Analysis of digital audio data
Singer recognition

The goal of this work is to study simple methods for classifying pieces of according to the identity of the singer. The recordings considered here are pieces of musics composed of two types of sources: "voice" and "music".

Description of the Data:

We have a corpus of 40 songs sung by 10 different singers (4 musics by singers). At each singer is awarded a label ranging from a to j. Each of these songs was segmented into two parts music and voice. The first corresponds to the first 15 seconds of the accompaniment and the second to 15 seconds after the first minute. In total we have 80 musical extracts, 40 with a sung part, 40 without sung part, divided into 10 classes labeled from a to j depending on the singer.
Note that, according to the musical style, the accompaniment (music) is likely to be present in both excerpts of the same pieces, while the singer is likely to be present only in the second extract. This segmentation is arbitrary, but works well in the case of popular pieces.

Calculation of the descriptors:

We implement the function [V, Label2, Label10] = computefeatures (dataPath) which provides:

- the matrix V comprising the 80 characteristic vectors v_i to n dimensions of our pieces of music (line: pieces of music; column: coefficient)
- the vector label2 indicating for each piece if it is a sung part (label2 (i) = 1) or a musical part (label2 (i) = 2)
- the vector label10 indicating for each piece whether it is the singer a (label10 (i) = 1), the singer b (label10 (i) = 2), and so on until singer j (label10 (i) = 10)

We will begin by representing our data using the power spectrum. To do this we implement the function *Xi* = pSpec (xi , nfft, nwin, hopsize) which calculates the spectrogram of power *Xi* (freq × time) of the signal *xi* by performing a short-term Fourier transform with a hamming window.

Input Parameters:
dataPath: directory path containing the sound files (see dir function)
nfft: Number of points in the FFT

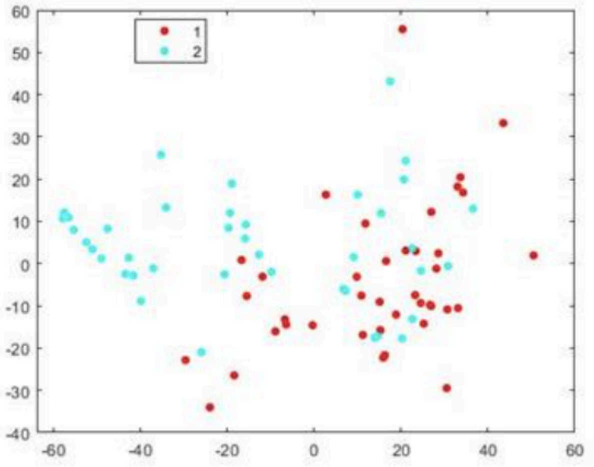
nwin: size of the analysis window. (zero-padding if nwin < nfft, nfft must be a power of 2)

hopsize: no progress of the analysis window (typically nwin / 4 or nwin / 2)
The characteristic vectors v_i will be obtained by realizing the temporal mean of the representations X_i corresponding.

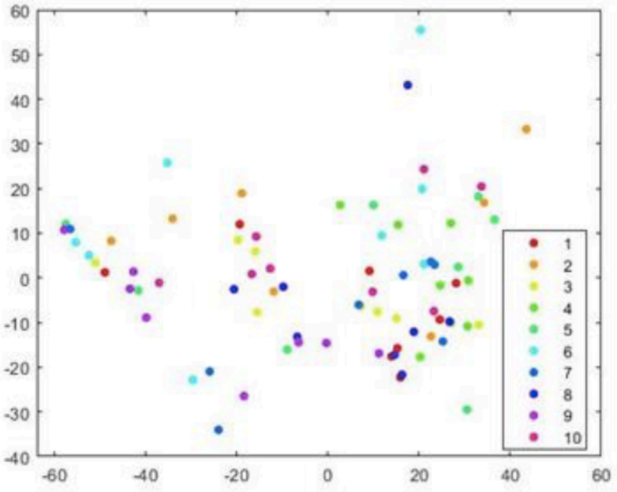
Visualization :

In order to visualize the relative positions of our points described by the vectors v $_i$, we will have to reduce the size of our space of representation. To do this we will use the Principal Analysis Component (PCA).We implement the dataVisu (V, label) function that displays on a 2-dimensional space the points representing our music pieces

Results by label2:



Results by label10:

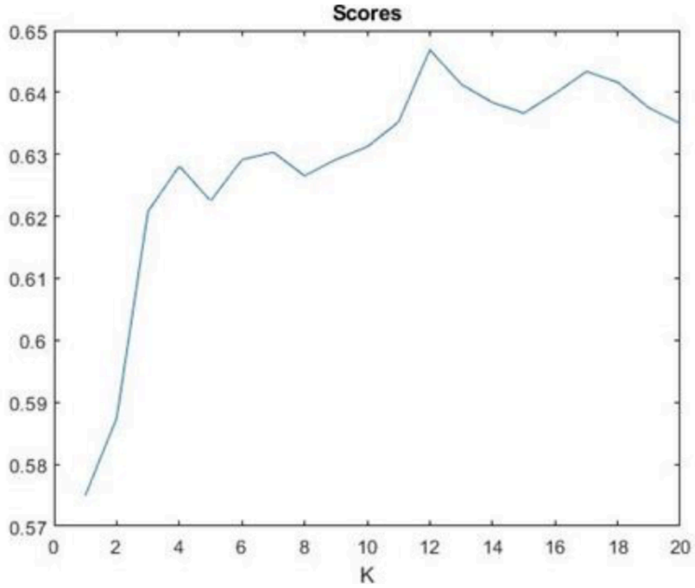


The result shows us that we can’t determinate the singers with this descriptors. However we may be able to detect if the song is music or voice but the segmentation will not be excellent.

Evaluation of the description system:

To evaluate our description system, we will use the so-called precision metric at rank K. It is a question of calculating, for each piece of music represented by the vector v_i , the number of K nearest pieces v_j having the same label as v_i . The accuracy at rank K is calculated in counting for the set of vectors v_i , the number of K nearest pieces v_j having the label of v_i and normalizing by the set of data tested (K * 80).
We implement the function [score] = precAtK (K, V, Gt) which calculates the precision at rank K for the matrix V , depending on the ground truth Gt (label2 or label10).
We will choose K by considering as truth the ground changes
Calculate the precisions for different values of K by considering as ground truth changes (label10) or the type of extract (label2).

Here is the evolution of the score depending of the K values.



We can see a maximum of the score for K=12 and we can also confirm the limit of the method because the maximum score is about 0.648.

Search by similarity

We will now use our representation system to detect automatically if a piece has voice.

We implement the function [label] = detectVoice (x, K, V, label2) which classifies an input signal x buy using the nearest neighbor K method, depending on the V matrix containing our vectors of descriptions v_i and the label2 vector containing the labels (1 = voice, 2 = music) of the vectors v_i .
We tested this function on the bent_music and bent_voice files (present in the folder…). You can also test it on any piece at your disposal.

Results.

After testing our program, the results wasn’t that good. In fact we succeeded to detect the music (label = 2 for bent_music but it was the same for most pieces of bent_voice. We suppose that the descriptors aren’t relevant so we tested the program with other descriptors like the logarithm of the spectrogram.
And the result was much better :

Spectrogram of power:
Score = 0,5153 with K=12
Logarithm of spectrogram:
Score = 0,7063 with K=4