.96 = 0.94$$

$\therefore$ $E(\text{Humidity}) = \frac{1}{2}[0.7 + 0.94] = 0.82$

4



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

$$= 0.82$$

$\therefore$ Lowest entropy is obtained for E(outlook) = 0.8

Hence root will be "Outlook" for the decision tree.

4)b) A decision tree can terminate if all the tuples in the dataset have been classified and the scope of any further classification is over.

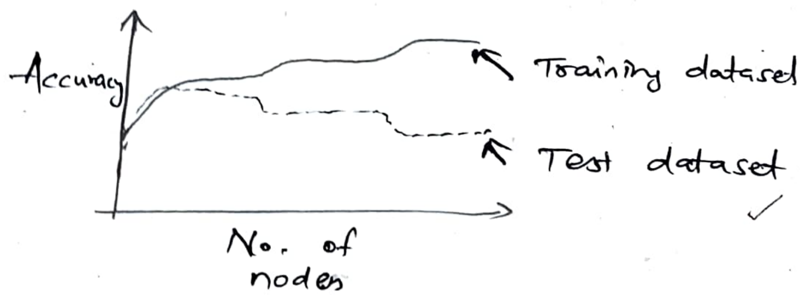
1

Also, the decision tree can terminate if any further classification is not possible by any branch of the decision tree.

1)

1)c) Model overfitting can occur by training too much on a limited data set so that the model tends to generaliz the training data too much and hence gives a large error for all the ~~whole~~ ~~insta~~ other instances of data.

It can also happen that the no. of nodes in the decision tree is so high that it is unable to perform well on the test data.



Overfitting can be avoided in the decision trees by either

3

i) Prepruning: Stopping the expansion of tree before getting to the perfect classification results which effecti helps, not to over-generalise on the training dataset.

ii) Post-pruning: A "fully-grown" decision tree can be pruned by removing some of the brances so that it can retain its performance for the testing dataset too!

5) a) The Naive Bayes classifier is given by the following formula

$$P(C_i|X) = \frac{P(X|C_i) \cdot P(C_i)}{P(X)}$$; X = data sample with unknown class.

C_i = Class label C_i

$$\propto P(X|C_i)\, P(C_i)$$; Since $P(X)$ = const for all classes.

Here, there are two classes i.e "Yes" and "No"

and X = < Outlook = Overcast, Humidity = High, Wind = Weak >.

$$P(Yes|X) \propto P(X|Yes) \cdot P(Yes)$$

$$= P(Overcast|Yes) \cdot P(Humidity = High|Yes) \cdot P(Wind = Weak|Yes) \times P(Yes).$$

5

$P(Overcast|Yes) = 0$

$P(Humidity = High|Yes) = \frac{1}{2}$

$P(Wind = Weak|Yes) = 0$

$P(Yes) = \frac{2}{5}$

$\Rightarrow P(Yes|X) = 0.$

$$P(No|X) = P(Overcast|No) \cdot P(Humidity = High|No)\, P(Wind = Weak|No) \cdot P(No)$$

$$= \left(\frac{1}{3}\right) \times \left(\frac{2}{3}\right) \times \left(\frac{1}{3}\right) \times \left(\frac{3}{5}\right) = \frac{2}{45}$$

Thus, prediction will be Play Tennis = "No"

5) b) There is an error in calculating $P(X \mid Yes)$ since two of the probabilities turn out to be zero.

This can be corrected using Laplacian correction method.

For $C_i =$ "Yes"

Outlook
Sunny
Rain

$\xrightarrow[\text{"overcast"}]{\text{Add}}$

Outlook
Sunny
Rain
Overcast

Also ~~C_i~~ → Play tennis = "Yes"

3

Wind
Strong
Strong

$\xrightarrow[\text{tuple of "weak"}]{\text{Add 1}}$

Wind
Weak
Strong
Strong

Modified probabilities

$\therefore P(\text{Overcast} \mid \text{Yes}) = \frac{1}{3}$

$P(\text{wind} = \text{weak} \mid \text{Yes}) = \frac{1}{3}.$

$\therefore P(\text{Yes} \mid X) = \frac{1}{3} \times \frac{1}{2} \times \frac{1}{3} \times \frac{2}{5} = \frac{1}{45}.$

and as before $P(\text{No} \mid X) = \frac{2}{45}.$

On normalisation, $P(\text{Yes} \mid x) = 1/3$ and $P(\text{No} \mid x) = 2/3$

$\therefore$ Prediction will be Play Tennis = "No."

5) c) If any feature like Humidity has continuous values, we can use this algorithm but in a different manner.

We can use Gaussian Naive ~~Bayes~~ Bayes Classifier, where the ~~∞~~ probabilities for continuous variables will be given by

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad ; \quad \mu = \text{mean of the variable}, \quad \sigma^2 = \text{Variance of the variable.}$$

3

Rest process of finding the ~~classifier~~ class will still remain the same.

$$P(C_i|X) = P(X|C_i) \cdot P(C_i)$$

$$= P(C_i) \prod_{i=1}^{K} P(x_k|C_i) \quad \left\{ \text{Assuming independence of the attributes} \right.$$

The max value of $P(C_k|X)$ $\forall$ $k \in \{1, 2, \ldots n\}$ will give the class label for an instance.

6) a)

Following is the confusion matrix for the given model

Actual \ Prediction	Yes	No
Yes	1	1
No	1	2

True positive $\begin{pmatrix} \text{Actual} = Y \\ \text{Predict}^n = Y \end{pmatrix} = 1$ (Sunny, Normal, Strong)

False positive $\begin{pmatrix} \text{Actual} = N \\ \text{Predict}^n = Y \end{pmatrix} = 1$ (Overcast, Normal, Strong)

True negative $\begin{pmatrix} \text{Actual} = N \\ \text{Predict}^n = N \end{pmatrix} = 2$ (Sunny, High, weak / Rain, High, strong)

False negative $\begin{pmatrix} \text{Actual} = Y \\ \text{Predict}^n = N \end{pmatrix} = 1$ (Rain, High, Strong)

4

i) Precision $= \dfrac{\text{True positive}}{\text{True positive} + \text{False positive}} = \dfrac{1}{1+1} = \dfrac{1}{2} = 50\%$

ii) Recall $= \dfrac{\text{True positive}}{\text{True positive} + \text{False negative}} = \dfrac{1}{1+1} = \dfrac{1}{2} = 50\%$

iii) F-score = Harmonic mean of Recall & Precision

$$= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$= \frac{2 \times \frac{1}{2} \times \frac{1}{2}}{\frac{1}{2} + \frac{1}{2}} = \frac{1}{2}.$$

6) b)

i) Holdout method: In this method, the data is divided into two parts for training and testing separately.

If the same data is used for training & testing purpose then there are chances that model might overfit.

Hence to actually evaluate the performance a separate testing data is used.

ii) Cross-validation: The data is divided into k subsets and each time (for k iterations) one of the k subsets is used for testing and the rest (k-1) subsets are used for training.

4

k=3 →

train	train	test
test	train	train
train	test	train

iii) In bootstrap, a fixed no. of tuples are sampled with replacement from the training dataset and the tuples which are left in this process are used for testing the model.

For example in 0.632 bootstrap, data is sampled for d times hence there are $\left(1-\frac{1}{d}\right)^d \approx e^{-1}$ no. of tuples that can be left out, which will further be used for testing.

6) c) The main purpose of ensemble of classifiers is to increase the accuracy the classification accuracy by combining multiple classifiers and then making a decision based on what majority of the classifiers are saying. This classification method can significantly increase the accuracy of models.

Adaboost algorithm

This ~~is~~ algorithm ~~trains~~ its classifiers iteratively in a sequential manner. Here, voting is based not on majority but on the weights assigned to the classifiers according to their performance. Idea here is to make ~~a~~ any classifier to train more on those data which were incorrectly classified by its previous classifier.

3

Steps:

1) Assign equal weights to all the tuples
2) Let the classifier be tested to the data.
3) Find the error(e) by taking weighted avg. of errors in each tuple
4) Find the performance index given by ~~$\frac{1}{2}\ln\left(\frac{1-e}{e}\right)$~~ $\frac{1}{2}\ln\left(\frac{1-e}{e}\right)$
5) Update weights of the tuples
   i) For incorrectly classified ones: multiply by $e^{\text{perfo. index}}$
   ii) For correctly classified ones: divide by $e^{\text{perf. index}}$

This helps to put more emphasis on correctly classifying the incorrectly classified data of this classifier ~~f~~ for the next classifier.