CSC C11 – Fall 2017
Assignment 3 – How to tell you're reading news

Assignment due date: Friday, Dec. 1 - 10am

Code to be submitted on the mathlab server by the above due date

This assignment can be completed individually, or in a team of 2 students

Student Names (last, first)

Student #1:
Student #2:

Student numbers:

Student #1:
Student #2:

Student UtorIDs:

Student #1:
Student #2:

We hereby affirm that all the solutions we provide, both in writing and in code, for this assignment are our own. We have properly cited and noted any reference material we used to arrive at our solution, and we understand both the written solutions and code submitted with our assignment.

_____ _____

Student #1 signature ***Student #2 signature***

(note: 3 marks penalty if any of the above information is missing)

CSC C11 – Fall 2017
Assignment 3 – How to tell you're reading news

Your final assignment continues the work you did in A2. Remember that our task back then was to take a pool of documents, and sort them into clusters that belong to one of 5 specific topics. You explored issues related to the dimensionality of your dataset as well as to the usefulness of pre-processing your input documents to make clustering easier.
In this assignment, you will deal with the same set of documents except that now we are given the following task: If we are provided the ***classes*** or ***categories*** for a subset of these documents (our ***training set***), can we build a classifier that will correctly sort the remaining documents into the correct categories? The goal of such a system would be to automatically classify any future documents without the need for human supervision.

Classification is one of the most important tasks in Machine Learning, let's have a go at it!

Learning Objectives – after completing this assignment you should be able to:

Explain the difference between a training data set and a testing data set, and use them appropriately in the design and implementation of a classifier.

Understand and apply PCA to an input data set in order to produce a low-dimensional version so that we can train classifiers that can not be trained on the original data.

Expand the concepts you learned during clustering with GMMs in order to understand how Gaussian Class Conditionals work for classification.

Explain and implement a Naive Bayes classifier.

Skills Developed:

Implementing and using PCA – one of the most useful tools in data science for data analysis, visualization, compression, and dimensionality reduction.

Implementing Naive Bayes – in particular handling multiplications of tiny numbers by working in the log domain.

Training, testing, and comparing classifiers of different types.

Reference material:

The lecture notes on classifiers

Comments included in the starter code.

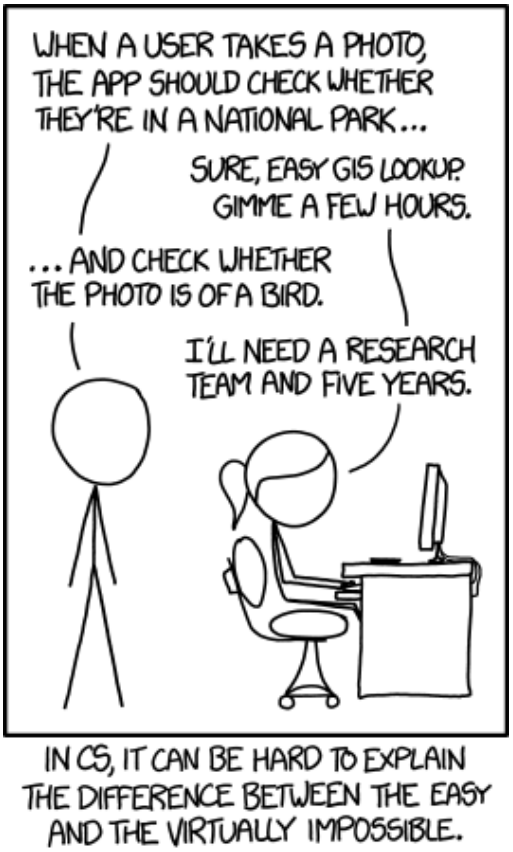
Acknowledgements:

Document training data provided by: http://mlg.ucd.ie/datasets/bbc.html

CSC C11 – Fall 2017
Assignment 3 – How to tell you're reading news

Classification

It is a widely used ML task, and has many forms. Of course sorting data into categories, but also detection tasks (is there a person in this picture?), tracking (is the same person found in a sequence of images, and if so where?), pattern recognition (did you just say 'Ok Google'?), and so on. Depending on the task, it can be very difficult.



Courtesy of xkcd.com:
https://xkcd.com/1425/

Your Task:

Download and unpack the starter code. It contains a set of functions you need to complete for this assignment.

You will implement:

A Gaussian Class Conditional classifier
A Naive Bayes classifier
PCA for dimensionality reduction

Along the way you will explore issues such as how classification accuracy changes between training data and testing data, how the dimensionality of your data affects classification accuracy, and whether or not making assumptions about the independence of features (as we do with the Naive Bayes classifier) Is a reasonable thing to do even when we know features are not actually independent.

You will use the same document vectors you clusteres in A2, except this time you have labels for the training and testing sets (these were provided by human observers).

1) Implement and test the code that does PCA

Complete the function that takes an input data set, computes the PCA eigenvectors and the mean of the dataset, and returns a low-dimensional representation of the dataset. I suggest you test this on a simpler dataset than the BBC dataset, for example, feed it the pixels of a colour image (one per row, 3 values per pixel) and make sure you can ***reconstruct*** the original data from the PCA representation. ***Our auto-tester will check your PCA code computes the correct low-dim representation, so make sure your PCA code does the right thing.***

Once you are happy your PCA code works – compute a low-dimensional representation of the BBC ***training*** dataset (using 3 dimensions), and ***plot on a 3D graph the low dimensional vectors for the input documents.*** The colour for each point should be given by the class for the corresponding document – choose 5 any colours you like.

CSC C11 – Fall 2017
Assignment 3 – How to tell you're reading news

1) (cont.)

Save the plot as '***BBC_low_dimensional.pdf***'. You will submit this plot with your code. Answer this question:

- Do the points from different classes look reasonably separable in this space?

2) Implement the functions that train and classify using Gaussian Class Conditionals.

For this part, your GCC model will be trained to have 5 components (one per category in the BBC dataset), and you will use the ***low dimensional representation*** you obtained from PCA. This part should be reasonably straightforward to do since it is very similar to what you did For estimating the parameters of Gaussian components in your GMM clustering code from A2.

The training function will return the parameters of the GCC model, and then your classification function uses those parameters to determine classes for an input dataset.

Needless to say, you will train the GCC model on the ***BBC training dataset***. ***Do not use testing data for training the GCC model.***

Once you're done implementing your GCC model, you will carry out the following experiments:

Train a GCC model for data with dimensionality 3, 5, 6, 7, 9, 10, 15. For each of these cases, report:

- Classification accuracy on the training set
- Classification accuracy on the testing set

Classification accuracy is just the percentage of labels returned by your classifier that agrees with the labels attached to the dataset.

Answer these questions:

- What is the optimal dimensionality for the GCC classifier? (justify this choice based on your experimental observations).

3) Implement the functions that train and test a Naive Bayes classifier. For this part you will be working with ***full-dimensionality data***. Be sure to pay attention to notes in the starter code.

Once you're done implementing your NB classifier, report:

- Classification accuracy on the training set
- Classification accuracy on the testing set
- For this classification problem, which classifier (GCC or NB) works best?
- Why is the ***weaker*** classifier less accurate? (i.e. explain in terms of input data, classifier structure, or assumptions made by the classifier or training process why it performs worse).

CSC C11 – Fall 2017
Assignment 3 – How to tell you're reading news

(5 mark penalty for not following these instructions carefully):

Step 4 – Package your solution and submit it.

Fill-in 'autotester_id.txt'. Put the answers to your questions in a ***text file called 'our_A3_answers.txt'***, and compress your code, answers, and plot unto a single compressed file called:

Classify_studentNo1_studentNo2.tgz **(e.g. Classify_11223344_55667788.tgz)**

The *tar* command syntax is:

```
>tar  -cvzf  name_of_your_compressed_tar_file.tgz  *.m  *.txt *.png
```

Then submit the compressed file ***(don't copy/paste, type the command)***

```
>submit -c cscc11f17 -a A3 -f name_of_your_compressed_tar_file.tgz
```

Double check that your compressed file uncompresses properly, and that it contains all the filed noted above.

Marking Scheme	
Written answers	15 marks
Working code	35 marks (auto-tested)
In-class quiz	50 marks (individual)

Optional Crunchy Bonus (up to 25 marks)

For bonus marks – come up with a different, crunchy application of classification and write a working demo script.

Find an ***interesting*** dataset attached to a problem you find cool. Then train and test classifiers and report on how well they perform for the dataset you chose. You can use PCA, pre-process your data, etc. Be sure to provide a complete description of the problem and your approach to developing a classifier for it.

As usual, you can work on this with your team partner.