CSCI479/679 Introduction to Data Mining

Assignment #2 (100 points): Naïve Bayes Classifier

The objectives of this project are:

1. To implement the Naïve Bayes Classifier algorithm.
2. To learn how to assess the classification accuracy.

1 Description

In this assignment you will implement the Naïve Bayes Classifier algorithm. You will write an Octave or R script called naiveBayes.m or naiveBayes.R that implements the Naïve Bayes classification algorithm. You should test your implementation on several datasets (iris, irisPC, Leukemia, buyComputer). The datasets will be uploaded along with this assignment to the blackboard page of the course. Each dataset is partitioned into two datasets, namely the training data and the testing data. The last attribute in each dataset is a class label. Your goal is to learn a model using the training data and use the model to classify the testing data.

2 Approach

Recall that the Bayes theorem allows us to write the posterior probability in terms of the likelihood and prior probability. Please refer to the class notes for the formulas. One significant aspect of the naïve Bayes approach is that it makes the "naïve" assumption that attributes are all independent. This leads to a much simpler way of calculating the joint probability as a product of dimension-wise probabilities:

$$P(x|c_i) = P(x_1, x_2, \cdots, x_d|c_i) = \prod_{j=1}^{d} P(x_j|c_i)$$

For categorical data, the independence assumption leads to the following direct estimate of the probability per dimension:

$$P(x_j|c_i) = \frac{\#\ of\ times\ value\ x_j\ occurs\ in\ D_i}{|D_i|}$$

Moreover, to obtain non-zero probabilities you should employ the *Laplace correction*, where you add a count of one to the observed counts of each value for each class.

For numeric data, use the probability density function (pdf) for the normal distribution to return the likelihood:

$$P(x_j|c_i) = \frac{1}{\sqrt{2\pi {\sigma_j^i}^2}} \exp^{-(x_j-\mu_j^i)^2/2{\sigma_j^i}^2}$$

where μ_j^i, and σ_j^i are the mean and standard deviation for the j attribute for the data points with class label c_i.

To get $P(x_j|c_i)$, you can use the "normpdf(xj,m,s)" function in Octave or the "dnorm(xj,m,s)" function in R, where m is the mean, and s is the standard deviation.

3 DataSets:

3.1 Iris Dataset:

The iris dataset contains 150 instances where each instance has 4 attributes (measurements) and a class label indicating the type of iris plant. The attribute information are as follows: 1.) sepal length in cm, 2.) sepal width in cm, 3.) petal length in cm, 4.) petal width in cm The original dataset has 3 classes: Iris Setosa, Iris Versicolour, Iris Virginica. For more datasets, follow this link: `http://archive.ics.uci.edu/ml/datasets.html`.

In this assignment, we combined 2 classes into one class so we have a total of 2 classes only, (-1 and 1). The data in irisPCTraining.txt and irisPCTesting.txt is the results of applying data reduction using PCA technique on the original data.

3.2 Leukemia Dataset:

A genetic translocation that occurs in acute lymphoblastic leukemia (ALL) that is associated with a mixed-lineage leukemia gene (MLL) results in noticeably worse outcomes. The data set contains gene expression levels for the highest **98** differentially expressed genes for **42** samples. In total, we have 24 samples with ALL (class 1) and 18 samples with MLL (class -1). The dataset was generated using the dataset from the following paper: Armstrong et al (2001), Nat Gen, 30:41-7.

4 Your Program:

4.1 What input parameters your program should take:

Your program should accept the file names of the training data and the testing data. It also should take a flag variable indicating whether the attributes are numeric (1) or categorical (0). Ideally, you would want to get a flag for each attribute. For now, assume that all of the attributes are either numeric or categorical. **Example:**
trainingFile = "irisTraining.txt"
testingFile = "irisTesting.txt"
flag= 1

Alternatively, you can submit two programs, one for numerical data and the other for categorical data.

4.2 Programming Language:

You can implement the algorithm in any programming language. However, I recommend you to use Python, R, or Matlab (Octave).

4.3 What your program should report:

Your program output should consist of the following information:

1. The classification accuracy on each testing dataset.
2. The number of true positives(TPs), false positives(FPs), true negatives (TNs), and false negatives (FNs).
3. The classification Precision and Recall.

5 What to turn in:

Your code (.R, .py, or .m). You should also submit the results on each of the testing datasets in a readme.txt file.

5.1 Submission

Your should submit your assignment to the black board. The file should be named in the following format, useridAssig2.R, useridAssig2.py or useridAssig2.m. Late submission will get a 10% penalty for every day.

6 Grading

Start first by implementing the program to handle datasets with numerical attributes. you will get 100 points for this part. Furthermore, if you implement it to handle datasets with categorical attributes, you will get further 20 points bonus.

Good Luck!