—— I^st Internals.

05/01/2023

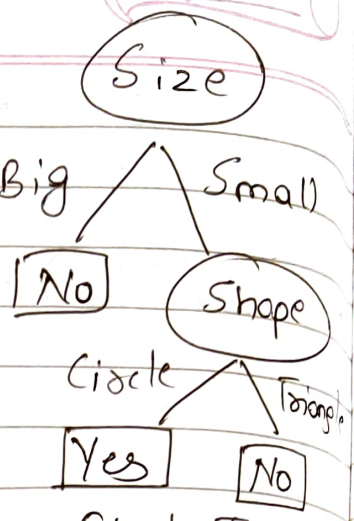
Issues in Decision Tree Learning.

* Avoiding overfitting the data.
* Incorporating Continous valued attributes
* Alternate measures for selecting the attribute
* Handling training examples with missing attribute value
* " Attributes with differ in cost.

Shape	+ve	-ve	
Triangle	0	1	=> 0
Circle	2	0	=> 0

$I(Shape) = 0$

Gain = 0.918



Final Tree.

Continued.

Issues in Decesion ~~Training~~ Tree learning.

* Incorporating continued valued attributes.

=> If we have " " s we can make it as Discrete value by using Discrete valued intervals.

ex! - Age - 4, 10, 14, 24, 24, 34, 64, 84, 95

[child, Young, Adult, Teenage Senior Vsenior

0-5 6-12 20-35 13-19 60-79 80-100

In some attributes we can find Discrete valued intervals we use other method where we find to calculate threshold value 'c'

Step 1: sort

2:- Look for where value classification is changing

3:. And calculate Gain as the previous sum.

Temp	40	48	60	72	80	90
Play Tennis	No	No	Yes	Yes	Yes	No

$\frac{48+60}{2} = 54$

Threshold value 'c'

54

48 48 60 72 80 90

$\frac{80+90}{2} = 85$

85

K-NN Algorithm:-
-> K-Nearest Neighbour Algorithm
-> Multi classification
-> K stands for no. of neighbours.
-> The minimum value of K is always 3

Distance Measures.
-> Euclidean Distance

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

-> Manhattan Distance

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

Value of K will be changed based on the Accuracy

Confusion Matrix

	0	1
0	00	01
1	10	11

0 -> 0 -> True +ve
0 -> 1 -> false Neg
1 -> 0 -> 11 +ve
1 -> 1 True -ve

↓
No of Correct predictions

$$\text{Accuracy} = \frac{00 + 11}{00 + 01 + 10 + 11}$$

Issues continued.
3) Alternate measures for selecting the attribute.
Information Gain

It favours for the attributes which has more value.

Disad:- Broader tree at one level.

Alternate measure is Gain Ratio.

↓

Split information

$$\text{Split info} = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log \frac{|S_i|}{|S|}$$

where S_1 to S_c are 'c' subsets of examples resulting from partitioning 's' by the 'c' valued attribute A.

$$\text{Gain Ratio}(S, A) = \frac{\text{Gain}(S, A)}{\text{Split info}(S, A)}$$

08/01/2024

1) Handle the training examples with missing attribute

Gain (S, A) A(x) = temp / ~~Humidity~~ / wind / Outlook
hot strg sunny

Training example $\langle x, c(x) \rangle$, A(x) => Null

2 strategies

1) whichever value is max in that attribute then fill it with that max value irrespective of c(x)
2) c(x) of that instance if it +ve or -ve check the target concept whichever is +ve ~~take~~ or -ve then fill it correspondingly. acc to c(x) value.

Handling attributes with diff costs.

Some times " are associated with costs/ weight.

ep:- To predict medical disease

Temp | Pulse | BloodTest | Biopsy | Target

$$\frac{\text{Gain}^2(S, A)}{\text{Cost}(A)}$$

null
Based on ~~Dat~~ Value.

v. imp Avoiding Overfitting in the Data.

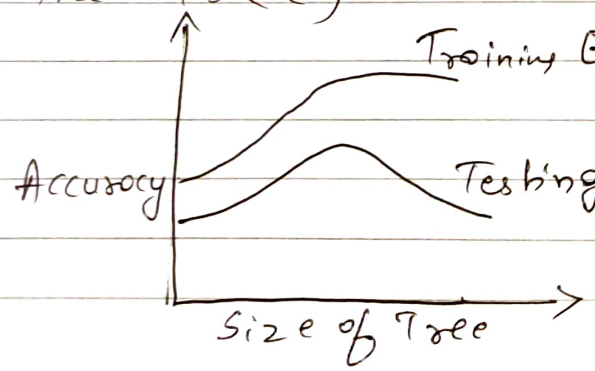
less more
~~more~~ error in train data & ~~less~~ error in test data.

when we apply hypothesis on training sample it show less error
" " " some " " testing sample it shows more error. because it has learnt more than ~~#~~ cut is required.

Given an hypothesis h, it said to overfit the training examples if the hypothesis h is performing better over the training set than over the testing set. so Training error(h) < Testing error(h).
error less accuracy more.

why is this overfitting causing in ID3
-> Noise.
-> if Size of data set is small They are not in a position to test.

eg:- 1 (Noise)
· (outlook=sunny, Temp=hot, Humidity=Normal, wind=storm : Play Tennis = NO)
(for the Tree)



It

Avoiding overfitting in the Data.

Strategy:-

① Stop growing the tree after certain level

② Grow the tree completely, allow overfitting to occur, then we go for pruning the tree.

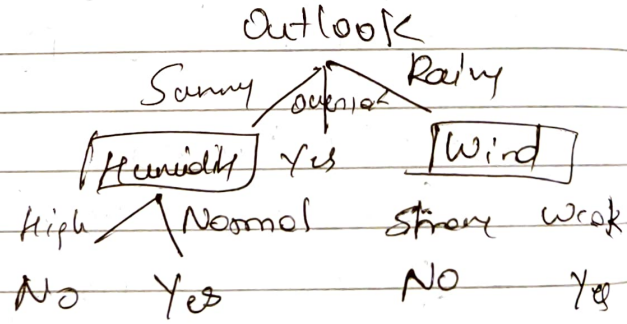
① Disadv:- It is not easy to determine where we stop growing the tree. (practical difficult)

② It is mostly used strategy.

② — i reduced error pruning.
   — ii Rule-post pruning.

i In the tree that is already built, every Decision node becomes candidate to prune, pruning is made until we get high accuracy. or while pruning accuracy is checked if it is high it is chopped off and replaced with leaf node.

Humidity & wind



i) It Considers each of the Decision nodes in the tree to be Candidate for pruning. Pruning a decision node consists of removing the subtree rooted at that node, making it a leaf node and assigning it to the most ~~complication~~ common

Nodes are removed only if the resulting pruned tree performs better that the original tree for

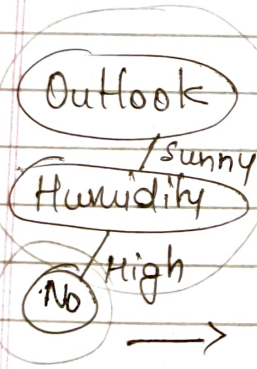
Pruning of tree continues until further pruning is harmful. The pruning is stopped one the accuracy ~~start~~ starts decreasing.

ii) Rule-post-Pruning.

After the tree is built it is ~~set~~ converted into set of rules. for every Leaf node there is a rule. Rule is in the form of path from the root node to Leaf node.

and

Example: if (outlook = sunny) $\wedge$ (Humidity = high) then (playtennis = No)



This path is known as Rule Antecedent (pre-condition)

$\longrightarrow$ Rule Consequent (post-condition)

- Only pre-condition is pruned.

1) * Build a Decision Tree from the Training set, grow the tree completely until training data is fit allow the overfitting to occur.
* Convert the tree into set of rules by creating a rule for each path from the root node to leaf node.
* Prune each rule by removing any tree conditions so that the accuracy is improved.
* we generate one rule for each leaf node in tree.
* The path from the root to leaf becomes Rule Antecedent or : pre-condition.
* A classification at leaf node becomes the rules consequent or the the post-condition.

* Each rule is pruned by removing the Antecedant or the precondition.
* This process is continued until there is improvement in the estimated accuracy.
* No pruning is performed if there is no improvement in the accuracy.

9/01/2024 3rd Module.

Bayesian learning.

Computing probabilities to find some inferences/ prediction

Bayes Theorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

$P(B|A)$ → Likelihood
$P(A)$ → Prior Probability
$P(A|B)$ → Posterior Probability
$P(B)$ → Marginal Probability.

Ex:- What is the probability of finding the king Given that it is a facecard in a pack of card.

facecards = J & K $4 \times 3 = 12$

$$P(\text{King} | \text{facecard}) = \frac{P(\text{face} | \text{king}) \cdot P(\text{king})}{P(\text{face})}$$

$$= \frac{4/4 \times 4/52}{12/52}$$

$$1 \times \frac{4}{52} \times \frac{52}{12} = 3$$

or $\frac{4}{12} = \frac{1}{3}$

$$P(K/F) = \frac{1}{3}$$

→ Bayesian learning provides a probabilistic approach for inferences.

→ Bayesian learning Algorithms calculate explicit probabilities for hypothesis and are among the most practical approaches for certain types of learning problems

imp features of Bayesian learning

* Each training example can incrementally increase or decrease the estimated probability that the hypothesis is correct. This provides more flexibility than Algorithms that completely eliminate hypothesis if it is found to be inconsistent.

* Prior information can be combined with observed data to determine the final probability of the hypothesis

* Bayesian method can accomodate hypothesis that makes probabilistic predictions.

classmate
Date
Page

* New instances can be classified by combining the predictions of multiple hypothesis weighted by their probabilities

Bayes Theorem:-

It provides a way to calculate the probability of hypothesis based on its prior probability, The probability of observing various data given the hypothesis, and the observed data it self

$$P(h|D) = \frac{P(D|h) * P(h)}{P(D)}$$

$P(h)$ → Prior P of h

$P(D)$ → marginal P

$P(D)$ = prior probability that training Data D will be observed, i.e the probability of D given no knowledge about which hypothesis holds.

$P(D/h)$ = probability of observing data D in which the hypothesis h, holds. we are interested in $P(h/D)$ ie the prob that h holds given the observed training data D. It is also called posterior prob of h

Many a times we have set of hypothesis h and we are interesting in finding the most probable hypothesis Data D. Any such maximally probable hypothesis is called maximum a posterious hypothesis (MAP)

$$h_{map} = \underset{h \in H}{\arg\max}\ P(h|D)$$

$$\Rightarrow \underset{h \in H}{\text{argmax}} \frac{P(D|h) * P(h)}{P(D)}$$

maximum a posterior probability hypothesis.

$$h_{map} = \arg\max P(D|h) \times P(h)$$

Consider a medical diagnosis problem in which ~~there~~ there are 2 alternative hypothesis

→ That the patient has particular form of cancer
→ That ~~the~~ ~~don't~~ doesn't have.

→ The Available data is from a particular with 2 possible outcome +ve or -ve. There is prior knowledge a
also a lab test.

The test returns a possible result in only 98%.

→ The available data is from a particular lab test with 2 possible outcomes i.e +ve and -ve. There is prior knowledge that over entire population of people only 0.008 have this Disease also lab test is only an imperfect indicator of this disease. The test returns a correct +ve result in only 98% of the cases in which the disease is actually present and a correct -ve result is only 97% of cases in which the disease is not present. In other cases the test returns the opposite result. Suppose we have a new patient for whom the lab test returns a +ve result should be diagnosed the patient has having cancer or not.

18/01/2024

conditional probability. cancer is there but it is +ve in 98%.

Solu^n :- ~~Consider~~

$P(\text{cancer}) = 0.008$

$P(\oplus/\text{cancer}) = 0.98$ $\quad P(\ominus/\sim\text{cancer}) = 0.97$

+ve already when they have cancer

$P(\oplus/\sim\text{cancer}) = 0.03$

$1-0.008$

$P(\sim\text{cancer}) = 0.992$ $\quad 1-0.98$

They have cancer & give -ve result, $P(\ominus/\text{cancer}) = 0.02$

$h_1 = \text{cancer}$ $\quad$ ~~$h_2 = \text{not cancer}$~~ $h_2 = \sim\text{cancer}$

$h_{map} = \underset{h \in H}{\arg\max}\ P(h|D)$

$h_{map} = P(D/h) \times P(h)$

$= P(\oplus/\text{cancer}) \cdot P(\text{cancer}) = 0.98 \times 0.008$

$= 0.0078.$

$h_2 map = P(\oplus/\sim\text{cancer}) * P(\sim\text{cancer})$

$= 0.03 * 0.992$

$= 0.0298$

$h_{map} = \sim\text{Cancer}$

---

* Bayes Theorem and Concept learning.

$c : X \to \{0,1\}$

$\langle\langle x_1, d_1\rangle \langle x_2, d_2\rangle \ldots \langle x_m, d_m\rangle\rangle$

x_i = instances, target $= d_i$

$d_i = c(x_i)$

based on the instance concept is learnt & target is found.

x_i	d_i
<Sun warm hot ...>	Yes
<Sun hot cold ...>	no

$\langle x_1, x_2 \ldots x_m \rangle$ - set of instances are fixed.

$D = \{d_1, d_2 \ldots d_m\}$

last column can be made as set which belongs to positive

Bruteforce map learning Algorithm.

Bayes Theorem and Concept learning

~~Concept~~ Consider a finite Hypothesis space H defined over an instance space X. Task is to learn some target concept $c: X \to \{0,1\}$

Let the sequence of training examples be $\{\langle x_1, d_1\rangle, \langle x_2, d_2\rangle \ldots \langle x_m, d_m\rangle\}$ where x_i is some instance from X and d_i is the target value of x_i, i.e $d_i = c(x_i)$

We assume that the sequences of instances $\langle x_1, x_2 \ldots x_m\rangle$ is held fixed so that the training data ~~set~~ D can be written just as a sequence of target values $D = \{d_1, d_2 \ldots d_m\}$

We can design a concept learning algorithm to output the map hypothesis based on bayes theorem which we call it as. Brute force MAP learning Algorithm.

* Brute force MAP Learning Algorithm

imp.

Step 1:- for each hypothesis h in H calculate the posterior probability $P(h|D) = \frac{P(D|h) * P(h)}{P(D)}$

Step 2:- Output the hypothesis h_{map} with the highest posterior probability i.e $h_{map} = \underset{h \in H}{\text{argmax}}\ P(h|D)$

$$h_{map} = \underset{h \in H}{\text{argmax}} \frac{P(D|h) * P(h)}{P(D)}$$

$P(h) = \frac{1}{|H|}$ $h_1 = \dots$ $h_2 \dots$ $h_3 = \dots$

$h_1 + h_2 + h_3 \dots = 1$

when $d_i = h(x_i) \Rightarrow 1$ (Consistent)

" $\neq$ " $\Rightarrow 0$ (in " ")

Assumptions.

Here we must specify what values are to be used for $P(h)$ and $P(D|h)$. In order to choose the probability distribution $p(h)$ & $P(D|h)$ we make the following assumptions.

1) The training data is noise free i.e $d_i = c(x_i)$
2) The target concept c is contained in the hypothesis space H
3) We have no prior reason to believe that any hypothesis is more probable than the other

Given these assumptions what values do we have to

Given that no prior knowledge is available that one hypothesis is more likely than an other. It is reasonable to assign the same prior probability to every hypothesis h in H. Also these prior probability sum to one. Hence we can write

$P(h) = \frac{1}{|H|}$ $\forall$ h in H

$P(D|h)$ is the probability of observing the target values $D = \{d_1, d_2 \dots d_m\}$ for the fixed set of instances $\langle x_1, x_2 \dots x_m \rangle$ given that hypothesis h holds. Since we assume noise free data the Probability of observing the classifica" is one, if $d_i = h(x_i)$ and 0 if $d_i \neq h(x_i)$ $\therefore P(D|h) =$

$\begin{cases} 1, & \text{if } d_i = h(x_i) \ \forall \ d_i \text{ in } D \\ 0, & \text{otherwise.} \end{cases}$

In other words the probability of data Given hypothesis h is 1 if d is consisent

$$P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)}$$

with h and 0 otherwise

Given these choices for $P(h)$ & $P(D|h)$ we now have a fully defined bruteforce map learning Algorithm.

Compute the posterior probability $P(h|D)$ of every hypothesis using Baye's Theorem

Case 1:- h is inconsistent with the Training data D, $P(h|D) = \frac{0 \times P(h)}{P(D)} = 0$ $P(D|h) = 0$.

The Posterior probability of hypothesis inconsistent with D is always 0.

Case 2: h is consistent with D.

$$P(h|D) = \frac{1 * 1/|H|}{\frac{|VS_{H,D}|}{|H|}} = \frac{1}{|VS_{H,D}|}$$

where $VS_{H,D}$ is a subset of Hypothesis from H that are consistent with D, ie it is the version space w.r. with respect to D.

To summarize we can write

$$P(h|D) = \begin{cases} 1/|VS_{H,D}|, & \text{if } h \text{ is consistent} \\ 0, & \text{otherwise} \end{cases}$$

* with this we can say <u>every consistent hypothesis is a map hypothesis</u>

Note:- $h_{map} = \underset{h \in H}{\operatorname{argmax}}\ P(D|h) * P(h)$ when Prior P is not given we take some value & hence ignore it and write it as

$h_{ML} = \underset{h \in H}{\operatorname{argmax}}\ P(D|h)$

In some cases we assume that every hypothesis in this equally probable that is $P(h_i) = P(h_j)$ for all h_i & h_j belonging to H.
$P(h_i) = P(h_j) \forall h_i, h_j \in H$.
In this case further we simplify.

$$h_{map} = \underset{h \in H}{\operatorname{argmax}}\ P(D|h) * P(h)$$

and need to consider only the term $P(D|h)$ to find the most probable hypothesis.

$P(D|h)$ is often called as the Likelihood of the $P(D|h)$ ——→ and any hypothesis that maximises $P(D|h)$ is called maximum likelihood hypothesis denoted by h_{ML}.

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}}\ P(D|h)$$

22/01/2024

v.imp ML and LS Error Hypothesis

Maximum Likelihood & least squared Error

$f: X \longrightarrow R$ (Continous valued target

In the target by associating noise we can make it Continous valued.

we consider a problem of learning a continous value target function. Consider an instance space X and the hypothesis space H consisting of some class of Real valued functions over X i.e each h in H is of the form of function $h: X \to R$ where R is the set of Real numbers

The problem faced by the learner L is to learn an unknown target function $f: X \to R$
A set of 'm' training examples are provided

p = probability density function

classmate
Date
Page

where the target value of each example is corrupted by a random noise drawn according to a normal probability distribution.

In other words each training ex is of the form $\langle x_i, d_i \rangle$ where $d_i = f(x_i) + e_i$, where $f(x_i)$ is noise free value of target func^n & e_i is the random variable representing the same

imp Bayesian Analysis will show that under certain assumptions any learning algorithm that minimizes the squared error between the output hypothesis predictions and the training data will output a maximum likelihood hypothesis.

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}}\ P(D|h)$$
$$h_{ML} = \underset{h \in H}{\operatorname{argmax}}\ p(D|h)$$

∴ we are now dealing with continous value

$(\text{Error})^2 = (\text{Actual} - \text{Predicted})^2$ or $(\text{Predicted} - \text{Actual})^2$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}}\ P(D|h) \Rightarrow LS$$

$\therefore h_{ML} = LS$

We show this by considering ML hypothesis and we start with the equation.

→ we write the same eqn with p to refer to the probability density

→ we are taking every instance as independent hence we are taking product. eg:- $P(A) \cdot P(B)$

probability of Density Normal Distribution $= p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

$\log e^x = x$

Assuming that training examples are mutual independent, given h we can write $p(D|h)$ as the product of various $p(d_i|h)$ that can be shown as

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} p(d_i|h)$$

$p(D_i|h)$ can be written as a normal distribution with variance σ^2. and mean $\mu = f(x_i)$ or $h(x_i)$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - \mu)^2}$$

$$= \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i - h(x_i))^2}$$

$\log(ab) = \log a + \log b$

$$= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \ln\left(e^{-\frac{1}{2\sigma^2}(d_i - h(x_i))^2}\right)$$

(constant we can discard.)

$$= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

$\Rightarrow$ independent of h

$$\underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} -\frac{1}{2\sigma^2}(d_i - h(x_i))^2$$ // maximizing -ve term same minimi +ve term.

$$h_{ML} = \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^{m} \frac{1}{2\sigma^2}(d_i - h(x_i))^2$$

$$h_{ML} = \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

ML = least squared error

23/01/2024

imp. Maximum Likelihood hypothesis for predicting Probabilities.

Learner -> learn to predict probabilities.

Medical data set X -> instance.
(it contains symptoms of Disease

Target func^n $f(x) = 1$ => survives
$f(x) = 0$ => Dies

It is a probabilistic func^n

set of patients 92% survives 8% dies.

$f : X \to [0,1] \Rightarrow P(f(x) = 1) = 0.92$

$f' = P(f(x) = 1) = 0.92$

what is the p that patient survive / $P(f(x) = 1)$

f' => Machine should learn this.

* The objective is to learn to predict probabilities suppose we have a medical data set instance space X represents patience ~~hoving~~ in terms of symptoms of a disease. Target function $f(x) = 1$, if the patient survive the disease and $f(x) = 0$, if he doesn't.
Here f is a probabilistic function. Suppose we have a collection of patients exibhiting the same set of symptoms, we might find that 92% survives & 8% do not. We want to find out what is the probability that $f(x) = 1$ for this we learn a target function $f' : X \to [0,1]$ such that $f'(x) = P(f(x) = 1)$

Based on our example $f'(x) = 0.92 = P(f(x) = 1)$
To learn this function f' we desire a maximum likelihood hypothesis for f'.

because it is indepent we can write in product form.

$D = (x_1 d_1 x_2 d_2 \ldots\ldots x_i^{\circ} d_i)$

For this, we must obtain an expression for $P(D/h)$

$$P(D/h) = \prod_{i=1}^{m} P(x_i d_i / h) \rightarrow (1)$$

Let us assume that training data D, is of the form $D = \{\langle x_1 d_1 \rangle \langle x_2 d_2 \rangle \ldots\ldots \langle x_m, d_m \rangle\}$ where $d_i = 0$ or 1 for $f(x_i)$.

Treating both x_i and d_i as random variables and assuming that every training example is independent of other.

we can write $P(D/h)$ as (1)

(only target d_i depends on hypothesis not the patient details or x_i ordi hypo = {survive, dies})

we assume that the probability of any particular instance x_i is independent of hypothesis h, when x is independent of h and survival of the patient alone i.e d_i depends on h we can rewrite (1) as below

$$P(D/h) = \prod_{i=1}^{m} P(d_i/h, x_i) \times P(x_i) \rightarrow (2)$$

Now what is the prob $P(d_i/h, x_i)$ of observing $d_i = 1$ for a single instance x_i, given that hypothesis h holds, &

h is our hypothesis regarding ~~this~~ ~~the~~ target func^n which computes this probability.

Therefore $P(D_i = 1 / h(x_i)) = h(x_i)$ and in general is:

$$P(d_i|h, x_i) = \begin{cases} h(x_i) & \text{if } d_i = 1 \\ 1 - h(x_i) & \text{if } d_i = 0 \end{cases} \rightarrow (3)$$

In order to substitute (3) in (2) ~~for~~ $P(D|h)$ we re express 3 as follows, that is for each

$$P(d_i|h, x_i) = h(x_i)^{d_i} \cdot (1 - h(x_i))^{1-d_i} \rightarrow (4)$$

Substitute eq (4) in eq 2

$$P(D|h) = \prod_{m}^{m} (1 - h(x_i))^{1-d_i} \times P(x_i) \rightarrow (5)$$

Now we write an expression for max Likelihood hypothesis.

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{m} h(x_i)^{d_i} (1 - h(x_i))^{1-d_i} \times P(x_i) \rightarrow (6)$$

$P(x_i)$ is independent of h so we will drop it, for rest of the equation by applying log transformation.

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} d_i \cdot h(x_i) + (1 - d_i) \log(1 - h(x_i)) \rightarrow (7)$$

Equation 7 describes the Quantity that must be maximised inorder to obtain the maximum likelihood hypothesis for the given problem setting.

* imp Naive Bayes Classifier.

objective: predict class of the new sample / same as KNN

It is a highly practical bayesian learning method. The objective of Naive Bayes Classifier is to predict the class of the new sample. It Applies to learning task where each instance x is described by Conjunction of Attribute values and where the target function $f(x)$ can take on any value from some finite set of V

09/01/24 where we can represent datasets as a conjunction of attributes.

Tasks → $\langle a_1\ a_2\ a_3\ a_4 \rangle$

if $a_1 \wedge a_2 \wedge a_3 \wedge a_4$ then ES = Yes / No

(finite set V = {contains all target values yes no} collection of target values)

"Naive Bayes Classifier always predicts the most probable target value. (V_{MAP})

$$V_{map} = \underset{h \in H}{\operatorname{argmax}}\ P(V_j / a_1, a_2, a_3 \ldots a_n)$$

A set of training examples of

describes the

and the new instances.

A Set of training examples of target function is provided and a new instance is described by the tuple of attribute values $\langle a_1, a_2, a_3 \ldots a_n \rangle$ the learner is asked to predict the value of the target class for the new instance. the Bayesian approach for classifying the new approach is to assign the most probable target value denoted as V_{MAP}, given the attribute values $\langle a_1, a_2 \ldots a_n \rangle$ that describe the new instance. we can write V_{MAP} as:

$$V_{MAP} = \underset{V \in V}{\operatorname{argmax}}\ P(V_j / a_1\ a_2 \ldots a_n)$$

we can use Bayes theorem to rewrite this expression as follows.

$$V_{MAP} = \underset{V_j \in V}{\operatorname{argmax}} \frac{P(a_1\ a_2 \ldots a_n / V_j) \times P(V_j)}{P(a_1, a_2 \ldots a_n)}$$

$$V_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \; P(a_1, a_2 \ldots a_n / v_j) \times P(v_j) \rightarrow (1)$$

we can attempt to calculate the second term in equation (1) i.e it is easy to estimate each of ~~$V_{MAP} = \arg$~~ the $P(v_j)$ by counting the frequency with which each target value v_j ~~how to~~ occurs in the training data, and we need to know how is to estimate the first term.

The Naive Baye's classifier is based on the ~~we can attempt to calculate~~ assumption that the attribute values are conditionally independent given the target value in other words the assumption is that given the target value of the instance the probability of the conjunction $\langle a_1 a_2 \ldots a_n \rangle$ is just the product of the probabilities of the individual attributes i.e $P(a_1 a_2 \ldots a_n / v_j)$ can be written as $P(a_1 a_2 \ldots a_n / v_j) =$

$$\Pi_i (a_i / v_j) \rightarrow (2)$$

Substitute (2) in (1), we have the approach used by Naive bayes Classifier as follows.

$$V_{NB} = \underset{v_j \in V}{\operatorname{argmax}} \; P(v_j) \times \Pi_i (a_i / v_j) \rightarrow (3)$$

where V_{NB} denotes the target value o/p by the Naive Bayes classifier.

$$V_{NB} = \underset{v_j \in V}{\operatorname{argmax}}\ P(v_j) \times \pi_i P(a_i / v_j)$$

Example-1

* Playtennis dataset

new instance = (outlook = sunny, temp = cool, humidity = high, wind = strong)

$$v_{NB} = \underset{v_j \in \{Yes, No\}}{\operatorname{argmax}}\ P(v_j) \times P(\text{outlook} = \text{sunny} / v_j) * P(\text{temp} = \text{cool} / v_j) * P(\text{humidity} = \text{high} / v_j) \times P(\text{wind} = \text{strong} / v_j)$$

$$v_{NB}(yes) = \underset{v_j \in Yes}{\operatorname{argmax}}\ P(yes) * P(\text{sunny} / yes) * P(\text{cool} / yes) * P(\text{high} / yes) \times P(\text{strong} / yes)$$

$$= \frac{9}{14} * \frac{2}{9} * \frac{3}{9} * \frac{3}{9} * \frac{3}{9}$$

~~(0.642) (0.142) (0.214) (0.214) (0.214)~~

(0.642) (0.23) (0.34) (0.34) (0.34)

$= 0.005$

$$V_{NB}(No) = \underset{v_j \in No}{\operatorname{argmax}}\ P(No) * P(\text{Sunny} / No) * P(\text{cool} / No) * P(\text{high} / No) * P(\text{strong} / No)$$

$$= \frac{5}{14} * \frac{3}{5} * \frac{1}{5} * \frac{4}{5} * \frac{3}{5}$$

(0.357) (0.6) (0.2) (0.8) (0.6)

$= 0.02$

$V_{NB}(No) > V_{NB}(Yes)$

hence new instance is classified as No.

Example-2

#	Color	legs	height	Smelly	Species.
1	white	3	short	yes	M
2	Green	2	Tall	No	M
3	Green	3	short	yes	M
4	white	3	short	yes	M
5	Green	2	short	No	H
6	white	2	Tall	No	H
7	white	2	short	Yes	H
8	white	2	Tall	No	H

$P(Spec=M) = 4/8$

$P(Spe=H) = \frac{4}{8}$

New instance = Color = Green, legs = 2, Height = Tall, Smelly = No)

$V_{NB} = \underset{v_j \in (M,H)}{\text{argmax}} \; P(v_j) \times P(Color=Green/v_j) \times P(legs=2/v_j) \times P(Height=Tall/v_j) \cdot P(Smelly=No/v_j)$

$V_{NB} = \underset{v_j \in M}{\text{argmax}} \; P(M) \times P(Green/M) \times P(2/M) \times P(Toll/M) \times P(No/M)$

$\frac{4}{8} \times \frac{2}{4} \times \frac{1}{4} \times \frac{1}{4} \times \frac{1}{4}$

$(0.5)(0.5)(0.25)(0.25)(0.25)$

$= 0.0039$

$V_{NB} = \underset{v_j \in H}{\text{argmax}} \; P(H) \times P(Grw/H) \; P(2/H) \; P(Toll/H) \; P(No/H)$

$= \frac{4}{8} \times \frac{1}{4} \times \frac{4}{4} \times \frac{2}{4} \times \frac{3}{4}$

$(0.5)(0.25)(1)(0.5)(0.75) = 0.0468$

$v_{NB}(H) > v_{NB}(M)$

new instance is classified as H.

Example-3

#	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4.	Yellow	Sports	Domestic	No
5.	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes
.	New	Red,	SUV,	Domestic.

$P(Yes) = \frac{5}{10} = \frac{1}{2}$

$P(No) = \frac{5}{10} = \frac{1}{2}$

$v_{NB} = \underset{v_j \in Yes}{\operatorname{argmax}} P(Yes) \cdot P(Red/Yes) \cdot P(SUV/Yes) \cdot P(Domestic/Yes)$

$= \frac{5}{10} \cdot \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{2}{5}$

$(0.5)(0.6)(0.2)(0.4) = 0.024$

$v_{NB} = \underset{v_j \in No}{\operatorname{argmax}} P(No) \cdot P(Red/No) \cdot P(SUV/No) \cdot P(Domestic/No)$

$= \frac{5}{10} \cdot \frac{2}{5} \cdot \frac{3}{5} \cdot \frac{3}{5}$

$= (0.5)(0.4)(0.6)(0.6) = 0.072$

$v_{NB}(No) > v_{NB}(Yes)$

∴ Hence new instance is classified as No.

Example - 4

#	A	B	C	Class.
1	0	0	0	+
2	0	0	1	−
3	0	1	1	−
4	0	1	1	−
5	0	0	1	+
6	1	0	1	+
7	1	0	1	−
8	1	0	1	−
9	1	1	1	+
10	1	0	1	+

$P(+) \frac{5}{10}$ $P(-) = \frac{5}{10}$

new inst (0 1 0)

$P(+)\ (0/+) \cdot P(1/+)\ P(0/+)$

$V_{NB} = \frac{5}{10} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{1}{5}$

$= (0.5)(0.4)(0.2)(0.2) = 0.008.$

$V_{NB} = P(-)\ P(0/-)\ P(1/-)\ P(0/-)$

$(0.5) \cdot \frac{3}{5} \cdot \frac{2}{5} \cdot 0 \quad = 0.$

$V_{NB}(+) > V_{NB}(-)$

Hence new instance can be classified as '+'.

Example - 5 $P(+) = \frac{4}{7}$ $P(-) = \frac{3}{7}$

new instance (Japan, Audi, Red, 1990, Sports)

$V_{NB}(+) = \frac{4}{7}\left(\frac{4}{4}\right)(0)(0)\left(\frac{1}{4}\right)(0) = 0$

$V_{NB}(-) = \frac{3}{7}\left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\left(\frac{2}{3}\right)(0)\left(\frac{1}{3}\right) = 0$

$V_{NB}(+) = V_{NB}(-) = 0$

So we can choose arbitrarily either + or −

01/02/2024

most imp ★ Bayesian Belief Network.

Consider a set of Random variables $Y_1, Y_2 \ldots Y_n$ where each variable Y_i can taken on a set of possible values $V(Y_i)$ we define the joint space of the set of variables Y, ~~as the~~ to be the cross product of $V(Y_1) \times V(Y_2) \ldots \times V(Y_n)$
The probability distribution over this joint Space is joint " "

~~A~~ Bayesian Belief network describes the joint probability distribution for the set of variables.

Conditional Independence.

Let x y z be three discrete valued random variables. we say that x is conditionally independent y given z, if the probability distribution learning x is independent of the value of y given a value for z, i.e if (for all) ~~$\forall x_i y_j k$~~ x_i, y_j, z_k . if $P(X=x_i / Y=y_j / Z=z_k)$ then $P(X=x_i / Z=z_k)$ where $x_i \in V(x)$, $y_j \in V(Y)$, $z_k \in V(z)$.

We can rewrite this expression as

$$P(X/Y, Z) = P(X/Z)$$

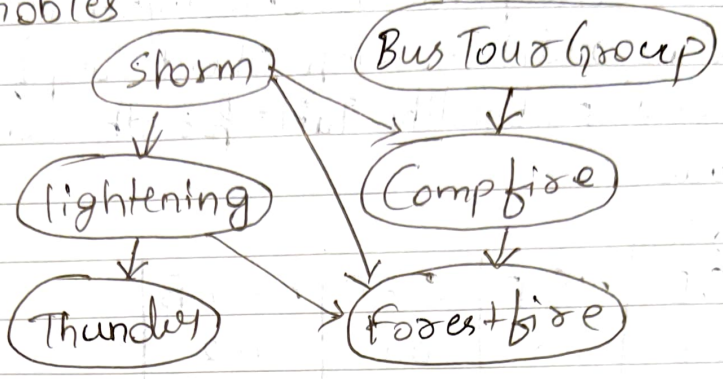
This can also be extended to a set of variables

$$P(X_1 X_2 \ldots X_m / Y_1, Y_2 \ldots Y_n, Z_1, Z_2 \ldots Z_l) =$$

$$P(X_1 \ldots X_m / Z_1 \ldots Z_l)$$

Representation of Bayesian Belief Network.

BBN represents the joint PD for a set of variables



The Bayesian n/w in the figure represents the joint probability Dist over the Boolean Variables, storm, lightining thunder, BusTourGroup Campfire and forestfire.

Each variable in the joint space ~~be~~ is represented by a node in the network. For each variable 2 types of information are specified.

① -> The network arcs represent that the variable is conditionally independent of its non-descendent in the n/w, Given its immediate predecessor in the n/w. we say that x is a descendent of y, if there is a directed path or arc from y to x.

② -> A Condi^{n} Prob table is given for each variable describing the PD for that variable Given the values of its immediate predecessor

2/2/24 In this fig consider the node campfire the n/w nodes and arcs represent the campfire is conditionally independent of its non-descendents lighting and thunder, given its immediate parents

C = campfire
P = Probability
S = storm
B = Bus tour Group

Bus Tour Group

or predecessors storm and BTG. this means that once we know the values of variables storm and BusTourGroup, the variable lightning and thunder provide no additional info about campfire. The conditional probability Table for the campfire node is as follows.

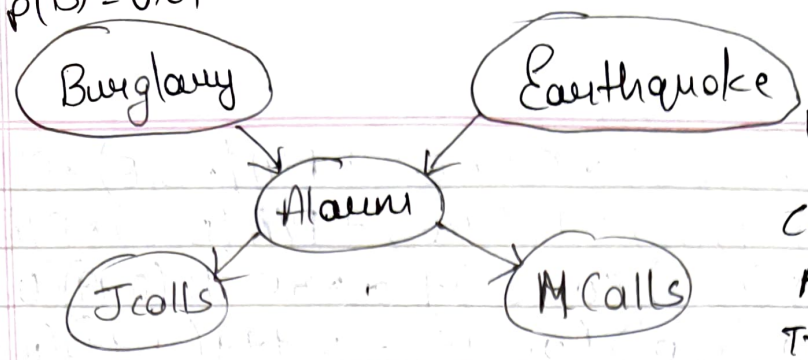
Campfire

	S, B	~S, B	S, ~B	~S, ~B
C	0.6	0.5	0.1	0.2
~C	0.4	0.5	0.9	0.8

- The top left entry expresses the assertion that the prob of (P (C = True / S = True , B = True))
- The set of conditional prob tables for all the variables together with the set of conditional independence assumptions described by the network describes the full joint probability distribution for the network.

Example-2 There is a burglar alarm installed at home. It is fairly reliable at detecting burglary but also sometime it responds to minor earthquakes. You have 2 neighbours J & M who have promised to call you at work when they hear the alarm, J always calls when he hears the alarm but sometimes confuses telephone ringing with the alarm and calls too. M likes loud music and sometimes misses the alarm.

a) what is the prob that the alarm has sounded but neither burglary nor earthquake has occured by both J and M calls.



CPT for M calls

A	P(M/A)	P(~M/A)
True	0.70	0.30
false	0.01	0.99

(CPT) Conditional prob table for alarm.

B	E	P(A/B,E)	P(~A/B,E)
T	T	0.95	0.05
T	f	0.94	0.06
f	T	0.29	0.71
f	f	0.001	0.999

CPT for J calls

A	P(J/A)	P(~J/A)
True	0.90	0.1
false	0.05	0.95

Joint Prob Distribution

$P(J \cap M \cap A \cap \sim B \cap \sim E)$

$= P(J/A) \cdot P(M/A) \cdot P(A/\sim B, \sim E) \cdot P(\sim B) \cdot P(\sim E)$

$= 0.90 \times 0.70 \times 0.001 \times 0.998$

$= 0.0006224$

05/2/2024

$P(J) = P(J/A) * P(A) + P(J/\sim A) * P(\sim A)$

$= 0.90 * P(A) + (0.05) * P(\bar{A})$

$P(A) = P(A/B,E) * P(B \cap E)$

$+ P(A/\sim B, E) * P(\sim B \cap E)$

$+ P(A/B, \sim E) * P(B \cap \sim E)$

$+ P(A/\bar{B}, \bar{E}) * P(\bar{B} \cap \bar{E})$

$P(A) = 0.95 * P(B) * P(E) + 0.29 * P(\bar{B}) * P(E) +$

$0.94 * P(B) \cdot P(\bar{E}) + 0.001 * P(\bar{B}) \cdot P(\bar{E})$

$= 0.95(0.01)(0.002) + 0.29(0.99)(0.002) + 0.94$

$(0.01)(0.998) + 0.001(0.99)(0.998)$

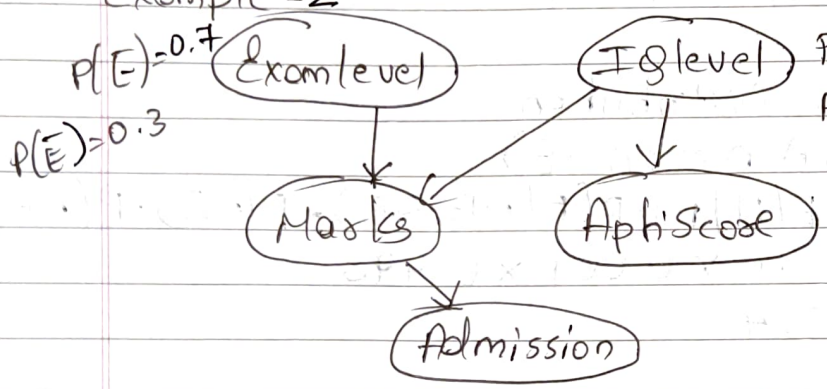
$P(A) = 0.0109$

Because $P(B \cap E)$ is independent $P(B) \cdot P(E)$

$P(\bar{A}) = P(\bar{A}/B,E) * P(B \cap E) + P(\bar{A}/\sim B,E) * P(\bar{B} \cap E)$
$+ P(\bar{A}/B,\bar{E}) * P(B \cap \bar{E}) + P(\bar{A}/\bar{B},\bar{E}) * P(\bar{B} \cap \bar{E})$
$= (0.05)(0.01)(0.002) + 0.71(0.99)(0.002)$
$+ (0.06)(0.01)(0.998) + 0.999(0.99)(0.998)$
$= 0.9890$ or $1 - 0.0109$

$P(J) = 0.90(0.0109) + 0.05(0.9890)$
$= 0.0592$

Example -2



IQ	P(AS/IQ)	P(Ā/IQ)
0	0.75	0.25
1	0.4	0.6

IQ	E	P(M/IQ,E)	
0	0	0.6	0.4
0	1	0.9	0.1
1	0	0.5	0.5
1	1	0.8	0.2

M	P(A/M)	P(Ā/M)
0	0.6	0.4
1	0.9	0.1

→ Calculate the probability that inspite of the exam level being difficult the student having a low IQ level and a low aptiscore manages to pass the exam and secure admission in the university

Exam level ← Easy - 0, Difficult - 1

IQ level ← low - 0, high - 1

Aptiscore ← low - 0, high - 1

marks ← fail - 0, pass - 1

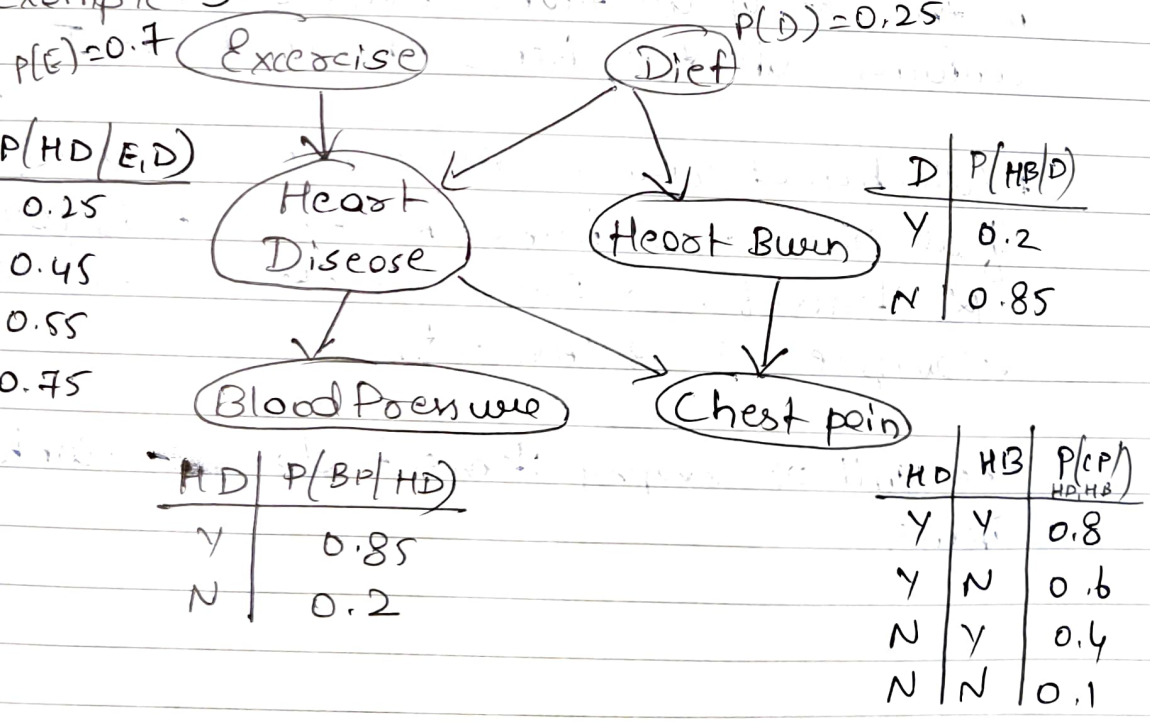
P(E) = Difficult Exam level
P(IQ) = low IQ level.

Admission { Secure/No = 0
Yes = 1

P(a=1, m=1, i=0, e=1, s=0)
P(a=1/m=1) * P(m=1/e=1, i=0) * P(i=0) * P(e=1) * P(s=0/i=0)
= (0.9)(0.9)(0.8)(0.7)(0.75)
= 0.3402

→ Calculate the probability that the student has a high IQ level & Apti score, the exam being easy yet fails to pass and doesn't secure admission.
P(a=0, m=0, i=1, e=0, s=1)
P(a=0/m=0) * P(m=0/e=0, i=1) * P(i=1) * P(e=0) * P(s=1/i=1)
= 0.6 * 0.5 * 0.2 * 0.3 * 0.4
= 0.0072

Example - 3



E	D	P(HD/E,D)
Y	Y	0.25
Y	N	0.45
N	Y	0.55
N	N	0.75

D	P(HB/D)
Y	0.2
N	0.85

HD	P(BP/HD)
Y	0.85
N	0.2

HD	HB	P(CP/HD,HB)
Y	Y	0.8
Y	N	0.6
N	Y	0.4
N	N	0.1

What is the Prob that the Person is having high BP
not solved.

06/02/2024

Association Analysis.

Methodology to identify interesting patterns or relationships.

Market Basket Transaction.

collec^n of items

Tid	Bread	Butter	Jam	Milk	P_{ro}
1	0	1	1	0	1
2					
3					

2 issues:-
→ computationally expensive
→ result in spurious relationships. (by chance)

$T = \{t_1, t_2 \dots t_n\}$. $I = \{i_1, i_2 \dots i_m\}$

Itemset: collection of items.

Let $I = \{i_1, i_2, i_3 \dots i_d\}$ be the set of all items in the market basket data. Let $T = \{t_1, t_2, t_3 \dots t_n\}$ be the set of all transaction. Each Transaction T_i contains subset of items chosen from I.

A collection of zero or more items is called itemset. If an itemset contains 'k' items it is called k itemset. {Bread, Butter, Jam} is an example of 3 itemset.

The Null or empty itemset doesn't contain any items.

Transaction width

It is the number of items present in the transaction. A Transaction t_i is said to contain an itemset x, if x is a subset of d_i.

An important property of an itemset is called support count.

(Support count is the no. of transaction that contain a particular itemset x. It is represented as $\sigma(x)$, support is called as "support count" of x

Mathematical Representation.

$$\sigma(x) = |\{t_i | x \subseteq t_i, t_i \in T\}|$$

Association Rule.

It is an implication of form $X \rightarrow Y$ where X & Y are disjoint itemset i.e $X \cap Y = \phi$

Two important measure) ×

-> Support - Always +ve

Transaction

Support Count

No of transaction that contain a particular itemset x, it is represented as $\sigma(x)$

ex: $X = \{Bread, Butter\}$

Tid	Bread	Butter	Jam	Support count $\{t_i\}$
t_1	1	1	0	
t_2	1	0	0	

Represented by σ_X where X is itemset.

$$\sigma_X = |\{t_i | x \subseteq t_i, t_i \in T\}|$$

Association Rule

we write in the form of implication ($X \rightarrow Y$)

Ex:- {Bread, Butter} → {Rice}

If I by these ↗ then I will this ↘

It is a implication expression of form $X \rightarrow Y$ where X and Y are disjoint itemset i.e $X \cap Y = \phi$

The strength of an association rule can be measured in terms of support(s) and confidence (c)

Support: $s(X \rightarrow Y) = \frac{T(X \cup Y)}{N}$ where N is no of transaction.

Confidence: $c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$

formulation of Association Rule Mining Problem or Association Rule Discovery.

strength of rule is found by s&c.

Brute force // Identify Given a set of Transec"

T => set of rules // support ≥ min support threshold } => strong rules

confidence ≥ min confidence } which obeys both

2 tasks:-

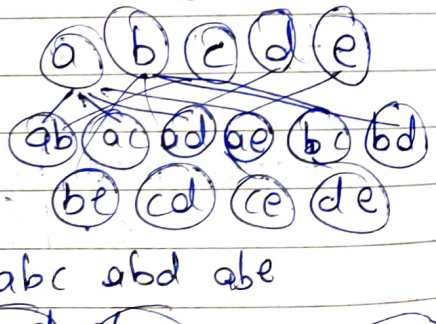
① -> Frequent Itemset Generations } => "In terms minus"

② -> Rule Generations

① Item sets I = {a, b, c, d, e}

calculate only support Lattice structure

{objective:- To identify strong rules}



$2^K - 1$ itemsets.

T.no			
1	a b c	a→3	ab = 2
2	a b	b→3	ac = 2
3	c d e	c→3	
4	d e	d→3	
5	a c d	e→3	
6	b, e		

* Reduce the complexity of ① 2 types
* Reduce the no. of itemset
* Reduce the no. of comparison.

we use Apriopri Principle to Reduce the no. of itemset to calculate the support.

Apriori Principle

states that if a particular itemset is found to be frequent then all its subset are also said to be frequent. Conversly if an itemset is to be infrequent, all its superset is said to be infrequent.

① Given set of items, identify the itemset that satisfy the min supp are frequent itemsets.

9/02/24

② Rule Generation

One frequent itemset are identified, next is to generate Rule Generation. (Only strong rule are) Here itemset should satisfy min confidence conf > minconf. Finally only strong rule are selected.

* find the Frequent itemset and generate Association Rule for the Given Problem.

I = {A, B, C}

A, B, C, AB, BC, CA, ABC

I = {A, B, C}

A → BC
B → AC
C → AB
AB → C
AC → B
BC → A

5.5 (6)

minsupp = 50%. $\frac{50}{100} \times 5$

$T = \{T_1, T_2, T_3, T_4, T_5\}$ minsupp = 2

$I = \{i_1, i_2, i_3, i_4, i_5, i_6\}$ minconf = 75%

Tid	Transactions
T_1	i_1, i_3, i_4
T_2	i_2, i_3, i_5, i_6
T_3	i_1, i_2, i_3, i_5
T_4	i_2, i_5
T_5	i_1, i_3, i_5

To find strong rules

→ Find out frequent itemsets and support count.

1-itemsets	Support count
i_1	$i_1 = 3$
i_2	$i_2 = 3$
i_3	$i_3 = 4$
i_4	$i_4 = 1$ ✗
i_5	$i_5 = 4$
i_6	$i_6 = 1$ ✗

→ Consider 2 itemsets and find out frequent itemsets but using apriori principle where it is says some itemset are infrequent its superset are also infrequent. but i_4 & i_6 are infrequent cause support count < minsupport.

2-itemsets	Support count
$i_1\ i_2$	1 ✗
$i_1\ i_3$	3
$i_1\ i_5$	2
$i_2\ i_3$	2
$i_2\ i_5$	3
$i_3\ i_5$	3

$\sigma(X)$

3-Itemsets	Supp count
$I_1\ I_3\ I_5$	2
$I_2\ I_3\ I_5$	2

4-itemset	Supp count
$I_1\ I_2\ I_3\ I_5$	1 ✗

→ satisfy min supp condition.

Frequent itemsets	Supp count
I_1	3
I_2	3
I_3	4
I_5	4
$I_1\ I_3$	3
$I_1\ I_5$	2
$I_2\ I_3$	2
$I_2\ I_5$	3
$I_3\ I_5$	3
$I_1\ I_3\ I_5$	2
$I_2\ I_3\ I_5$	2

Rule: $X \rightarrow Y$
$X \cap Y = \phi$

Rule Generation is based on frequent itemset Generation.

1-itemset we will leave it cause we cannot write X→Y format.

Association Rules	Confidence, $C(X \rightarrow Y) = \frac{\sigma(XUY)}{\sigma(X)}$
$I_1 \rightarrow I_3$	3/3 = 1 = 100%
$I_3 \rightarrow I_1$	3/4 = 75%
$I_1 \rightarrow I_5$	2/3 = 66% ✗
$I_5 \rightarrow I_1$	2/4 = 50% ✗
$I_2 \rightarrow I_3$	2/3 = 66% ✗
$I_3 \rightarrow I_2$	2/4 = 50% ✗
$I_2 \rightarrow I_5$	3/3 = 100%
$I_5 \rightarrow I_2$	3/4 = 75%
$I_3 \rightarrow I_5$	3/4 = 75%
$I_5 \rightarrow I_3$	3/4 = 75%
$I_1 \rightarrow I_3, I_5$	2/3 = 66% ✗
$I_3 \rightarrow I_1, I_5$	2/4 = 50% ✗

$i5 \rightarrow i1, i3$ 2/4 = 50% finally we have to write down all strong rules. which is not crossed.

$i1, i3 \rightarrow i5$ 2/3 = 66% ×

$i1\ i5 \rightarrow i3$ 2/2 = 100%

$i3\ i5 \rightarrow i1$ 2/3 = 66% ×

$i2 \rightarrow i3, i5$ 2/3 = 66% ×

$i3 \rightarrow i2, i5$ 2/4 = 50% ×

$i5 \rightarrow i2, i3$ 2/4 = 50% ×

$i2, i3 \rightarrow i5$ 2/2 = 100%

$i2, i5 \rightarrow i3$ 2/3 = 66% ×

$i3, i5 \rightarrow i2$ 2/3 = 66% ×

II IA.

12/2/24.

2) $I = \{A, B, C, D, E\}$ $T = \{T_1\ T_2\ T_3\ T_4\ T_5\ T_6\ T_7\ T_8\ T_9\}$

T_1	A, B
T_2	B, D
T_3	B, C
T_4	A, B, D
T_5	A, C
T_6	B, C
T_7	A, C
T_8	A, B, C, E
T_9	A, B, C

min sup = 2
min conf = 50%

9 min support is given in percentage

Ex:- 60%

min support = $\frac{60}{100} \times 9$

$\frac{54}{10} = 5.4$ ↓ no. of transaction

1 - itemset		support count
T_1	A	6
T_2	B	7
T_3	C	6
T_4	D	2
T_5	E	1 ×
T_6		
T_7		
T_8		
T_9		

2-itemset	SC.		3-itemset	supp
AB	4		ABC	2
AC	4			
AD	1	x	4-itemset	
BC	4			
BD	2			
CD	0	x		

Frequent itemset	Supp C
A	6
B	7
C	6
D	2
AB	4
AC	4
BC	4
BD	2
ABC	2

Association Rules.	Confidence.		
A→B	4/6 = 66.67%	D→B	2/2 = 100%
B→A	4/7 = 57.14%		
A→C.	4/6 = 66.67		
C→A	4/6 = 66.67		
B→C	4/7 = 57.14%		
C→B	4/6 = 66.67		
B→D	2/7 = 28.57 x		
A→B,C	2/6 = 33% x		
B→A,C	2/7 = 28 x		
C→A,B	2/6 = 33 x		
AB→C	2/4 = 50%		
BC→A	2/4 = 50%		
AC→B	2/4 = 50%		

min conf = 60%

T_1	Bread, Buns, Sauce	min sup = 33.33%
T_2	Bread, Buns	$\frac{33.33}{100} \times 6 = 1.99$
T_3	Bread Coke, Chips	$= 2$
T_4	Chips, Coke	
T_5	Chips, Sauce	
T_6	Bread, Coke, Chips	

1-itemset	Supp C
Bread	4
Buns	2
Sauce	2
Coke	3
Chips	4

2-itemset	Supp C	
Br, Bu	2	
Br, Sa	1	X
Br, Co	2	
Br, Ch	2	
Bu, Sa	1	X
Bu, Co	0	X
Bu, Ch	0	X
Sa, Co	0	X
Sa, Ch	0	X
Co, Ch	3	

3 itemset	Supp C
Br, Co, Ch	2

frequent items	Supp C
Br	4
Bu	2
Sa	2
Co	3
Ch	4
Br, Bu	2
Br, Co	2
Br, Ch	2
Co, Ch	3
Br, Co, Ch	2

Association Rules

Br → Bu	2/4 = 50% X	~~Ch → Br~~	~~2/4 = 50% X~~
Bu → Br	2/2 = 100%	Co → Ch	3/3 = 100%
Br → Co	2/4 = 50% X	Ch → Co	3/4 = 75%
Co → Br	2/3 = 66%	Br → Co Ch	2/4 = 50% X
Ch → Br	2/4 = 50% X	Co → Br Ch	2/3 = 66%
Br → Ch	2/4 = 50% X	Ch → Br, Co	2/4 = 50% X

Co Ch → Br 2/3 = 66.7%
Br Ch → Co 2/2 = 100%
Br Co → Ch 2/3 = 66.7%

13/2/24

N = No. of transactions.
F = Frequent itemsets.
K = no. of items in itemsets.

Apriori Algorithm.

1. K = 1
2. $F_K = \{i \mid i \in I \wedge \sigma(\{i\}) > N \times \text{minsupp}\}$ // Generate all itemsets. K = 1, 2, 3
3. Repeat.
4. K = K + 1
5. C_K = Candidate generation (F_{K-1})
6. C_K = Candidate Prune (C_K, F_{K-1})
7. for each transaction $t \in T$ do
8. C_K = subset (C_K, t)
9. for each candidate itemset $c \in C_K$ do
10. $\sigma(c) = \sigma(c) + 1$
11. end for
12. endfor
13. $F_K = \{c \mid c \in C_K \wedge \sigma(c) > N \times \text{minisupport}\}$
14. until $F_K = \phi$
15. Result $\cup F_K$

∪ = Union.

Generation of freq itemsets.

two imp process: → ① Candidate Generation
to generate freq Itemsets → methods
→ ② Candidate Pruning.

① putting every possibility of itemsets are Candidate Itemset.
All Cand. I. may be or not frequent Itemset.

Candidate Itemsets.

1-Itemset

A	3	— f I
B	0	X
C	1	X
D	4	— FI
G	3	— f I

this C_K of K = 2 is given F_1.

③ pruning is done based on support pruning.

① Candidate Generation

Candidate Generation Procedure < Complete
                                < Non-Redundent.

it should follow both types to generate CG,I

Complete => it should not omit any possibilites.

Non-R => Any itemset should not occur more than once.

Diff ways we can generate CG,I itemsets.

1) Bruteforce method.

Candidate C
1-itemset 2-IS.
A AB, AC
B AD, AE
C BC, BD, BE
D CD, CE
E DE

Candidate.

* Disadvantage! - Pruning is expensive which takes lot of time.

K = level of itemsets.

2) $f_{k-1} \times f_1$ Itemsets.

Take Bread, Butter example (Refer last page) consider we have to generate 3 itemset. we can do this in $f_2 \times f_1$

f_2	f_1	3-itemset
Bread Bun	Bread	
Bread Coke	Buns	
Bread Chips	Sauce	→ Bread, Bun, Sauce
Chips Coke	Coke	Bread, Bun, Coke
	Chips	✓ Bread, Bun, Chips
	Redundency	✓ Bread Coke Sauce
		Bread Chips Buns

Dis:- Candidate itemset generated are not non-Redudor in 2)

* Items should be arranged in lexicographical. orde

Bread chips Coke Bun

hence we can remove Redundent itemsets.

eg: Bread chips combining with Bun we cannot proceed cause bun < chips in alphabetical order.

3) $f_{k-1} \times f_{k-1}$ to form 3-itemsets.

eg:- {Bread, Sauce} {Bread Jem} {Bread Milk}

{Jem, Milk, Sugar} {Jem, Milk}

1-item should match when f_{k-1} *

whenever we are considering this atleast f_{k-1} $k-2$, which 1st item should match.