Introduction

What's the 'goal' of this project? We want to classify pieces of music by genre. The principal issue lies in the definition of genre itself, because a particular songs, or music in general, is classified using too many different aspects which are mostly subjective.

Musical genres have no strict definitions, they can change using one point of view or another, as they are a product of multiple factors.

It is true, however, that musical pieces which belong to a specific genre share similar characteristics (like rhythm, harmonic structure, etc...). So it's possible to classify genres using the characteristic which is most objective: the sound itself.

The first step is reducing the information we want to analyze using only useful informations. We can maybe be interested in how the spectrum is distributed or how the sound is loud or how it changes his loudness.

Short-term feature extraction

The audio signal is divided in windows. For each window it's calculated a vector of features, some based on time signal representation and some others on frequency representation.

From this operation it's obtained a vector of N elements, depending on the duration of the audio considered, his sample frequency, the window size chosen and the how much the windows are overlapped.

In this case I chose to divide songs in 3 groups, according to 3 different frame-size features extraction: 20ms, 40ms and 100ms.

Features in time domain (energy and zero crossing rate)

The Feature in time domain to examinee are the Energy (magnitude) of the signal and the Zero-Crossing Rate, the frequency at which the signal changes from positive to negative or back.

Features in frequency domain (spectral centroid and spread, spectral rolloff and mel-frequency cepstral coefficients)

The spectrum can be described using simply two features: the centroid which is the main point of the spectrum distribution and the spread which is the standard deviation of a spectrum distribution.
The spectral rolloff calculates the amount of energy cumulated until a certain point in the frequency.
Mel-Frequency Cepstral Coefficients is a feature which represents the spectrum bands according to the mel-scale, that is an isophonic (mostly subjective) coefficient.

Nearest Neighbor Classifier

After the extraction of the features two groups of vector are initialized, one for the train of the model the other one for the test.
Each group is composed by one vector containing the feature vectors and another vector containing the labels used to verify the test.
In this case three different genres have been taken (ambient, house and minimal) with three songs each for the train and three for the test.
The classifier works using the kNN algorithm. When a new test is proposed it calculates the euclidean distance between the k nearest neighbours according to the euclidean distance in a multi-dimensional space, where the value for every dimension is given by the value of every single feature used in the predictor.
When the k neighbour are identified it takes place what is called a Majority Vote: the new test will be labeled according to the label which appears most between the k neighbours.
Trying with different values for k, it's possible to optimize the results, in this case I run the script with different intervals of k [1, 2, 4, 8, 10, 20].
I decided to divide the extracted features in three different groups using different frame size for the extraction (20ms, 40ms, 100ms). This may cause a lower percentage for the final recognition rate, because for each execution we are using 6 songs instead of the all 18 for every execution.

Results

The lowest recognition goes around 35.4% with k=2 and 40ms frames for the time domain features, 36.48% for frequency domain features with k=1 and 100ms frames, 36.45% for time and frequency domain features together. The best recognition rate is not this high because dividing the song in 3 different groups, make more difficult for the algorithm to “recognize” the genres. The best rate is when the frame is smaller and k is bigger, according to this we reached 45.81% for time domain features, 46.71% for the frequency domain features and 46.8% for time and frequency domain, all of them with 20ms frames and k=20.