Chordify - A Musical Chord Recognizer

Vishal Raj Dutta, *20150115* & Aditya Adhikary, *2015007*

***Abstract*—The aim of this project is to build a classifier to classify different music samples into a predefined set of chords. We attempt Automatic Chord Extraction, which is the task of assigning chord labels and boundaries to a piece of musical audio, with minimal human involvement.**

I. Introduction

CHORD recognition is the process of detecting a chord from a piece of audio. The transcription of chords has been carried out manually which is tiresome, time-consuming and involves the knowledge of music. **Pitch** is defined as the perceptual ordering of sounds on a frequency scale, and is approximately proportional to the logarithm of frequency. Pitches can be described as a combination of letters and numbers, where each pitch comes from a set of 12 pitch classes = {C, C#, D, D#, E, F, F#, G, G# A, A#, B#}. A **Chord** is 3 or more pitches sounded simultaneously or functioning as if sounded simultaneously.

II. Progress

A. Dataset

We have collected the MIREX 2008 dataset, which consists of 180 songs from 12 albums of the Beatles in .mp3 format, and an annotation of the same from the Isophonics website. Of these annotations, we are only considering the chord annotations per frame.These frames are essentially time chunks. Labelling of frames has been conducted by expert knowledge. A small sample of the annotation is given below:

11.459070 12.921927 A
12.921927 17.443474 E

where the first two integers are the starting and ending times of the frame and the third is the chord label.

B. Preprocessing

We preprocessed the raw mp3 files by loading it into a numpy matrix as a floating point time series. We then removed the percussive frequencies by using a decomposition algorithm which carried out Median-filtering harmonic percussive source separation (HPSS). This is because percussive frequencies do not contribute to the chord, but harmonic frequencies do. Then, we segmented the resulting matrix into frames depending on a time window like 500 milliseconds. Thus, each frame can be considered to be a separate training sample. We then labeled each frame by looking at the chord which that frame has been annotated with for the maximum amount of time in that frame. For example, if the frame happened to fall in the middle of two frames in the annotation, we choose the chord label which occurs for more time out of the two.

C. Feature Extraction

The chroma vector or pitch class profile (PCP) is the most commonly used signal representation for musical harmonic analysis. The PCP feature vector represents the sound energy in each of the twelve pitch classes, and are typically derived by mapping each frequency spectrum to a corresponding pitch class. The input signal is broken into fragments and converted to the frequency spectrum by Discrete Fourier Transform (DFT), transforming it from the time domain to frequency domain. Then each frequency spectrum is mapped to the corresponding pitch class. For our purpose, we have used the Constant-Q-Transform (CQT), which is better suited to musical data. We have also applied the Short-time Fourier transform (STFT) and other techniques for feature extraction. We then apply PCA/LDA on the resultant vector (NxW, where N is the no. of frames and W is the frame window) for dimensionality reduction.

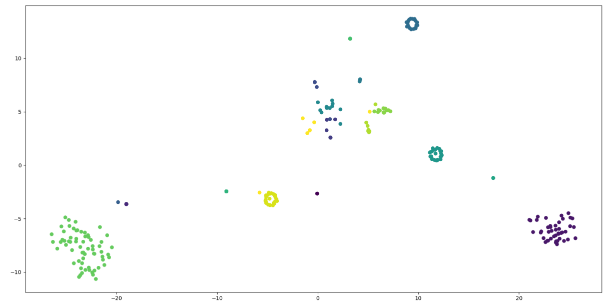

Fig. 1. LDA TSNE visualization

D. Evaluation Metrics and Baseline Classifier

One of the simplest evaluation metrics we have used (per song) is **Relative Correct Overlap**, given by

$$RCO = \frac{|correctly\_identified\_frames|}{|total\_frames|}$$

If we average the RCO over every song, we get the Average RCO (micro-averaged). Or, we take the average over all frames present in all the songs, we get the Total Relative Correct Overlap (macro-averaged).
We have used a simple correlation based technique first, in which for each chord, we calculate the mean feature vector. Then for each frame, we assign that chord label which has the highest correlation with that frame.
We have also used a Gaussian Naive Bayes classifier as a baseline.

E. Results

We have so far achieved an ARCO value of 0.45 and TRCO of 0.467 over a subset of all the songs. This is using the correlation-based classifier on the vectors obtained from the CQT chromagram, and using LDA for dimensionality reduction of the samples, by keeping the time window as 500 milliseconds.This is noticeably lesser than the state-of-the-art (Chroma,HMM) which has an ARCO value of 0.7957 and TRCO value of 0.8091 .

III. Conclusion and Todos

We have so far successfully pre-processed the dataset, carried out preliminary feature extraction and dimensionality reduction methods,

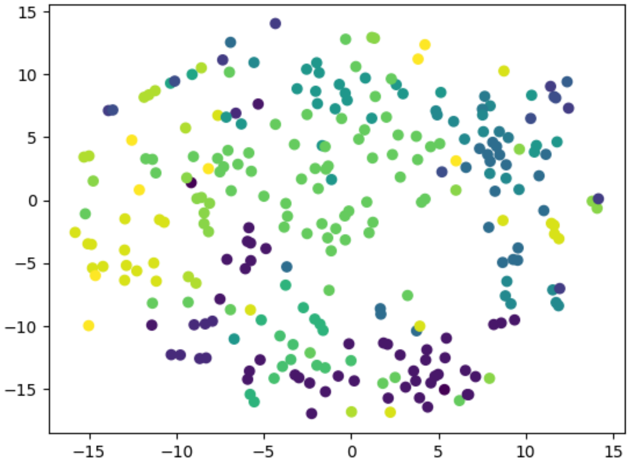

Fig. 2. STFT TSNE visualization

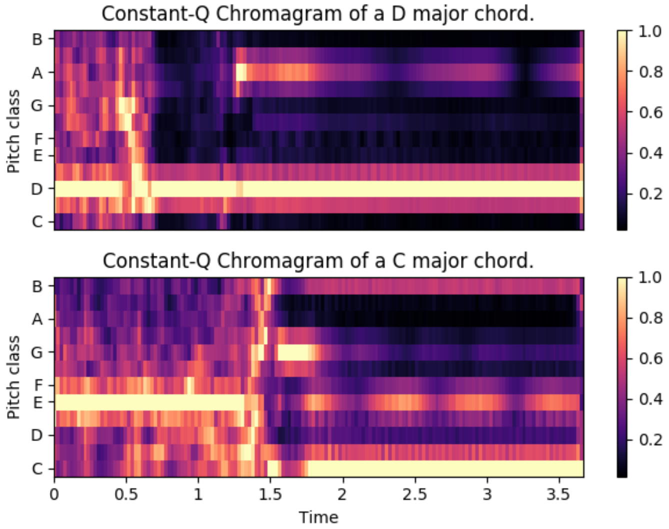

Fig. 3. Chromagram of C and D chord

and obtained our basic objective of the classification of individual chords independent of their sequence and order of occurence. In future, we have to work on incorporating this sequential influence of chords on each other by using an HMM model. We have to also use ensemble classifiers, and need to work on tuning the parameters of the classifiers.

REFERENCES

[1] A Machine Learning Approach to Automatic Chord Extraction, Matthew McVicar
[2] Automatic Chord Recognition for Music classification and Retrieval
[3] Improving Pitch Class Profile for Musical Chords Recognition Combining Major Chord Filters and Convolution Neural Networks