EQ2341 Pattern Recognition and Machine Learning

Assignment 2 Song Recognition

Jiaojiao Wu
Antoine Camus

KTH Royal Institute of Technology

01-05-2020

Contents

1. Description

This assignment focuses on song recognition, the main task is to design a feature extractor that can do melody recognition and check if it can work well for the given examples and the design requirements.

The **GetMusicFeatures** file can extract the pitch and the energy contours of the melody in sound, it returns a matrix containing the pitch, the correlations, and the intensity estimations. It is used to create features for melody recognition.

$$\text{frIsequence} = \begin{pmatrix} f_1 & f_2 & \dots & f_T \\ \tau_1 & \tau_2 & \dots & \tau_T \\ I_1 & I_2 & \dots & I_T \end{pmatrix}$$

Each column represents one frame in the analysis. Elements in the first row are pitch estimates in Hz (80–1100 Hz), the second row estimates the correlation coefficient (rho) between adjacent pitch periods, while the third row contains corresponding estimates of per-sample intensity. The features matrix provides almost all the data needed for feature extraction.

2. Main tasks

2.1 Get pitch and intensity profiles of the three recordings

There are three files given in songs.zip file, with two from the same melody, melodies 1 and 2, and one from another song, melody 3. Melody 2 has a higher intensity than melody 1. Before doing feature recognition, it is necessary to get the overview of the three files. By plotting the pitch and intensity of each song, we now have a general understanding of the song, it would be useful to check of the feature extractor correctly works.

Figure 1 shows pitch and intensity of the first song file **melody1.m**.

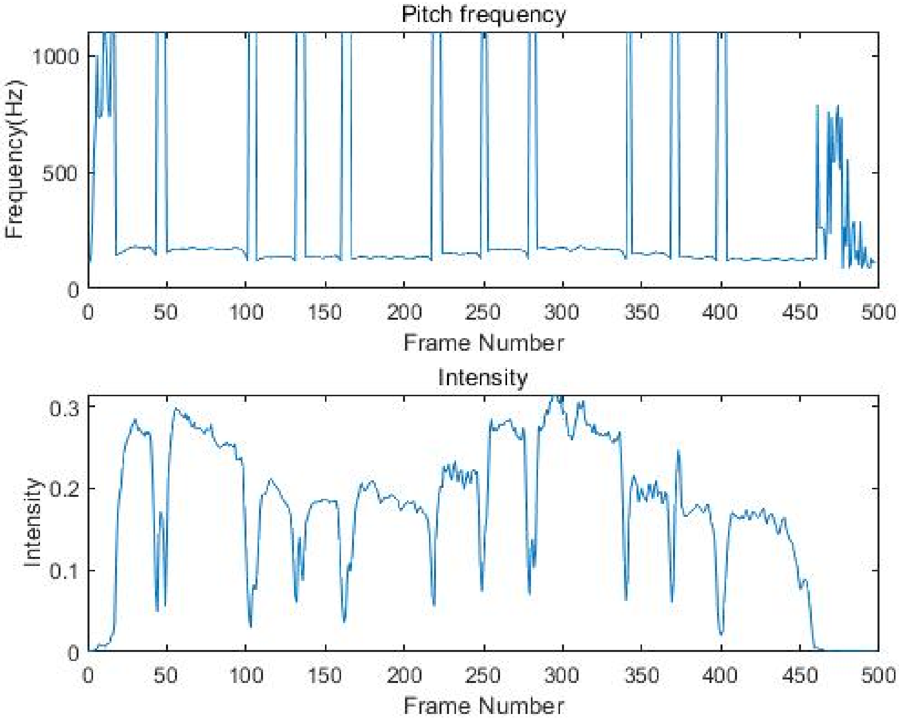

Figure 2.1: pitch and intensity of melody$_1$.wav

Figure 2 shows pitch and intensity of the second song file **melody2.m**.

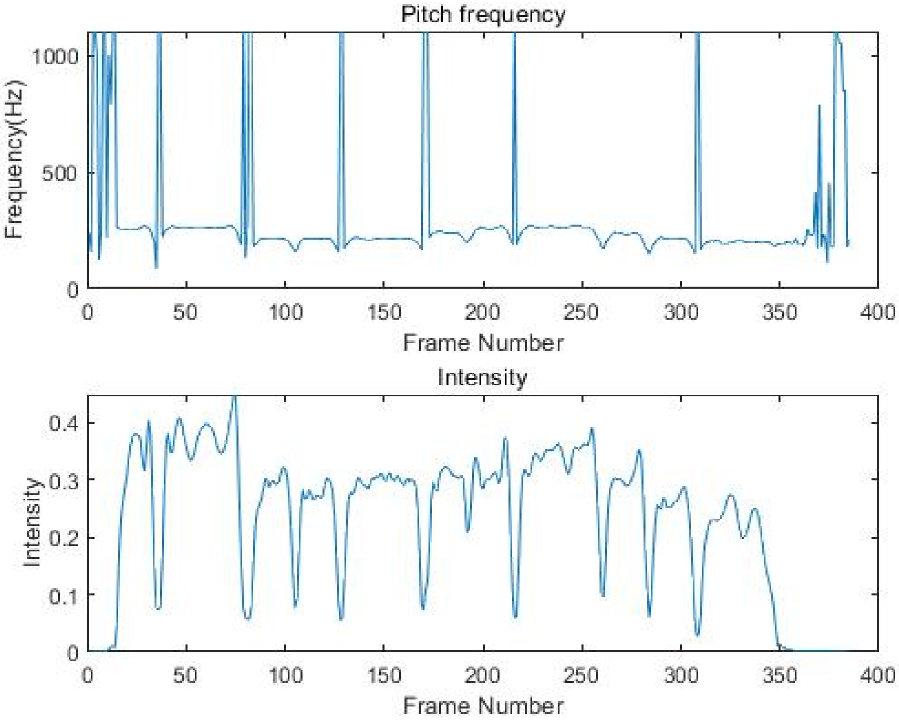

Figure 2.2: pitch and intensity of melody$_2.wav$

Figure 3 shows pitch and intensity of the third song file **melody3.m**.

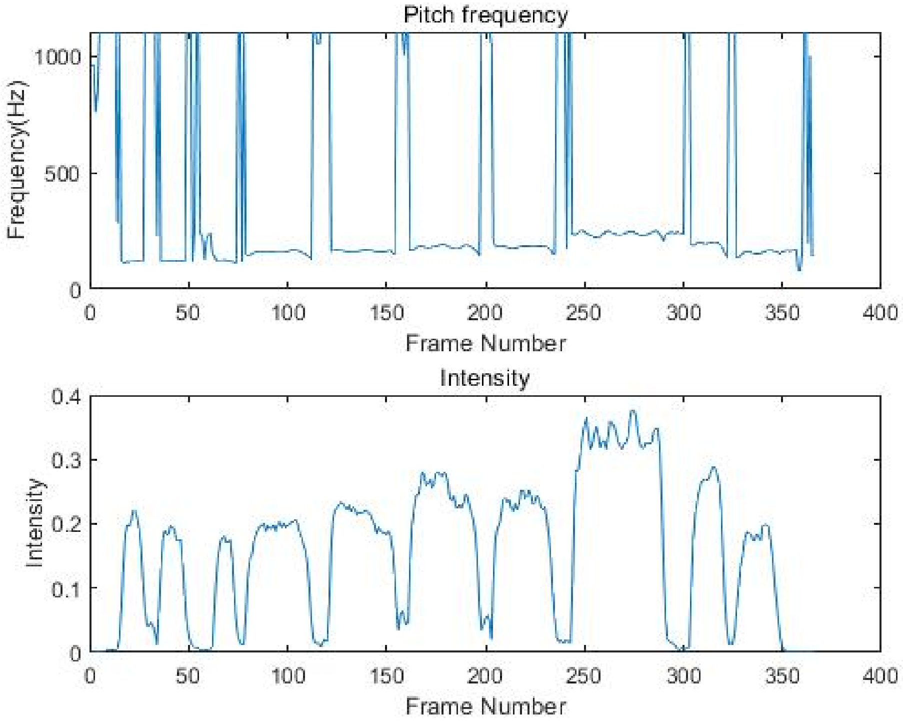

Figure 2.3: pitch and intensity of melody$_3$.*wav*

2.2 Design the feature extractor

The feature extractor is designed by two main steps. First is to distinguish between voiced and silence. It divides the whole song into two parts. Then the second step transforms the frequency of voiced segments into semitones.

A melody is formed by concatenating a number of stretches of various semitones with different duration, potentially with silent segments in between, to form a sequence of notes. So the semitones and the silent segments are the main features in a melody, to perform efficient melody recognition, it is necessary to separate the two parts. There are many ways to do classification, such as K-means clustering, which using unsupervised learning and then get the identified categories.

While using K-means clustering to divide the melody into 2 clusters, the pitches, correlation coefficient and the intensity of each frame are assumed as feature data, which is returned in **GetMusicFeatures** function. To do the classification, use the return vector of **kmeans** function in MatLab, the vector **u** consists of two values, each value means one category, thus the matrix returned in **GetMusicFeatures** function can be divide into two clusters based on the value and position of vector **u**, one is the silent segments, another be the melody part. For the melodies, the most common tuning system should be the twelve-tone equal temperament, which divides the octave into 12 parts, all of which are equal on a logarithmic scale, with a ratio equal to the 12th root of 2. Using the twelve-tone equal temperament, the melodies can be divided into semitones with continuous output.

2.3 Check the Feature Extractor

2.3.1 Test of extracted features

We use the feature extractor to test the three given songs. We plot the semitones figure based on window frame, the figure can be seen as fig. 2.4, which has the same properties as the original features.

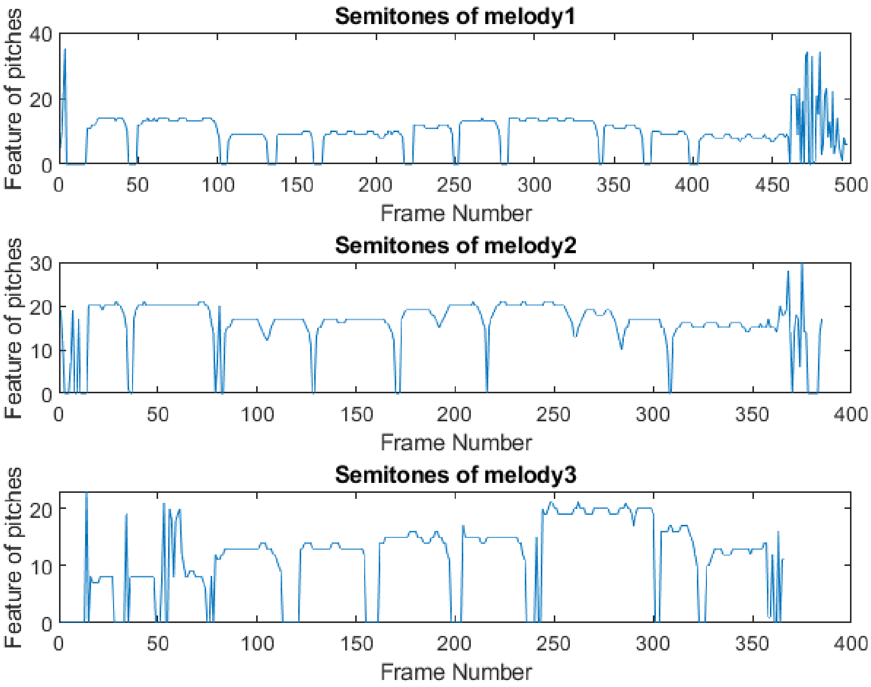

Figure 2.4: Semitones of three songs

2.3.2 Test of feature output with a transposed pitch track

The feature extractor should also be insensitive to transposition. So we multiply the pitch track returned by **GetMusicFeatures** by 1.5, and use this for the feature extractor.

The output of the semitones after transposition is shown in the figures below, with fig. 2.5. We can observe and compare the pitch frequency, the intensity and the semitones of the melody 1. Fig. 2.6 and fig. 2.7 shows the attributes of the second and third songs respectively.

From the figures we can see that the feature outputs between a melody and its transposed pitch track are consistent with the original one.

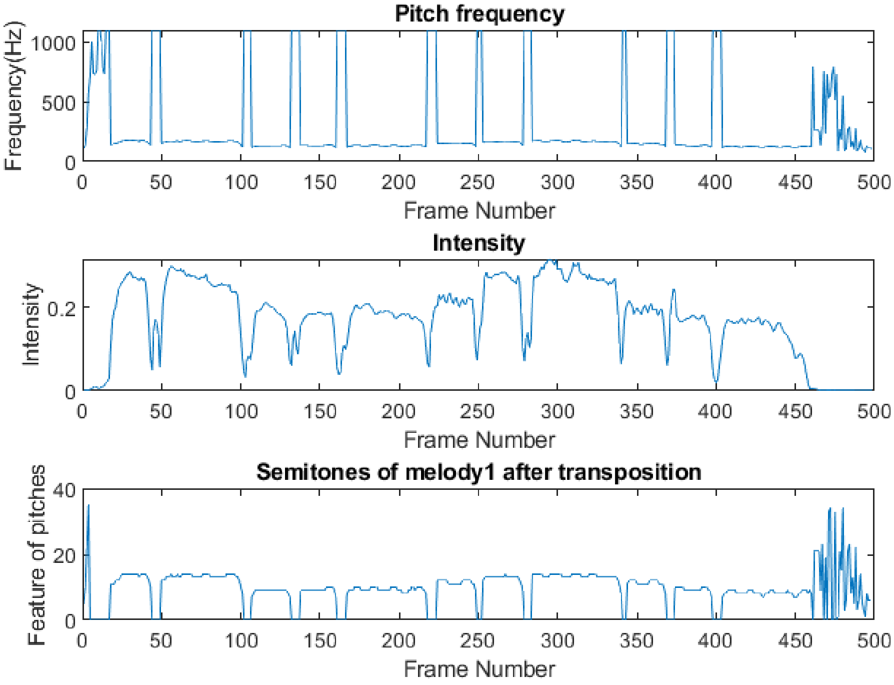

Figure 2.5: Semitones of melody 1 after transposition

2.4 Analyse the feature extraction system

For the designed feature extraction system, it uses the pitches, correlation coefficient and the intensity of each frame to distinguish the song into two clusters. By using k-means clustering, which is an linear classification, it can separate the whole song into two clusters, and then use

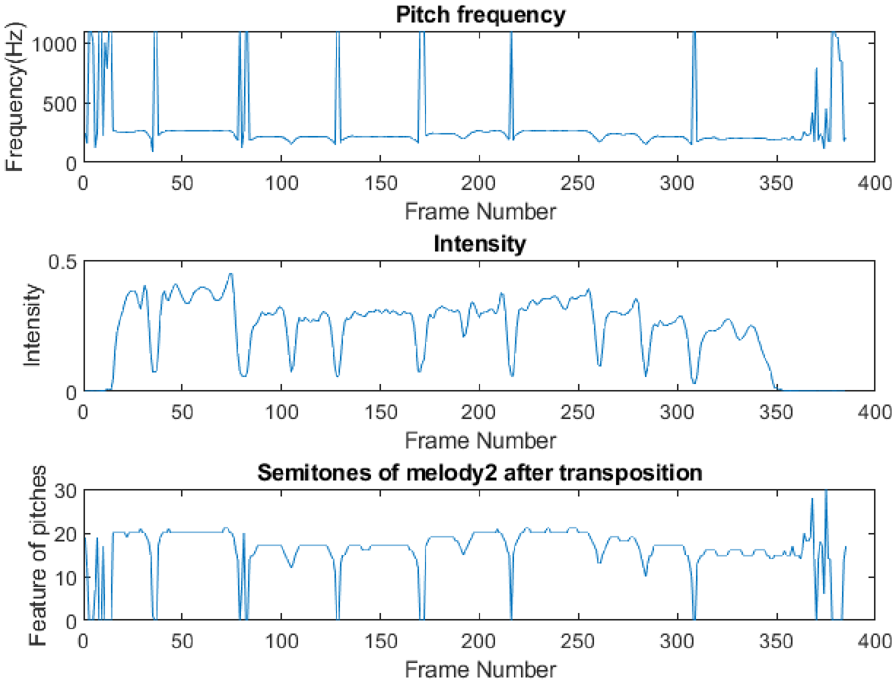

Figure 2.6: Semitones of melody 2 after transposition

the twelve-tone equal temperament to transfer pitches into semitones. It has very good robustness as categories have been settled by the value of k in **kmeans** function before extraction. It is not particularly sensitive if the same melody is played at a different volume, as well as brief episodes.

However, if there exist too higher pitches or noises mixed in the melodies, then the 2 clusters extractors cannot work well. And if there is a long-time song, k-means clustering would do lots of calculations and the feature extraction time would be very long, which means the efficiency of the extractor will become very poor.

The feature extraction is based on the pitches, correlation coefficient and intensity of each window, this means this feature extraction method does not rely on timbre. Only the pitch and volume of the sound are used in it. Thus, it doesn't change if we use different instruments to generate such song or if someone is singing.

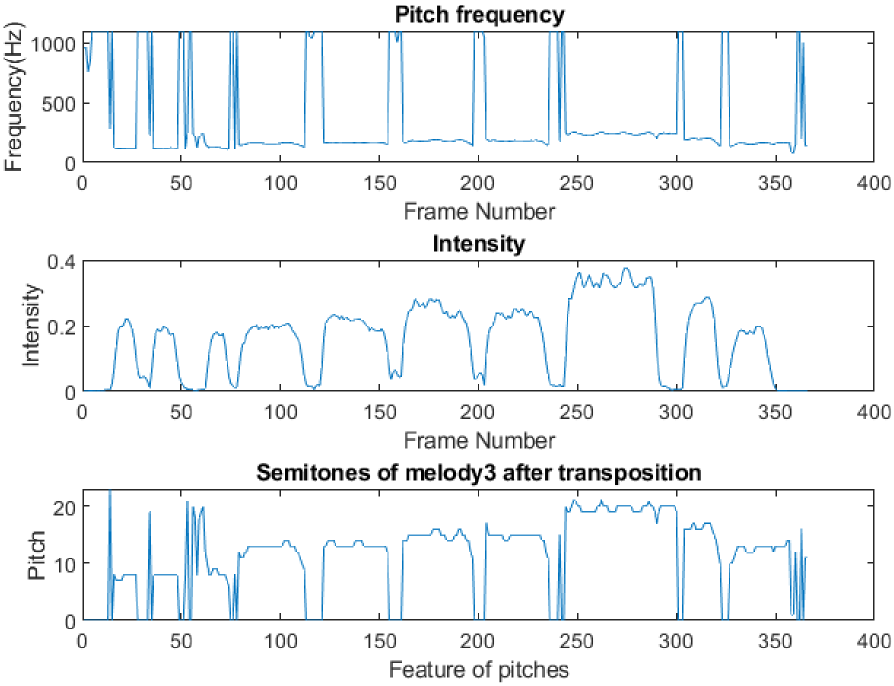

Figure 2.7: Semitones of melody 3 after transposition

2.5 Discussion about the choice of the output distribution

Since the values of the pitches are integers, we can consider that the @Discrete output distribution as the best. But, for each note, we can observe on the Fig. 2.6 its pitch is around a mean. Therefore we can choose the @GaussianMix output distribution because, in each melody, many notes and the output are continuous. It will be a crucial decision for the data training.