Inf2B assignment 2
Music genre classification

Hiroshi Shimodaira and Benigno Uria
Due: 4pm, Friday 28 March 2014

This assignment is out of 100 marks and forms 12.5% of your final Inf2b grade.

Assessed work is subject to University regulations on academic conduct: `http://www.ed.ac.uk/schools-departments/academic-services/students/postgraduate-taught/discipline/academic-misconduct`
Do not show any written material or code to anyone else. Never copy-and-paste material into your assignment and edit it. High-level discussion without exchange of written materials is fine. You may obtain standard routines for computing Gaussian probability densities, from lecture notes or other websites. *Credit any 3rd-party code for these operations.*

Extensions:
The School does not normally allow late submissions. See: `http://www.inf.ed.ac.uk/student-services/teaching-organisation/for-taught-students/coursework-and-projects/late-coursework-submission` for exceptions to this rule, e.g. in case of serious medical illness or serious personal problems. Any extension request must be made to the ITO, not the lecturer.

1 Introduction

The goal of this assignment is to classify acoustic recordings of musical fragments from the GTZAN dataset (`http://marsyas.info`) into their corresponding music genres using some of the classification techniques studied.

2 Data

Each music fragment to be classified consists of a vector of 124 real-valued features extracted by preprocessing acoustic recordings. Each fragment corresponds to a music genre:

id	genre
1	blues
2	classical
3	country
4	disco
5	hiphop
6	jazz
7	metal
8	pop
9	reggae
10	rock

The data is stored in a matlab file named gtzan.mat located in /afs/inf.ed.ac.uk/group/teaching/inf2b/cwk2 This file contains 20 matrices: `fold1_features`, `fold1_classes`, ..., `fold10_features`, `fold10_classes`.

Matrices with a name ending in “features” contain a musical fragment per row. Matrices whose names end in “classes” contain the *genre’s id* of the corresponding row in the corresponding “features” matrix.

As you can see, the dataset has been shuffled and split into ten subsets, also called folds. Which will allow us to do 10-fold crossvalidation. This means we will repeat our experiments 10 times, each time leaving one of the folds out of the training data and using it to measure the performance of the classifier. By averaging the 10 results we obtain a more accurate estimate of the classification performance.

You can also load the data in Python:

```
import scipy.io

data = scipy.io.loadmat(‘‘gtzan.mat’’)
```

in which case `data` will be a dictionary containing the 20 matrices.

In the same directory you will also find:

- `0.README`: a text file describing the content of each file.
- `audio`: subfolder containing the audio files in a reproducible format, and text files listing the correspondence between audio files and rows in the ten data folds. This is provided in case you want to listen to any of them.
- `list_of_features.txt`: a text file describing the 124 features used for classification.
- `Tzanetakis_Cook_2002.pdf`: a paper describing the dataset and feature processing.

3 Classification methods (60 marks)

Implement the following classification methods in either Matlab or Python, and utilize them to classify the data provided using 10-fold crossvalidation. Report the classification accuracy average (i.e. the average accuracy of the 10 experiments) for each method. Answer the questions posed in this section as part of your report, do not do it in the code or in a separate document.

3.1 k-nearest neighbours (30 marks)

- Classify the data using k-nearest neighbours utilizing the Euclidean distance and values of k 1, 3 and 5. In case of ties, *i.e.* if several classes have the greatest number of points in the k nearest neighbours, remove the furthest point one at a time until the tie is broken.
- Report the confusion matrix for $k = 1$. That is, report a table where the entry in row i and column j shows the proportion of datapoints of class i that were classified as belonging to class j. Are all genres equally difficult to classify? What is the most common error?

3.2 Gaussian models (30 marks)

- Assume a uniform prior distribution over classes. What is the prior probability for each class? What classification accuracy would we obtain if we classified each music fragment by choosing a label at random?
- Fit a diagonal covariance Gaussian model to each genre. What is the classification accuracy?. Are the classification boundaries of this method linear? (There is no need to run experiments to answer this last question)
- Fit a Gaussian model to each music genre, all of them sharing the same full covariance matrix (remember to substract the mean of its class to each datapoint before calculating the shared covariance matrix). Obtain a linear discriminant function from these models. Classify the test dataset using the linear discriminant function. What is the classification accuracy? What is the shape of the classification boundaries of this method? (There is no need to run experiments to answer this last question)

4 Report (40 marks)

Write a technical report (about two pages long) summarising the results obtained in the previous section and commenting on the advantages and disadvantages of each method.

An excellent report will include: tables summarising all results, an analysis of the accuracy of each method, comments about the computational complexity of each method (both training complexity and classification complexity), a substantiated recommendation for a method to use in a production system, and a proposal for further experiments.

5 Submission

You must submit your work before the deadline given at the head of this document.

Save your report file in *pdf* format and name it `report.pdf`. Remember to place your student number prominently at the top of the document. Create a folder named `LearnCW`, copy your report pdf file in it and copy all your code into a subfolder named *src*. The structure of the directory should look like this (the code files may be named differently):

```
|-- LearnCW
   |-- report.pdf
   |-- src
      |-- main.py
      |-- gaussian.py
      |-- knn.py
```

Submit it from a DICE machine using: `submit inf2b 2 LearnCW`

We can only mark files if they are submitted. Please, make sure you have included your report.pdf file and all your code is in the `src` subfolder.