Capstone Project Proposal

Machine Learning Nanodegree

Patrick Marshall

October 22nd, 2017

Accurate chord recognition in music using neural networks

Domain Background

The aim of my capstone project is to do automated and accurate chord recognition in a music file, for instruments such as guitar or piano. Indeed, except for very well-known international tunes and tunes which chord progression is published by the author/composer, it is often really hard to grasp a good accurate chord transcription when trying to learn a new song. Most of the time, transcriptions for little-known tracks or live performances do not exist in the web's largest chords databases (i.e https://tabs.ultimate-guitar.com). For some other songs, even good musicians tend to hear different chords from one another.

MIR (Music Information Retrieval) laboratories around the world such as LabRosa are doing research on a vast number of topics regarding music. A competition called MIREX[1], takes place every year in which these labs compete and covers a large number of topics regarding machine learning applied to music.

Chord detection solutions based on dictionaries of chroma features have already existed for years, such as the great Chordino plugin for the Sonic Visualizer software[2]. As in recent years, machine learning and deep learning have been applied successfully in various fields, the accuracy of current techniques is being challenged again. As a lot of work has been done on the speech recognition topic, but, to my mind, its results and techniques (such as the use of deep learning for time series analysis) can be used successfully on other topics such as this one.

Problem Statement

As a capstone project, I would like to craft a solution based on machine learning techniques learnt throughout the nanodegree that would be able to identify chords automatically based on the audio features of the track. Given an audio track (in *.mp3 or *.flac format) split into time segments, my aim is to produce readable output containing a list of the estimated chords heard within those intervals (more or less like a movie subtitle file).

[1] http://www.music-ir.org/mirex/wiki/MIREX_HOME

[2] http://www.vamp-plugins.org/

This problem consists in correctly identifying a chord given a segment, which is a multi-label classification (as each chord is a potential output class).

Datasets and Inputs

As I intend to base my work on the MIREX competition's rules[3], I will use the public datasets on which every contestant's submission must be based. Those datasets are the following:

Billboard

This dataset contains a subset of the Billboard dataset from McGill University, which originally contains samples of American popular music from the 1950s through the 1990s. Audio features, as well as chord annotation are available from http://billboard.music.mcgill.ca. I will use this dataset for training purposes.

25 audio features are dispatched in two *.csv files for each of the 890 annotated singles that were extracted from original audio files using the NNLS Chroma plugin[4]. Each song is divided into multiple frames: the number varies according to the sampling frequency (approximately 3000 rows for a 3min song), which yields 2,5 to 3 million rows in total.

Given the fact that the MIREX competition has been going on for a certain number of years, researchers have come up with a specific file format that defines what chord can be heard in a certain time interval: the labelling is done in *.lab format. Just like subtitles files, it is basically a text file, each line representing a time segment and the textual name of the chord that can be heard at that time:

```
...
41.2631021 44.2456460 B
44.2456460 45.7201230 E
45.7201230 47.2061900 E:7/3
47.2061900 48.6922670 A
48.6922670 50.1551240 A:min/b3
...
```

Isophonics (not sure, maybe for testing)

I will also try to use the Beatles and Queen subsets of the Isophonics datasets[5] for testing purposes. This dataset is provided by the Centre for Digital Music at Queen Mary, University of London, and has been used for Audio Chord Estimation in MIREX for many years. The only issue with this dataset is that the features have to be extracted manually using the NNLS plugin, given the album references.

NOTE: if this process turns out to be too long, I will use the Billboard dataset only, split differently.

The download procedure for both datasets can be found in the README file of this submission.

[3] http://www.music-ir.org/mirex/wiki/2017:Audio_Chord_Estimation
[4] http://www.isophonics.net/nnls-chroma

[5] http://www.isophonics.net

Solution Statement

A solution to the problem would be to apply supervised learning techniques over a set of features for each audio segment of the track, and determine a maximum likelihood class given a multi-class output.

The first phase of the analysis is to compile and gather all these features from the audio, and then train a MLP network over this data, using the annotation work that has been done by music research labs.

Another extra track to explore is to run a convolutional neural network over time series data (i.e raw sound spectral data) to identify patterns that would lead to chord identification.

Evaluation Metrics

The metric used to evaluate the quality of an automatic transcription is defined by the MIREX rules as the following:

$$\mathrm{CSR} = \frac{\mathrm{totaldurationofsegmentswhereannotationequalsestimation}}{\mathrm{totaldurationofannotatedsegments}}$$

The metric I intend to use is a simple extension of this metric, i.e. the proportion of correctly predicted chords over the list of time intervals in the track. It is called WCSR (Weighted Chord Symbol Recall). As having a correctly annotated segment does not make much sense, this metric basically measures how accurately a whole song's transcription is predicted. The model's overall accuracy will be measured as a mean of all songs' WCSR. Below is a quick example:

```
Start      End        Detected     Annoted
41.2631021 44.2456460 B            B
44.2456460 45.7201230 E            Emin
45.7201230 47.2061900 E:7/3        E:7/3
47.2061900 48.6922670 A            A
48.6922670 50.1551240 A:min/b3      A:min/b3

WCSR = 80%
```

Benchmark Model

I intend to benchmark my model with cross-validation, i.e. to use a 80% of the dataset for training and 20% of it for testing.

To emphasize on the potential benefits brought by the use of neural networks, I intend to train at least one other supervised learning classification model (ex: SVM) on the data.

I will also compare those scores to the ones obtained by the different teams doing the real competition[6], which are downloadable as a csv file.

[6] http://www.music-ir.org/mirex/wiki/2017:Audio_Chord_Estimation_Results

Project Design

My approach to solve the problem would be the following:

1) Preprocessing phase

I will craft a preprocessing flow to extract time-series-like features from multiple sources: the ones from NNLS-Chroma spectral analysis, to which I will append additional features extracted by the excellent LibRosa[7] and Yaafe[8] Python packages.

On top of these features, I will set a process that defines vectors containing output labels, i.e. chord annotations. These chords will have to belong to one of the predefined chord classes:

{A, B, C, D, E, F, G} x {N, min, maj, 7th}, to which I will add an 'unidentified' class.

This phase's aim is to get four perfectly clean matrices and output vectors *X_train, X_test, Y_train* and *Y_test*.

Amazing work has also already been done by Honglak Lee and al[9] as to perform unsupervised feature learning using CDB (Convolutional Deep Belief) networks, which should be a track to explore.

2) Model definition and crafting

Next step is to define a neural network architecture as well as another that will be trained using the previously-defined matrices and vectors.

Then, I will build a post-processing flow that takes the argmax of the probabilities of a chord being heard within a time interval. I will add a quick computation that aggregates all the output and CSR scores related to a given track, and provide the overall WCSR metric defined earlier for each song.

3) Training/testing/benchmarking/refining

Refining of the model will include testing multiple neural network architectures, adding or removing layers. I will also make use of the set of techniques I learnt during the course to reduce overfitting, prevent vanishing gradient issues, prevent dimensionality-related issues while not losing too much information.

I will also provide quick visualization about the model's most relevant features to identify a chord.

[7] https://github.com/librosa/librosa

[8] http://yaafe.sourceforge.net/

[9] https://papers.nips.cc/paper/3674-unsupervised-feature-learning-for-audio-classification-using-convolutional-deep-belief-networks

4) Extra: embed the model in an end-to-end solution

If I have some extra time, I will try and embed the python code from previous steps in an end-to-end executable process that will be capable of extracting the features, running the model and provide a complete output file with the predicted chords in the right time order.