Bayesian Classifiers

Bayesian classifiers are statistical classifiers which predict the class membership probabilities of tuples. It means probability of a given tuple to be in a particular class. Bayesian classifiers are based on Bayes theorem which will be explained in next section. Studies comparing classification algorithms have found a simple Bayesian classifier known as the naive Bayesian classifier is to be comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases. In theory Bayesian classifiers have the minimum error rate in comparison to all other classifers. However, in practice this is not always the case owing to inaccuracies in the assumptions made for its use, such as class-conditional independence, and the lack of available probability data.[1]

Bayes Theorem

This theorem was named after Thomas Bayes who did early work in probability and decision theory during the 18th century. Let **X** a data tuple. In Bayesian terminology, **X** is termed as 'evidence'. **X** is described by n attributes. Ie, **X={x1,x2,x3,x4.....xn}.** Let H be some hypothesis that the data tuple X belongs to a particular class C. For classification problems we want to find P(H|X), the probability that hypothesis H holds when attributes set/'evidence' X is given. That means we are looking for the probability that the tuple X belongs to class C, given that we know the attribute description of X.

Bayes theorem helps us to determine the probability P(H/X) in terms of P(H),P(X/H) and P(X). Bayes Theorem is

$$P(H/X)=\frac{P(X/H)P(H)}{P(X)}$$

Where P(X/H) gives the probability for tuple to occur when it is given that the hypothesis holds. P(H) is the probability for the hypothesis H to hold. And P(X) is the probability for the attribute set X to occur.

Naive Bayesian Classification

Naive Bayes Classifiers are Bayesian Classifiers which assume *class conditional independence.* Ie, it assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption simplifies the calculations a lot.
Naive Bayesian classifier works as follows.

1. Let D be a training set of tuples and their associated class labels. Each tuple is represented by n dimensional attribute vector in the form (x1,x2,x3....xn) depicting n measurements made on the tuple from n attributes, respectively, A1, A2,...An

2. Suppose that there are m classes, say C1,C2...,Cm. Given a test tuple X , the classifier has to predict the category of X. X will be belonging to the class with highest value for P(H/X)
   ie X will be belonging to a class Ci, if and only if
   P(Ci/X) > P(Cj/X) for 1 <= j <= m and i!=j
   That means we have to find the class for which P(Ci/X) is maximum, where P(Ci/X) is given as
   $$P(Ci/X)=\frac{P(X/Ci)P(Ci)}{P(X)}$$

3. As P(X) will be constant for all classes, we have to consider numerator term only. If we are taking equal number of training tuples for each class, then P(Ci) will be same for all classes. In

such cases we have to maximize P(X/Ci) only. Otherwise we have to maximize P(X/Ci)P(Ci). Where P(Ci) can be calculated as follows

$$P(Ci)= \frac{\text{Number of training tuples belonging to class Ci}}{\text{Total number of training tuples}}$$

4. When we have data sets with several number attributes, it is computationally expensive to calculated P(X/Ci). But when we assume *class conditional independence*, computational cost can be reduced significantly. With this assumption P(X/Ci) can be calculated as follows.

$$P(X/Ci)=\prod_{k=1}^{n} P(xk/Ci)=P(x1/Ci)\,xP(x2/Ci)\,x....xP(xn/Ci)$$

- If attribute Ak is categorical,then P(xk/Ci) is given as follows

$$P(xk/Ci)=\frac{\text{Number of tuples in class Ci having value xk for attribute Ak}}{\text{Total number of tuples in class Ci}}$$

- If attribute xk is continuous valued, then the following function can be used for finding P(xk/Ci)

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

so that

$$P(xk/Ci)=g(xk,\mu Ci,\sigma Ci)$$

5. To predict the class label of X, P(X/Ci)P(Ci) is evaluated for each class Ci. The classifier predicts that the class lable of tuple X is the class Ci if and only if

P(Ci/X) > P(Cj/X) for 1 <= j <= m and i!=j

Modification on Naive Bayes for document classification

For document classification problems calculation of P(xk/Ci) will be modified. It is required since attributes are not either categorical or continuous valued in case of document classification problems. In document classification problem we have the relation,

$$P(xk/Ci)\propto P(Ci)\prod_{1\le k\le nd} P(tk/Ci)$$

where nd is number of tuples in class Ci and total number of occurances of word xk in category Ci.

P(tk/Ci) can be calculated as follows

$$P(tk/Ci)=\frac{Tci}{\sum_{t'\in V} Tct'}$$

where Tci is number of occurences of word tk in category Ci. V is the vocabular, all words in training set.

Laplacian correction to avoid zero computational value

Inorder to avoid the zero probability value , when a word in testing tuple donot come in training set, we use Laplacian correction. Laplacian corrected formula is

$$P(tk/Ci) = \frac{Tci+1}{\sum_{t' \in V} (Tct'+1)} = \frac{Tci+1}{\sum_{t' \in V} Tct' + |V|}$$

where V is number of terms in vocabulary.

References

1. Data Mining Concepts and Techniques - Jiawei Han,Micheline Kamber, Jian Pei 3rd edition