New Mexico State University
Klipsch School of Electrical Engineering

EE565 Pattern Recognition and Machine Learning
Fall 2016 – Project #3
Due: 5:00pm Thu. Sep. 29

Name: _____

Grade: _____

Project

The goal of this project is to gain familiarity with the Gaussian Mixture Model (GMM) and use it to build a system that can identify an unknown musical song. You may already be familiar with this technology:

`http://www.shazam.com/`

`http://www.soundhound.com/`

We will utilize a subset (200 songs) from the uspop2002 corpus where each file contains a sequence of 20-D feature vectors from a song. The elements of the feature vector are the mel-frequency cepstral coefficients (MFCCs) which are widely used as "features" in audio and speech processing:

`http://en.wikipedia.org/wiki/Mel-frequency_cepstrum`

These features are stored in an HTK format file; there are HTK file readers available in MATLAB and Python. See for example

`http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`

`https://pypi.python.org/pypi/htk_io`

The sequence of feature vectors for each song has been randomly partitioned into data for training, test, and evaluation. You will use each training file to construct a GMM of the song's features. For each test file, you will compute the likelihood for each GMM and identify the song based on the maximum likelihood (ML). Since we will assume equal priors on the songs and ignore the common normalization factor, this decision is equivalent to a maximum a posteriori (MAP) decision. Supplied keys for the tests are included so you can check performance. In addition, there are two evaluations and you will submit your predicted song identities as part of the project; evaluation keys will be distributed after the project deadline. To expedite grading, predicted song identities, e.g. 'queen/A_Night_At_the_Opera/11-Bohemian_Rhapsody.htk' are to be written (one per row) to an output text file corresponding to each test file.

Companion files for this project may be found at

`http://www.ece.nmsu.edu/~pdeleon/Teaching/EE565/Projects/CompanionFiles3.zip`

In the course of developing the song identification system, you will have to make some design choices including the number of components to use in the GMM, whether to use full or diagonal covariance matrices, how to initialize the EM algorithm, maximum number of EM iterations, etc.

Your grade will not entirely be based on accuracy but this will, of course, be very important. Most important, however, is your design approach, code, simulation, and reporting.

Preliminary

There is excellent support for GMMs in MATLAB and Python. In MATLAB, you encouraged to use the following functions (see help for more details): `gmdistribution`, `gmdistribution.fit`, `pdf`, and `posterior`. In Python, you encouraged to use scikit-learn.

Problem 1: Gaussian Mixture Modeling of the Old Faithful Dataset

Recreate each plot in Figure 9.8. The following MATLAB code may be helpful.

```
% Scale data according to text
load 'faithful.txt'
X = zscore(faithful); % standardize (see p. 427)

% Eyeball the initial GMM params and build the object
K = 2;
mean_init(1,:) = [-1.5 1.0];mean_init(2,:) = [1.5 -1.0];
cov_init(:,:,1) = eye(2);cov_init(:,:,2) = eye(2);
weight_init(1) = 0.5;weight_init(2) = 0.5;
gmm_init_struct = struct('mu',mean_init,'Sigma',cov_init,'PComponents',weight_init);

% Estimate GMM params of data points using EM algorithm
Options = statset('MaxIter',5);
lambda = gmdistribution.fit(X,K,'Start',gmm_init_struct,'Options',Options);

% Plot contours of component densities
[x1,x2] = meshgrid(-4:0.1:4,-4:0.1:4); % Create grid.
[r,c]=size(x1);data = [x1(:) x2(:)];
px=mvnpdf(data,lambda.mu(1,:),lambda.Sigma(:,:,1)); % eval pdf #1
contour(x1,x2,reshape(px,r,c),3,'b'); % contour plot
hold on;
px=mvnpdf(data,lambda.mu(2,:),lambda.Sigma(:,:,2)); % eval pdf #2
contour(x1,x2,reshape(px,r,c),3,'r'); % contour plot

% Color data points according to posterior and plot
P = posterior(lambda,X);
cmap=[0:.01:1; zeros(1,101); 1:-.01:0];
colormap(cmap');
scatter(X(:,1),X(:,2),18,P(:,2),'filled');
hold off;

axis([-2.5 2.5 -2.5 2.5]);axis('equal');
```

Problem 2: Song Identification System

Report

Please submit a hardcopy report detailing your song identification system; also include plots from Problem 1. Please `mailto:pdeleon@nmsu.edu` a zip file containing all code for this project, data models from training (GMMs of song features), and output1.txt and output2.txt (predicted song identities from the evaluation). It goes without saying that all code should run "out-of-the-box" (no hard-coded paths).

Notes

Students are encouraged to discuss detailed, technical aspects with each other and Prof. De Leon. However, students must write all other required codes on an *individual* basis.