B

Bayesian Machine Learning

Eitel J.M. Lauría
Marist College, USA

INTRODUCTION

Bayesian methods provide a probabilistic approach to machine learning. The Bayesian framework allows us to make inferences from data using probability models for values we observe and about which we want to draw some hypotheses. Bayes theorem provides the means of calculating the probability of a hypothesis (posterior probability) based on its prior probability, the probability of the observations and the likelihood that the observational data fit the hypothesis.

$$\mathrm{P}(H \mid D) = \frac{\mathrm{P}(D \mid H) \cdot \mathrm{P}(H)}{\mathrm{P}(D)} \qquad (1)$$

$\mathrm{P}(H \mid D)$ is defined as the probability of a certain hypothesis based on a set of observational data given a certain context (posterior probability of hypothesis H); $\mathrm{P}(D \mid H)$ is the likelihood of the observations given a certain hypothesis; P(H) is the intrinsic probability of hypothesis H, before considering the evidence D (prior probability); and P(D) is the probability of the observations, independent of the hypothesis, that can be interpreted as a normalizing constant. Bayes rule can therefore be reformulated as shown in expression . This means that the probability of the hypothesis is being updated by the likelihood of the observed data.

$$\mathrm{P}(H \mid D) \propto \mathrm{P}(D \mid H) \cdot \mathrm{P}(H) \qquad (2)$$

BACKGROUND

The practical application of Bayes rule to machine learning is rather straightforward. Given a set of hypotheses H and a set D of observational data we can estimate the most probable hypothesis H given D, by comparing different instances of the previous expression for each hypothesis H and choosing the one that holds the largest posterior probability (also called maximum a posteriori probability or MAP).

$$\text{Most probable } H \equiv H_{\mathrm{MAP}} = \arg\max \left[\mathrm{P}(D \mid H) \cdot \mathrm{P}(H) \right] \qquad (3)$$

Suppose we have a classification problem where the class variable is denoted by C and can take values c_1, c_2, ..., c_k. Consider a data sample D represented by n attributes $A_1, A_2, ..., A_n$ of which the observations ($a_1, a_2, ..., a_m$) have been taken for each instance of D. Suppose that each instance of the data sample D is classified as c_1, c_2, ..., c_k. The Bayesian approach to classifying a new instance would then be to assign the most probable target value (a class value of type c_i) by calculating the posterior probability for each class given the training data set, and from them choosing the one that holds the maximum a posteriori probability.

$$c_{\mathrm{MAP}} = \arg\max_{c_i \in C} \left[\mathrm{P}(D \mid c_i) \cdot \mathrm{P}(c_i) \right] \qquad (4)$$

NAIVE BAYES CLASSIFICATION

Although the idea of applying full-blown Bayesian criteria to analyze a hypothesis space in search of the most feasible hypothesis is conceptually attractive, it usually fails to deliver in practical settings. Although we can successfully estimate P(c_i) from the training data, calculating the joint probability $\mathrm{P}(D \mid c_i)$ is usually not feasible: unless we have a very large training data set, we would end up with estimates that are representative of a small fraction of the instance space and are therefore unreliable. The naive Bayesian classifier attempts to solve this problem by making the following assumptions:

- Conditional independence among attributes of the data sample. This means that the posterior probability of D, given c_i is equal to the product of the posterior probability of each attribute.

$$\mathrm{P}(D \mid c_i) = \prod_{j=1}^{n} \mathrm{P}(A_j = a_j \mid c_i) \quad , c_i \in C \qquad (5)$$

- The conditional probabilities of each individual attribute can be estimated from the frequency distributions of the sample data set D as N_{ij}/N_i, where N_{ij} is the number of training examples for which attribute $A_j = a_j$ and class value is c_i; and N_i is the number of training examples for which the class value is c_i. If the prior probabilities P(c_i) are not known, they can also be estimated by drawing its probabilities from the sample data set of frequency distributions.
- To solve the cases in which there are very few or no instances in the data set for which $A_j = a_j$ given a certain class value c_i, which would in turn render poor estimates of $\mathrm{P}(A_j = a_j \mid c_i)$ or make it equal to zero, a common approach is to estimate $\mathrm{P}(A_j = a_j \mid c_i)$ as:

$$\mathrm{P}(A_j = a_j \mid c_i) = \frac{N_{ij} + \alpha_{ij}}{N_i + \alpha_i} \qquad (6)$$

where α_{ij} and α_i can be seen as fictitious counts coming out of our prior estimate of the probability we wish to determine. In rigor, this implies considering a conjugate prior probability given by a Dirichlet distribution (for more details see Ramoni & Sebastiani, 1999). A typical method for choosing α_{ij} and α_i in the absence of other information is to assume uniform distribution of the counts, which means that if an attribute has r possible values, α_{ij} = 1 and $\alpha_i = r$. This results in:

$$\mathrm{P}(A_j = a_j \mid c_i) = \frac{N_{ij} + 1}{N_i + r} \qquad (7)$$

These assumptions have the effect of substantially reducing the number of distinct conditional probability terms that must be estimated from the training data. To illustrate the use of the naïve Bayes classifier, consider the example in Table 1 adapted from Mitchell (1997). We are dealing with records reporting on weather conditions for playing tennis. The task is to build a classifier that, by learning from previously collected data, is able to predict the chances of playing tennis based on new weather reports. We can estimate the class probabilities P(play=yes) and P(play=no) by calculating their frequency distributions as follows:

P(play=yes) = (# of instances were play=yes) / (total # of instances) = 9/14
P(play=no) = (# of instances were play=no) / (total # of instances) = 5/14

The conditional probabilities can be estimated by applying equation , as shown in Table 1(d). For a new weather report W={outlook=rain, temp=hot, Humidity=high, windy =false} the classifier would compute

Table 1. Weather data set

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	No

(a) List of Instances

Attribute Name	Values
outlook	sunny, overcast, rainy
temperature	hot, mild cool
humidity	high, normal
windy	true, false
play (class attribute)	yes, no

(b) List of Attributes

(play = yes)	9/14
(play = no)	5/14

(c) Prior Class Probabilities

	play				play	
Outlook	yes	no		Humidity	yes	no
sunny	3/12	4/8		high	4/11	5/7
overcast	5/12	1/8		normal	7/11	2/7
rain	4/12	3/8				
Temperature				Windy		
hot	3/12	3/8		false	4/11	4/7
mild	5/12	3/8		true	7/11	3/7
cold	4/12	2/8				

(d) Conditional Probabilities