1. What is the Naive Bayes Classifier? How to estimate the Probabilities from data? Give an Example of classification.

→ Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, $X = (x_1, x_2, x_3, \ldots, x_n)$ - depicting n measurements made on the tuple from n attributes vector, $A_1, A_2, A_3, \ldots, A_n$

→ Suppose that there are m classes, $C_1, C_2, C_3, \ldots, C_m$. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the Naive Bayesian classifier predicts that tuple X belongs to the class C_i if and only iff

$$P(C_i/X) > P(C_j/X) \quad \text{for } 1 \leq j \leq m, j \neq i$$

Thus, we maximize $P(C_i|X)$. The class C_i for which $P(C_i|X)$ is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

$$P(C_i|X) = \frac{P(X/C_i)P(C_i)}{P(X)}$$

Scanned with CamScanner

→ As $P(X)$ is constant for all classes, only $P(X/c_i)\,P(c_i)$ needs to be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(c_1) = P(c_2)$ ... and we would therefore maximize $P(X/c_i)$ ... $P(c_m)$ and

→ Given data sets with many attributes, it would be extremly computationally expensive to compute $P(X/c_i)$. To reduce computation in evaluating $P(X/c_i)$, the naive assumption of class-conditional independence

$$P(X/c_i) = \prod_{k=1}^{n} P(x_k/c_i)$$

$$= P(x_1/c_i) \times P(x_2/c_i) \times \ldots \times P(x_n/c_i)$$

a) If A_k is categorical, then $P(x_k/c_i)$ is the number of tuples of class c_i in D having the value x_k for A_k, divided by $|c_{i,D}|$, the number of tuples of class c_i in D.

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

So that

$$P(x_k/c_i) = g(x_k, \mu_{c_i}, \sigma_{c_i})$$

→ $P(X/c_i)\,P(c_i) > P(X/c_j)\,P(c_j)$ for $1 \le j \le m$, $j \ne i$ the predict class label is the class c_i for which P ...

Scanned with CamScanner

Ex:- Dataset

RID	age	Income	Student	cd rating	Class buys-compu
1	youth	high	no	fair	no
2	youth	high	no	Excellent	no
3	middle-aged	high	no	fair	yes
4	Senior	medium	no	fair	yes
5	Senior	~~low~~	yes	fair	yes
6	Senior	low	yes	Excellent	no
7	middle-aged	low	yes	Excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	Senior	medium	yes	fair	yes
11	youth	medium	yes	Excellent	yes
12	middle-aged	medium	no	Excellent	yes
13	middle-aged	high	yes	fair	yes
14	Senior	Medium	no	Excellent	no

→ The data tuples are described by the attributes age, income, student and Credit rating

→ The Class label attribute, buys-computer, has two distinct values (namely, {yes, no})

→ C_1 Correspond to the class buys-computer = yes and C_2 Correspond to buys-computer = no.

X = (age: youth, income= medium, Student = yes, Credit_rating = fair)

we need to maximize $P(X/C_i)\ P(C_i)$, for = 1,2

$P(C_i)$, the prior probability of each class, can be Computed based on the training tuples:

P(buys_Computer = yes) = 9/14 = 0.643

P(buys_Computer = No) = 5/14 = 0.357

To Compute $P(X/C_i)$, for i = 1,2, we Compute the following Conditional probabilities:

P(age = youth | buys_Computer = yes) = 2/9 = 0.222

P(age = youth | buys_Computer = No) = 3/5 = 0.600

P(income = Medium | buys_computer = yes) = 4/9 = 0.444

P(income = Medium | buys_computer = no) = 2/5 = 0.400

P(income = Medium | buys_computer = yes) = 6/9 = 0.667

P(Student = yes | buys_Computer = No) = 1/5 = 0.200

P(Student = yes | buys_Computer = yes) = 6/9 = 0.667

P(Credit_rating = fair | buys_Computer = yes) = 2/5 = 0.400

P(credit_rating = fair | buys_Computer = No) = 2/5 = 0.400

Using these probabilities, we obtain

P(X | buys_Computer = yes) = P(age = youth | buys_computer = yes)

× P(income = medium | buys_Computer = yes) × P(student = yes | buys_Computer = yes) × P(credit_rating = fair | buys_Computer = yes) = 0.222 × 0.444 × 0.667

= 0.044

Similarly,

$P(x \mid buys\_Computer = no) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019.$

To find the class, C_i, that maximizes $P(X \mid C_i)\, P(C_i)$, we compute

$P(x \mid buys\_Computer = yes)\, P(buys\_Computer = yes)$
$= 0.044 \times 0.643$
$= 0.028$

$P(x \mid buys\_Computer = No)\, P(buys\_Computer = no)$
$= 0.019 \times 0.357$
$= 0.007.$

∴ the naive Bayesian classifier predicts buys_Computer = yes for tuple x.