Lab 3
Lab report due by 1:00 PM on Monday, March 6, 2017

This is a MATLAB only lab, and therefore each student needs to turn in her/his own lab report and own programs.

---

1 Introduction

This is the third and final lab on digital signal processing methods for content-based music information retrieval. In this lab we will develop distances to compare audio tracks, and classification methods to automatically detect the genre of a track. The distances will be based on the features that we have developed in the first and second labs: low level temporal and spectral features, mfcc coefficients, rhythm, and chroma.

The algorithms that you will develop will exploit a database of 150 training samples to learn the association between genres and songs. The performance of the algorithm will be evaluated using songs with know labels that are not part of the training set.

2 Terminology

For the purpose of this document we will be describing music using a terminology that is biased toward popular music. As a results we use the following words in a somewhat different context:

- an artist refers to the performer of a piece of music in the case of popular music, and a composer in the case of classical music. For instance, both Miles Davis and Johann Sebastian Bach are two artists.
- a song refers to a recording of (a part of) a piece of music. A song is identified with a CD track. The name does not imply any vocal performance.

3 Mission challenge

3.1 Genre classification

While the definition of a musical genre is clearly arbitrary, and influenced by a culture, we will define genre classification as the problem of classifying music according to a set of (personal) labels. In particular, we will consider the following labels:

1. classical (western classical music)
2. electronic
3. jazz/blues
4. metal/punk
5. pop/rock
6. world.

The last label "world" is really a category that is only defined as being the complement of the union of the other categories.

The goal of the lab is to develop an algorithm that can upload a song and return a label with a certain probability. For instance, if the song "Recitative" from the album "Dark Intervals" by Keith Jarret is selected, one would expect that the algorithm classifies it as "classical" or "jazz" with a high probability. The artist Keith Jarrett is a clear example of the difficulty of such a genre classification, and the need for a probabilistic framework. The lab focuses on the genre classification question.

4 The data

4.1 Raw audio data

The data available for the lab are Pulse Coded Modulation (PCM) encode audio files. These format provides a digital sampling of the acoustic wave created by the sound pressures changes as a function of time. The data for this lab has the following characteristics:

- sampling rate: 22,050 Hz
- sample size: 16 bit
- Number of channels: 1 (mono)
- Encoding: WAV

4.2 Available files

The composition of the training dataset is as follows:

- classical: 25 songs
- electronic: 25 songs
- jazz/blues: 25 songs
- metal/punk: 25 songs
- rock/pop: 25 songs

- world: 25 songs

We note that the lab is not asking you to refine the classification beyond the six categories.

Each song is a track of a CD and last for several minutes. While songs may have different lengths, we will work with an excerpt of fixed length extracted from the centre of the track.

4.3 But where are the songs?

An archive with the 150 songs is available. You will need a password to download the files; these have been emailed to you.

4.4 Copyright

5 Statistical Evaluation of the Performance of My Algorithm

5.1 Confusion matrix

The accuracy of the classification will be quantified using confusion matrices. In this lab, a confusion matrix will measure the correct classification rate of each genre. For a given classification experiment, you will construct a 6×6 matrix where the rows are the true genres, and the columns are the genres classified by your algorithm. The entry $R(i, j)$ of the confusion matrix R is the number of songs of genre i that were classified as genre j. In the example in Table 1 the total number of songs per class was:

$$\begin{bmatrix} 21 & 3 & 1 & 0 & 0 & 0 \end{bmatrix},$$

so there were 21 good identifications (diagonal entries of the matrix) out of 25 queries. This corresponds to 84 % correct answers overall. The correct classification rate per class was:

$$\begin{bmatrix} 0.84 & 0.88 & 0.52 & 0.92 & 0.36 & 0.40 \end{bmatrix}. \tag{1}$$

	classical	electronic	jazz	punk	rock	world
classical	21	3	1	0	0	0
electronics	3	22	0	0	0	0
jazz	0	1	13	2	8	1
punk	0	0	0	23	1	1
rock	0	2	3	6	9	5
world	3	7	1	0	4	10

Table 1: Example of a confusion matrix. The rows are the true genres, and the columns are the predicted genres.

5.2 Cross-validation

In order to evaluate the performance of your method, you will perform a 5-fold cross-validation 10 times. Each cross-validation experiment consists in the following sequence of operations.

- let $\mathcal{S}$ be the set of all 150 songs.
- let $\mathcal{G}_i, i = 1, \cdots, 6$ be the six sets of songs organized by genres:

$$\mathcal{S} = \bigcup_{i=1}^{6} \mathcal{G}_i.$$

- **repeat n =1:10** // `average over the randomization`

  1. randomly divide each genre $\mathcal{G}_i$ into 5 subsets of size 5

$$\mathcal{G}_i = \bigcup_{k=1}^{5} \mathcal{G}_i^k$$

     where $|\mathcal{G}_i^k| = |\mathcal{G}_i|/5 = 5$.

  2. **for** $k = 1$ to 5 `// round robin over the testing songs`
     - form the set of test songs

$$\mathcal{U} = \bigcup_{i=1}^{6} \{\mathcal{G}_i^k, \}$$

       and the set of training songs

$$\mathcal{L} = \bigcup_{i=1}^{6} \{\mathcal{G}_i^l,\ l = 1, \cdots, 5, l \neq k\}$$

     - test the performance of your algorithm on the 30 songs in $\mathcal{U}$ using the 120 training songs $\mathcal{L}$

- record the confusion matrix $R^{(k,n)}$

end for

end repeat

- **for** each entry (i, j) of the confusion matrix:
  - compute the average $\overline{R}(i, j)$
  - compute the standard deviation $\sigma_R(i, j)$

end for

The final average confusion matrix, and the associated standard deviation should be included in the report.

6 Distances between tracks

The goal of the classification is to assign a genre to a song. In this lab, we will focus on simple classification techniques, such as the k-nearest neighbors. The key ingredient of the approach is a distance that quantifies the similarity between tracks in terms of musical features. In the previous two labs, you have developed computational tools to extract musical features using sophisticated signal processing algorithms. You will combine these features (MFCC coefficients, rhythm, chroma, etc.) to construct a distance that maximally separate the different genres.

For the purpose of the discussion, we denote by $\boldsymbol{X}_i$ the set of features that you compute for track i.

6.1 Property of a distance between tracks

Because we are interested in classifying songs according to their genre, the distance should provide a natural partitioning of the dataset according to the following criterion:

1. for all genres, the maximum distance between two songs of this genre is always smaller than the shortest distance to any song of a different genre,

$$\max_{i,j \in \text{same genre}} d(\boldsymbol{X}_i, \boldsymbol{X}_j) < \min_{k,l \in \text{different genres}} d(\boldsymbol{X}_k, \boldsymbol{X}_l).$$

Because, the minimum and maximum distances may be very sensitive to anomalies, we replace this criterion with the following criterion:

1. for a given genre g, we define its centroid as

$$\overline{\boldsymbol{X}}_g = \frac{1}{25} \sum_{k \in \text{genre } g} \boldsymbol{X}_k \tag{2}$$

2. We also define the radius ρ_g of a genre g as the standard deviation of the genre

$$\rho_g = \frac{1}{25} \sum_{k \in \text{genre } g} d(\boldsymbol{X}_k, \overline{\boldsymbol{X}}_g) \tag{3}$$

3. Finally, we require that the genres do not overlap, and therefore if we consider two genres, g and g',

$$d(\overline{\boldsymbol{X}}_g, \overline{\boldsymbol{X}}_{g'}) > \rho_g + \rho_{g'} \tag{4}$$

6.2 Features

We extract an audio excerpt of 2 minutes from the center of each piece, and compute several audio features, and combine them into a vector $\boldsymbol{X}$. If the track is shorter than 2 minutes, we use the entire track. We will work with frames of size 512 samples (23 ms for the sampling rate equal to 22,050 Hz), and an overlap between frames of 50% = 256 samples. We have $N_F = 10,335$ overlapping frames of size 512 in 2 minutes of music.

We consider the following two vectors of features,

1. MFCC coefficients,

2. chroma (Normalized Pitch Class Profile)

Both sets of features yield a matrix $\boldsymbol{X}(k, n)$ that depends on a frequency (or pitch) index $k = 1, \ldots, K$, and a frame index $n = 1, \ldots, N_F$. We can think informally about $\boldsymbol{X}(1 : K, n)$ as the set of notes being played at time n.

In order to obtain more reliable classification results, and speed up the computation, we merge some of the Mel banks. We retain only 12 bands, as in the chroma representation. The new bands are defined by the next lines of MATLAB code.

```
t = zeros(1,36); (2.21)
t(1) =1;t(7:8)=5;t(15:18)= 9;
t(2)   = 2; t( 9:10) = 6; t(19:23) = 10;
t(3:4) = 3; t(11:12) = 7; t(24:29) = 11;
t(5:6) = 4; t(13:14) = 8; t(30:36) = 12;

mel2 = zeros(12,size(mfcc,2));

for i=1:12,
    mel2(i,:) = sum(mfcc(t==i,:),1);
end
```

In the remaining, we will consider that the mfcc coefficients are the 12 merged mfcc coefficients; in other words

```
mfcc = mel2;
```

6.3 Distance between features

Given two tracks s and s', we are interested in comparing the distribution of notes between the two tracks. We proceed as follows.

For each track, the distribution of vectors $\boldsymbol{X}(:, n), n = 1, \ldots, N_F$ is modeled as a multivariate Gaussian distribution in $\mathbb{R}^K$, where $K = 12$. In essence, this approach collapses all the frames together, and summarizes the 2-minute excerpt by a mean note, and a covariance matrix. In MATLAB, we compute

```
>> mu = mean(mfcc,2);
>> Cov = cov(mfcc');
```

To compare two tracks, we compare the distance between the two Gaussian distributions $G^s = \mathcal{N}(\mu_s, \Sigma_s)$ and $G^{s'} = \mathcal{N}(\mu_{s'}, \Sigma_{s'})$ that are estimated for each track. We use a standard approach that is used in statistics, and we compute a symmetric version of the Kullback-Leibler divergence:

$$KL(G^s, G^{s'}) = \frac{1}{2}\left(\text{tr}(\Sigma_{s'}^{-1}\Sigma_s) + (\mu_{s'} - \mu_s)^T\Sigma_{s'}^{-1}(\mu_{s'} - \mu_s) - K + \log\left(\frac{\det \Sigma_{s'}}{\det \Sigma_s}\right)\right). \tag{5}$$

We note that this distance is very similar to the Mahalanobis distance discussed in class.

Finally, the KL distance is rescaled using an exponential kernel, and we define the distance between the tracks s and s' as

$$d(s, s') = \exp\left(-\gamma KL(G^s, G^{s'})\right), \tag{6}$$

where the parameter γ is chosen in $[0, 1]$ to optimize the classification.

6.4 Distance matrix

Assignment

1. Implement the computation of the distance D given by (6). Your function should be able to use the 12 merged MFCC coefficients or the Normalized Pitch Class Profile.

2. Compute the matrix of pairwise distance

$$D(s, s'), s, s' = 1, \ldots, 150. \tag{7}$$

   Display the distance matrix as an image, and discuss its structure.

3. For each genre, compute the histogram composed of the 300 pairwise distances within that genre. Plot the six histograms on the same figure. Comment on the figure.

4. Compute the 6×6 average distance matrix between the genres, defined by

$$\overline{D}(i, j) = \frac{1}{25^2} \sum_{s \in \text{genre } i,\ s' \in \text{genre } j} d(s, s'), \quad i, j = 1, \ldots, 6, \tag{8}$$

5. Experiment with different values of γ, and find a value of γ that maximizes the separation between the different genres, as defined by the 6×6 average distance matrix $\overline{D}$.

6. Compare the MFCC and the Normalized Pitch Class Profile in terms of the 6×6 average distance matrix $\overline{D}$.

6.5 Impact of the length of the excerpt on the classification

Assignment

7. Evaluate the effect of the length of the audio segment using the following lengths: 30, 60, 120, 240 seconds. For each length you will to compare the separation of the different genres using the 6×6 average distance matrix $\overline{D}$.

7 Classification method

We are now equipped with a distance to measure the similarity between two tracks. We describe here a simple procedure to classify a song with unknown genre using the training data.

7.0.1 K-Nearest Neighbors

Given a song s, we wish to determine its genre. We proceed as follows.

We compute the distance between the song s and every other song in the training data, and we determine the five nearest songs (according to d defined by (6)). Among the five nearest songs, we find the genre that is the most represented, and assign this genre to the track s.

Assignment

8. Implement a classifier based on the following ingredients, as explained above,
   - computation of the 12 mfcc coefficients, or 12 Normalized Pitch Class Profile
   - modified Kullback-Leibler distance d defined by (6)
   - genre = majority vote among the 5 nearest neighbors
9. Using cross validation, as explained in section 5.2, evaluate your classification algorithm. You will compute the mean and standard deviation for all the entries in the confusion matrices.
10. Compare the performance of the classification using the 12 mfcc coefficients, or 12 Normalized Pitch Class Profile.

7.1 Improve the classifier

You can improve the classification algorithm in several directions.

Assignment

11. Improve the classifier using support vector machines (SVM).