Comp 14112 Lab 2: Naive Bayes Classifier

Tim Morris
Academic session: 2013-14

1 Introduction

This lab involves the implementation of a naive Bayes classifier for the task of differentiating between utterances of the words “yes” and “no”, as discussed in the lecture notes. Directories containing code and data can be found in the following directory

```
/opt/info/courses/COMP14112/labs/lab2
```

Make a copy of this directory in `~/COMP14112/ex2` in your file space. Make sure your CLASSPATH variable is set to include it.

2 Getting started

There are a number of classes in the `naivebayes` package that have `main` methods you can run to demonstrate some of the results from the lectures. The command

```
> java naivebayes.PlotSound 12
```

will plot the sound wave for the 12th example from the data set. The other plot produced when you run this method shows the 1st MFCC for each segment of the same signal and the 1st MFCC averaged over time. We will be using the time-averaged MFCCs as features in order to build a classifier. The examples are “yes” from 1–82 and “no” from 83–165. Try a few different examples and you will see how variable the sound waves are for different people saying the same word.

The command

```
> java naivebayes.PlotHistogram
```

will plot histograms of the time-averaged 1st MFCC for all the examples of each class. This is the same plot as on the left of Figure 4 in the lecture notes.

The command

```
> java naivebayes.PlotFittedNormal
```

will plot two normal densities fitted to the same data. These are the same lines as shown on the left of Figure 5 in the lecture notes.

Finally, the command

```
> java naivebayes.YesNoClassifier
```

uses a single feature in order to a classify the first example in the data set. The example is a yes and the classifier is quite confident that this is the correct classification, assigning a probability of over 0.9 to this class.

3 Guide to the code

You can look at the html documentation for the `naivebayes` package to see what the various classes do. In this lab you will be adapting two classes: `YesNoClassifier` and `Classifier`, so you should look at the code for these in particular. The `javagently` package is a very basic plotting program which you don’t have to worry about.

4 The tasks

You have three tasks.

1. (4 marks) Modify the code in the `main` method of `YesNoClassifier` in order to return the percentage of errors that the classifier makes on all 165 examples in the data set.
2. (4 marks) Complete the code in the `classify (double[] featureVector)` method of the `Classifier` class. This should implement a naive Bayes classifier that uses all of the feature vector components. Once you have implemented this method, evaluate its performance in comparison to the single feature approach.
3. (2 marks) Create a new constructor method for the `Classifier` class which estimates the priors p(C1) and p(C2) from the data.

5 Evaluation

You will get 7 marks for a correct working implementation and 3 marks for a full understanding of what you have done. You should **submit** your modified files: `YesNoClassifier.java` and `Classifier.java`. You should also run **labprint**.