Machine Learning
Artur d'Avila Garcez
4th Tutorial – Naïve Bayes

Background

Naive Bayes is a simple technique for constructing classifiers. Such models assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. Naive Bayes 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 three inches in diameter. A naive Bayes classifier assumes that each of these features contribute independently to the probability that the fruit is an apple, regardless of any possible correlations between colour, roundness and diameter. For some types of probability models, naive Bayes can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes uses *maximum likelihood*. Despite their simple design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers [Zhang, H. (2004). *"The Optimality of Naive Bayes"*. FLAIRS, 2004]. Still, a comprehensive comparison with other classification algorithms in 2006 showed that naïve Bayes classification is outperformed by other approaches, such as boosted trees or random forests [Caruana, R.; Niculescu-Mizil, A. (2006). *"An Empirical Comparison of Supervised Learning Algorithms"*. Proc. 23rd ICML, 2006]. An advantage of naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification.

Setup

Consider the following vector:

(likes shortbread, likes lager, eats porridge, watches England playing football, nationality)^T

A vector $x = (1, 0, 1, 0, 1)^T$ would describe that a person likes shortbread, does not like lager, eats porridge, does not watch England playing football, and is a national of Scotland. The right-most element in the vector denotes the class that we want to predict, and it can take two values: 1 for Scottish, 0 for English.

We will perform simple Naive Bayes to predict such data.

1. Unzip "naiveBayesIntro.zip" into any folder of your choice;
2. Open Matlab and inside it, locate the folder where the contents of "naiveBayesIntro.zip" have been unzipped;
3. Double click on "naiveBayesIntro.m" inside Matlab;
4. Left-click the editor and press F5 to run the code

You should see two results: one represents *P(Scottish|X)* and the other represents *P(English|X).*

Can you work out what the Matlab code does?

Change the test point on line 25 to use different values and check the results.

Gaussian naïve Bayes

Let us use the Gaussian naïve Bayes to classify whether a person is male or female based on the following features: height, weight and foot size, with training set:

Sex	height (feet)	weight (lbs)	foot size(inches)
Male	6	180	12
Male	5.92 (5'11")	190	11
Male	5.58 (5'7")	170	12
Male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

Assuming a Gaussian distribution, calculate the mean and variance of each feature given *male* or *female.*

From the frequencies in the training data or your knowledge of the frequencies in the larger population, calculate P(male) and P(female).

Given test data Sample1 and Sample2 below to be classified each either as male or female, calculate P(C=male|Sample1), P(C=female|Sample1), P(C=male|Sample2), P(C=female|Sample2), using:

$$p(C_k|x_1,\dots,x_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

The equation of the Gaussian can be applied using *normpdf* with mean and variance calculated earlier from the training data, to obtain the conditional probability densities, e.g. P(height=6|C=male), which can be a number greater than 1. Alternatively, use matlab's *cdf* to calculate the area under the curve as the probability of height being approximately 6, which cannot be a number greater than 1.

	Sex	height (feet)	weight (lbs)	foot size(inches)
Sample1:	?	6	130	8
Sample2:	?	5.6	167	9

Apply the Maximum a Posteriori (MAP) decision rule to decide whether Sample1 and Sample2 are male or female.