Report for Machine learning

Assignment 1 - S5089493 - Raghuveer Siddaraboina

Introduction

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.An advantage of naive Bayes is that it only requires a small number of training data to estimate the parameters necessary for classification.Standard method for naive Bayes classifier in matlab is by using the predefined functions such as “predict” and “resubPredict”.

Understanding Bayes theorem

Bayes' theorem relies on incorporating prior probability distributions in order to generate posterior probabilities. Prior probability, in Bayesian statistical inference, is the probability of an event before new data is collected.

Posterior probability is the revised probability of an event occurring after taking into consideration new information. Posterior probability is calculated by updating the prior probability by using Bayes' theorem. In statistical terms, the posterior probability is the probability of event A occurring given that event B has occurred.

Bayes' theorem thus gives the probability of an event based on new information that is, or may be related, to that event. The formula can also be used to see how the probability of an event occurring is affected by hypothetical new information, supposing the new information will turn out to be true.

Formula For Bayes' Theorem

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

where:

$P(A) =$ The probability of A occurring

$P(B) =$ The probability of B occurring

$P(A|B) =$ The probability of A given B

$P(B|A) =$ The probability of B given A

$P\left(A \bigcap B\right)) =$ The probability of both A and B occurring

Approach for Naive Bayes algorithm.

Let's understand it using the training data set of weather and corresponding target variable 'Play' . Now, we need to classify whether players will play or not based on weather condition. Following the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability and probability of playing.

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny).

Approach for the data sets

As per the given problem the dataset should be split into two paarts which is achieved by using randperm command

All the attributes and targets are tobe converted to integer values

The available dataset is checked for any variables with value less than 1 and if any found error will be issued

Laplace smoothing

Laplace Smoothing is a technique to smooth categorical data. Laplace Smoothing is introduced to solve the problem of zero probability. By applying this method, prior probability and conditional probability can be written as:

$$p_\lambda(C_k) = p_\lambda(Y = C_k) = \frac{\sum_{i=1}^{N} I(y_i = C_k) + \lambda}{N + K\lambda}$$

$$p(x_1 = a_j | y = C_k) = \frac{\sum_{i=1}^{N} I(x_{1i} = a_j, y_i = C_k) + \lambda}{\sum_{i=1}^{N} I(y_i = C_k) + A\lambda}$$

K denotes the number of different values in y and A denotes the number of different values in aj. Usually lambda in the formula equals to 1.

Runnig the code

In order to impliment the code rum “main.m” file.

Main.m just contains the function call for the NBclassifier.m and checks for right column size.NBclassifier.m takes inputs from main splits train into artibues of 4 and prior probability is calculated for each attribute in training section.In testing section each atribute is checked for probability and based ont he result, probability of yes and no is decided.In the end accuracy is calculated.