POLYTECHNIC UNIVERSITY
Department of Computer Science / Finance and Risk Engineering

Naive Bayesian Classifier

K. Ming Leung

Abstract: A statistical classifier called Naive Bayesian classifier is discussed. This classifier is based on the Bayes' Theorem and the maximum posteriori hypothesis. The naive assumption of class conditional independence is often made to reduce the computational cost.

Directory

- Table of Contents
- Begin Article

Last Revision Date: November 28, 2007

Table of Contents

1. Introduction

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given sample belongs to a particular class.

Bayesian classifier is based on Bayes' theorem. Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computation involved and, in this sense, is considered "naive".

1.1. Bayes' Theorem

Let $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ be a sample, whose components represent values made on a set of n attributes. In Bayesian terms, $\mathbf{X}$ is considered "evidence". Let H be some hypothesis, such as that the data $\mathbf{X}$ belongs to a specific class C. For classification problems, our goal is to determine $P(H|\mathbf{X})$, the probability that the hypothesis H holds given the "evidence", (*i.e.* the observed data sample $\mathbf{X}$). In other words, we are looking for the probability that sample $\mathbf{X}$ belongs to class C, given that we know the attribute description of $\mathbf{X}$.

$P(H|\mathbf{X})$ is the a posteriori probability of H conditioned on $\mathbf{X}$. Fox example, suppose our data samples have attributes: *age* and *income*, and that sample $\mathbf{X}$ is a 35-year-old customer with an income of \$40,000. Suppose that H is the hypothesis that our customer will buy a computer. Then $P(H|\mathbf{X})$ is the probability that customer $\mathbf{X}$ will buy a computer given that we know the customer's age and income.

In contrast, $P(H)$ is the a priori probability of H. For our example, this is the probability that any given customer will buy a computer, regardless of age, income, or ny other information. The a posteriori probability $P(H|\mathbf{X})$ is based on more information (about the customer) than the a priori probability, $P(H)$, which is independent of $\mathbf{X}$.

Similarly, $P(\mathbf{X}|H)$ is the a posteriori probability of $\mathbf{X}$ conditioned on$H$. That is, it is the probability that a customer $\mathbf{X}$, is 35 years old and earns \$40,000, given that we know the customer will buy a computer.

$P(\mathbf{X})$ is the a priori probability of $\mathbf{X}$. In our example, it is the probability that a person from our set of customers is 35 years old and earns $40,000.

According to Bayes' theorem, the probability that we want to compute $P(H|\mathbf{X})$ can be expressed in terms of probabilities $P(H)$, $P(\mathbf{X}|H)$, and $P(\mathbf{X})$ as

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)\ P(H)}{P(\mathbf{X})},$$

and these probabilities may be estimated from the given data.

2. Naive Bayesian Classifier

The naive Bayesian classifier works as follows:

1. Let T be a training set of samples, each with their class labels. There are k classes, $C_1, C_2, \ldots, C_k$. Each sample is represented by an n-dimensional vector, $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$, depicting n measured values of the n attributes, $A_1, A_2, \ldots, A_n$, respectively.

2. Given a sample $\mathbf{X}$, the classifier will predict that $\mathbf{X}$ belongs to the class having the highest a posteriori probability, conditioned

on $\mathbf{X}$. That is $\mathbf{X}$ is predicted to belong to the class C_i if and only if

$$P(C_i|\mathbf{X}) > P(C_j|\mathbf{X}) \qquad \text{for } 1 \leq j \leq m,\ j \neq i.$$

Thus we find the class that maximizes $P(C_i|\mathbf{X})$. The class C_i for which $P(C_i|\mathbf{X})$ is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

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

3. As $P(\mathbf{X})$ is the same for all classes, only $P(\mathbf{X}|C_i)P(C_i)$ need be maximized. If the class a priori probabilities, $P(C_i)$, are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1) = P(C_2) = \ldots = P(C_k)$, and we would therefore maximize $P(\mathbf{X}|C_i)$. Otherwise we maximize $P(\mathbf{X}|C_i)P(C_i)$. Note that the class a priori probabilities may be estimated by $P(C_i) = \text{freq}(C_i, T)/|T|$.

4. Given data sets with many attributes, it would be computationally expensive to compute $P(\mathbf{X}|C_i)$. In order to reduce compu-

tation in evaluating $P(\mathbf{X}|C_i)\ P(C_i)$, the naive assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the sample. Mathematically this means that

$$P(\mathbf{X}|C_i) \approx \prod_{k=1}^{n} P(x_k|C_i).$$

The probabilities $P(x_1|C_i), P(x_2|C_i), \ldots, P(x_n|C_i)$ can easily be estimated from the training set. Recall that here x_k refers to the value of attribute A_k for sample $\mathbf{X}$.

(a) If A_k is categorical, then $P(x_k|C_i)$ is the number of samples of class C_i in T having the value x_k for attribute A_k, divided by freq(C_i, T), the number of sample of class C_i in T.

(b) If A_k is continuous-valued, then we typically assume that the values have a Gaussian distribution with a mean μ and standard deviation σ defined by

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

so that

$$p(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

We need to compute μ_{C_i} and σ_{C_i}, which are the mean and standard deviation of values of attribute A_k for training samples of class C_i.

5. In order to predict the class label of $\mathbf{X}$, $P(\mathbf{X}|C_i)P(C_i)$ is evaluated for each class C_i. The classifier predicts that the class label of $\mathbf{X}$ is C_i if and only if it is the class that maximizes $P(\mathbf{X}|C_i)P(C_i)$.

3. Example: Using the Naive Bayesian Classifier

We will consider the following training set.

RID	age	income	student	credit	C_i: buy
1	youth	high	no	fair	C_2: no
2	youth	high	no	excellent	C_2: no
3	middle-aged	high	no	fair	C_1: yes
4	senior	medium	no	fair	C_1: yes
5	senior	low	yes	fair	C_1: yes
6	senior	low	yes	excellent	C_2: no
7	middle-aged	low	yes	excellent	C_1: yes
8	youth	medium	no	fair	C_2: no
9	youth	low	yes	fair	C_1: yes
10	senior	medium	yes	fair	C_1: yes
11	youth	medium	yes	excellent	C_1: yes
12	middle-aged	medium	no	excellent	C_1: yes
13	middle-aged	high	yes	fair	C_1: yes
14	senior	medium	no	excellent	C_2: no

The data samples are described by attributes *age*, *income*, *student*, and *credit*. The class label attribute, *buy*, tells whether the person buys a computer, has two distinct values, *yes* (class C_1) and *no* (class

C_2).

The sample we wish to classify is

$$\mathbf{X} = (age = youth, income = medium, student = yes, credit = fair)$$

We need to maximize $P(\mathbf{X}|C_i)P(C_i)$, for $i = 1, 2$. $P(C_i)$, the a priori probability of each class, can be estimated based on the training samples:

$$P(buy = yes) = \frac{9}{14}$$

$$P(buy = no) = \frac{5}{14}$$

To compute $P(\mathbf{X}|C_i)$, for $i = 1, 2$, we compute the following conditional probabilities:

$$P(age = youth|buy = yes) = \frac{2}{9}$$

$$P(age = youth|buy = no) = \frac{3}{5}$$

$$P(income = medium|buy = yes) = \frac{4}{9}$$

$$P(income = medium|buy = no) = \frac{2}{5}$$

$$P(student = yes|buy = yes) = \frac{6}{9}$$

$$P(student = yes|buy = no) = \frac{1}{5}$$

$$P(credit = fair|buy = yes) = \frac{6}{9}$$

$$P(credit = fair|buy = no) = \frac{2}{5}$$

Using the above probabilities, we obtain

$$\begin{aligned} P(\mathbf{X}|buy = yes) &= P(age = youth|buy = yes) \\ &\quad P(income = medium|buy = yes) \\ &\quad P(student = yes|buy = yes) \\ &\quad P(credit = fair|buy = yes) \\ &= \frac{2}{9}\frac{4}{9}\frac{6}{9}\frac{6}{9} = 0.044. \end{aligned}$$

Similarly,

$$P(\mathbf{X}|buy = no) = \frac{3}{5}\frac{2}{5}\frac{1}{5}\frac{2}{5} = 0.019$$

To find the class that maximizes $P(\mathbf{X}|C_i)P(C_i)$, we compute

$$P(\mathbf{X}|buy = yes)P(buy = yes) = 0.028$$

$$P(\mathbf{X}|buy = no)P(buy = no) = 0.007$$

Thus the naive Bayesian classifier predicts $buy = yes$ for sample $\mathbf{X}$.

3.1. Laplacian Correction

The Laplacian correction (or Laplace estimator) is a way of dealing with zero probability values.

Recall that we use the estimation

$$P(\mathbf{X}|C_i) \approx \prod_{k=1}^{n} P(x_k|C_i).$$

based on the class independence assumption. What if there is a class, C_i, and $\mathbf{X}$ has an attribute value, x_k, such that none of the samples in

C_i has that attribute value? In that case $P(x_k|C_i) = 0$, which results in $P(\mathbf{X}|C_i) = 0$ even though $P(x_k|C_i)$ for all the other attributes in $\mathbf{X}$ may be large.

In our example, for the attribute-value pair $student = yes$ of $\mathbf{X}$, we need to count the number of customers who are students, and for which $buy = yes$ (which contributes to $P(\mathbf{X}|buy = yes)$) and the number of customers who are students and for which $buy = no$ (which contributes to $P(\mathbf{X}|buy = no)$). But what if , say, there are no training samples representing students for the class $buy = no$ resulting in $P(\mathbf{X}|buy = no) = 0$? A zero probability cancels the effects of all of the other a posteriori probabilities on C_i.

There is a simple trick to avoid this problem.We can assume that our training set is so large that adding one to each count that we need would only make a negligible difference in the estimated probabilities, yet would avoid the case of zero probability values. This technique is know as Laplacian correction (or Laplace estimator). If we have q counts to which we each add one, then we must remember to add q to the corresponding denominator used in the probability calculation.

As an example, suppose that for the class $buy = yes$ in some training set, containing $1,000$ samples, we have 0 with $income = low$, 990 samples with $income = medium$, and 10 samples with $income = high$. The probabilities of these events, without the Laplacian correction, are 0, 0.990 (from 990/1000), and 0.010 (from 10/1000), respectively. Using the Laplacian correction for the three quantities, we pretend that we have 1 more sample for each income-value pair. In this way, we instead obtain the following probabilities (rounded up to three decimal places):

$$\frac{1}{1003} = 0.001, \quad \frac{991}{1003} = 0.988, \quad \frac{11}{1003} = 0.011,$$

respectively. The "corrected" probability estimates are close to their "uncorrected" counterparts, yet the zero probability value is avoided.

4. Remarks on the Naive Bayesian Classifier

1. Studies comparing classification algorithms have found that the naive Bayesian classifier to be comparable in performance with decision tree and selected neural network classifiers.

2. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.

References

[1] M. Kantardzic, *Data Mining - Concepts, Models, Methods, and Algorithms*, IEEE Press, Wiley-Interscience, 2003, ISBN 0-471-22852-4.

[2] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*, Elsevier 2006, ISBN 1558609016. This part of the lecture notes is derived from chapter 6.4 of this book.