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. Problem Statement and Motivation

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.

The problem statement can be stated as : Given a music audio, can we determine the sequence of chords present in that it?

II. Literature Review

We have been following a thesis entitled "A Machine Learning Approach to Automatic Chord Extraction", by Matthew McVicar. It goes into great detail about chords, musical function, feature extraction techniques, and describes a number of models such as Hidden Markov Models, Dynamic Bayesian Networks, Language Models and so on. Another paper titled "Automatic Chord Recognition for Music Classification and Retrieval" by Cheng et. Al describes an N-gram model to classify chord progressions. The paper "Neural Networks for Musical Chords Recognition" (Osmalskyj et. Al) presents an effective machine learning based method using a feed-forward neural network for chord recognition. A more recent paper, "Improving Pitch Class Profile for Musical Chords Recognition Combining Major Chord Filters and Convolution Neural Networks" uses deep learning approaches and achieves improvements of more than 40% in classification accuracy as compared to base line methods.

III. 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.

IV. Approach

A. 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.

B. 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.

C. 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.

D. Other classifiers

The previous methods have considered only the simple multi-class classification problem, where the sequential nature of chords do not matter. In reality, chords are closely related to scales and are more than often depend heavily on the previous chords which came in time. Hence, some more classifiers we used were:

LSTM RNN: A Long-Short Term Memory or LSTM Network is a special kind of Recurrent Neural Network (RNN) consisting of LSTM units. These units are essential in order to connect previous information to the present task; in our case, identifying a new chord based on a sequence of previously occurring chords. An LSTM unit usually consists of a cell (for memorizing values over an arbitrary period of time), an input gate, output gate and forget gate. The three gates are each neurons, i.e they compute an activation function of a weighted sum.

Structured Perceptron: For structured prediction problems like that of sequence labeling in NLP, such as Parts-of-Speech tagging (where we try to find a sequence of POS tags y for a given input sentence x so that each tag in y corresponds to one word in x), a structured perceptron model may be useful. The structured perceptron differs from the standard perceptron algorithm as it deals with sequences. The first stage requires selects a path through a sequence (eg. with the Viterbi algorithm). In the second stage, for every position in the predicted path that is correct, weights are not changed. For every position in the path that is incorrect, weights are adjusted so that those that contributed to the wrong prediction are decreased, and those that didn't contribute towards the correct prediction enough are increased.

E. Discussion on Approach

LSTMs are well-known to classify, process and predict time series given time lags of unknown size and duration between important events. In our case, the occurrence of a chord is usually dependent on the occurrence of a few chords before it, such as when a chorus is repeated in a song. Hence, this is ideal to model the dependence of time when looking at the sequence of chords.
Structured Perceptron can be used in our case because the input is a sequence. The output is highly structured in nature, since a chord label can be represented as a 12 dimensional binary vector, with those positions marked as 1 which contribute notes towards the chord. We could have also used an HMM based model in our approach, but we tried the above due to its novelty and simplicity.

V. Results

We achieved a baseline ARCO value of 0.45 and TRCO of 0.467 over a subset of all the songs 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 was noticeably lesser than the state-of-the-art (Chroma,HMM) which has an ARCO value of 0.7957 and TRCO value of 0.8091.
Using our LSTM based approach, we got a TRCO of 0.78, hamming score of 0.90, validation TRCO score of 0.608, hamming score of validation set 0.87, validation AUC = 0.8437, training AUC = 0.89, testing TRCO = 0.67, hamming score of test = 0.9613, AUC of test = 0.875. For our Structured Perceptron model, the testing TRCO is 0.587.

[No ARCO ,RCO could be calculated as sequences were mixed across different songs.]
Training set size : 11035, Validation set size : 4208, Validation set size : 4208, Number of samples in a sequence :32

VI. Analysis of Results

Compared to the previous baseline, the RNN+LSTM model has a much better performance in determining correct sequences of chords. Thus we can arguably say that the RNN LSTM model is better performing than the structured perceptron. Given the high training

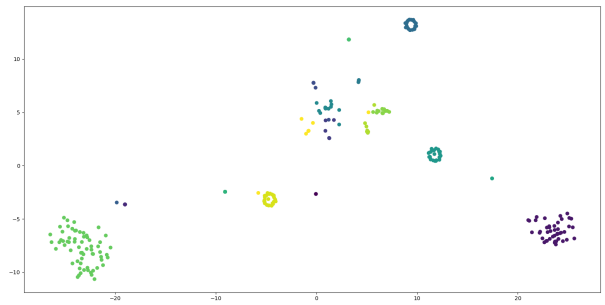

Fig. 1. LDA TSNE visualization

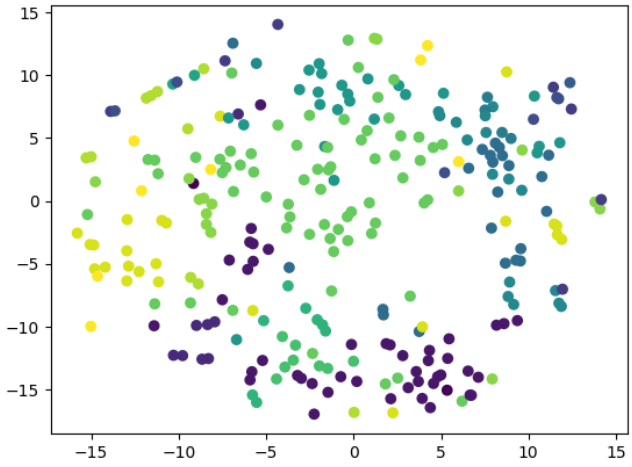

Fig. 2. STFT TSNE visualization

TRCO compared to the testing TRCO of RNN-LSTM one could say that the model tends to over fit the data plenty.To reduce overfitting , dropout was also used in a 2-layer LSTM setting.However, the testing gain observed was marginal for the same number of epochs.

VII. Work Distribution

Vishal - LSTM, Baseline classifier etc
Aditya - Feature Extraction, Structured Perceptron etc

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

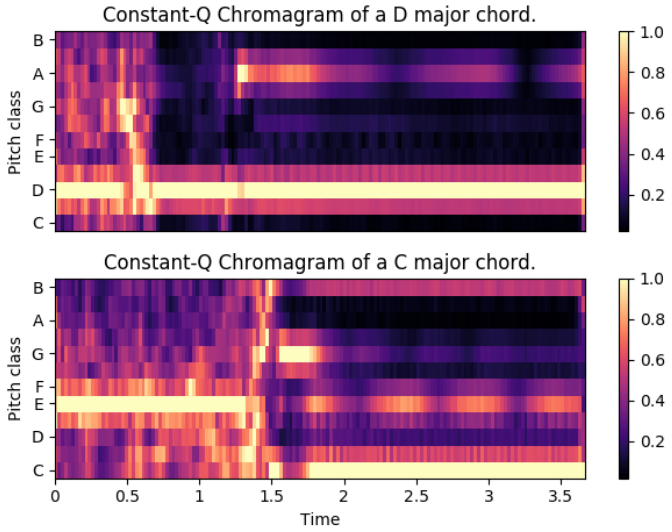

Fig. 3. Chromagram of C and D chord

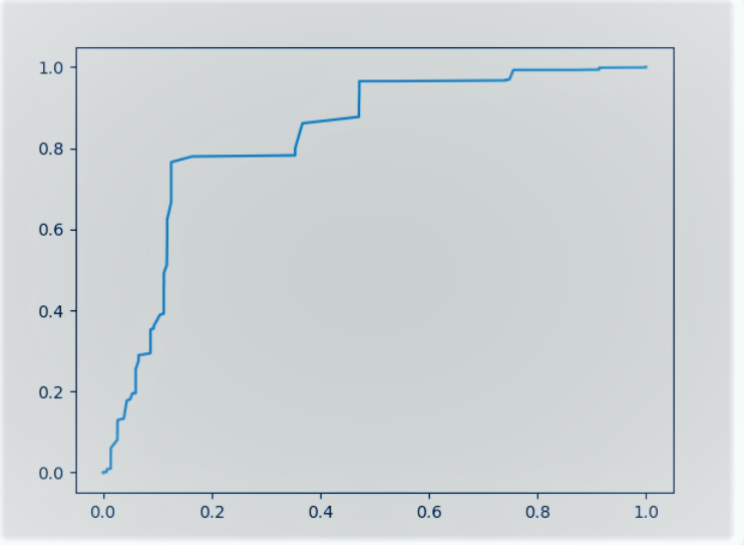

Fig. 4. ROC Plot of test data