A Supervised Approach To Musical Chord Recognition

Pranav Rajpurkar PRANAVSR@STANFORD.EDU
Brad Girardeau BGIRARDE@STANFORD.EDU
Takatoki Migimatsu TAKATOKI@STANFORD.EDU
Stanford University, Stanford, CA 94305 USA

Abstract

In this paper, we present a prototype of an online tool for real-time chord recognition. It makes use of a Hidden Markov Model in conjunction with Gaussian Discriminant Analysis. Unlike approaches to collect data through web-scraping or training on hand-labeled song data, we generate a copious amount of chord data programmatically. We improve the performance of system by substituting the usually tried Chroma features with a novel set of Chroma DCT-Reduced log Pitch features to push test accuracy on clean data to 99.96%. We finally propose a set of modifications to have the online system achieve a good balance between speed and accuracy.

1. Introduction

There is significant value in an automated tool to determine chords from audio. Knowing the progressions of chords underlying the melodies is an essential part of understanding, playing, and building on the music. To a curious learner of music, such a tool creates the opportunity to be able to play a new pop song without requiring good-quality hand-labelled chord tags. Equally useful to a learner is being able to receive feedback concerning the accuracy with which a chord was played, making such a system a good automated feedback tool, capable of being plugged in to an online MOOC class. To a song writer, the system is useful in being able to explore chords supporting the melodic content of the song.

Furthermore, the use of such a system extends into other machine learning tasks. The tasks of identifying a song from its waveform data, and of classifying the genre of song can be linked to finding the chord progressions underlying the harmonic content of the song. Hand-labelling chord names and marking chord changes in a song takes a lot of manual time and effort. An automated tool for this process would save such time, and would allow the development of new musical tools and research.

Most exisiting research in the field of chord recognition has been designed for offline processing. There are a few realtime implementations available https://files.nyu.edu/jb2843/public/Publications_files/Cho_Bello_ICMC2009.pdf and Idea of generatic audio from symbolic data http://dl.acm.org/citation.cfm?id=1178726 But we are first to do it online.

2. Data Generation

The chord prediction task is a multiclass classification task. In music, a chord is a harmonic set of notes sounding simultaneously. We choose the minor and major chords, the two most common sets of chords in popular music to classify on. Using the traditional twelve pitch scale (C, C#, D, D#, E, F, F#, G, G#, A, A#, B), we have 24 such distinct chords.

There are different ways of playing the same chord. The Cmaj chord, for instance, is the set of three notes C, E, and G played simultaneously. On a piano, these notes can be played on different octaves. Played on the fourth octave, the Cmaj would have the notes C4, E4, G4. It is also possible to play Cmaj in with E as the lowest note - E4, G4, C5 (first inversion form) and with G as the lowest note G4, C5, E5 (second inversion form).

To train the system, we generate training data programmatically in Python using INSERT. We generate midi files for each of the 24 chords, taking into account 8 octaves, and 3 inversion forms, to generate a total of 576 midi files. We then use these 576 midi files in

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

conjunction with 6 soundfonts for piano, guitar and violin to generate a total of 3456 sound files.

In musical chord recognition, feature extraction operates over frames. The generated wav files, which are an average of 4 seconds in length, are first split into frames of window size 10ms each, and a n-dimensional feature vector is extracted for each frame. We label each frame with the label of the chord on its corresponding sound file, to generate 1 million training examples.

3. Paralelling Speech Recognition

The pipeline of a chord recognition system is very similar to that of a speech recognition one and relies on techniques that were originally applied to speech recognition tasks. The use of the Hidden Markov Models, and the division of a sound file into frames on which the prediction task is performed, are two such techniques which have been reproduced in identifying chords. However, the task of finding chords is also different from speech tasks in a few ways, and these differences can be exploited to specialize a system in the task of chord recognition.

3.1. Feature Extraction

One important difference between the two surfaces in the choice of features for the tasks. Mel-frequency cepstrum coefficients (MFCC) are commonly used as features in speech recognition systems. These represents the short-term power spectrum of a sound. It has been found that MFCCs are closely related to timbre - quality that is associated to being able to distinguish between a voice, string instrument, wind instrument, and percussion instrument. These have traditionally be seen as bad features for chord recognition, since they discard the pitch content of the sound, and are simply used to set benchmarks for other features.

Chroma features are commonly used for chord recognition tasks. It is a representation in which the the entire spectrum of sound frequencies is distributed into 12 bins representing the traditional twelve pitch scale. An advantage of Chroma features is that they invariant to octaves and inversions. These features also possess a reasonable level of robustness to changes in instruments. We use the Matlab Chroma Toolbox to extract Chroma features for the frames.

Figure 1 shows the extract Chroma for C major. Note that the spikes at INSERT correspond to spikes at C, E, and G, the notes constituting the C major. This supports the idea that Chroma encodes the harmonic content of sound.

INSERT CHROMA IMAGE FOR C. MAYBE EVEN INVERSIONS

To test the performance of Chroma features against MFCC features, we start with a binary classification problem of distinguishing major chords from minor chords. An SVM with RBF kernel

$$K(x, z) = \exp(-\gamma \|x - y\|^2)$$

is trained, with regularization and kernel parameters ($\gamma = \frac{1}{\#\ \text{features}}$ and $C = 100$). Table 1 shows the results, and confirms that chroma features are much better suited to the task of chord recognition than MFCCs are.

, the SVM gives a test accuracy of 51.0%. Next an SVM with RBF kernel is trained using basic chroma features, and gives an accuracy of 97.7% on a test set.

4. Frame Model

Since we have a multiclass classification problem, we first use multinomial logistic regression or softmax regression to get a performance baseline. To test the accuracy of frame model, we predict the chord on each frame. Our frame accuracy on the training set gives an accuracy of 1233

5. Mixer Model

Our frame model outputs a prediction for each frame. Our final classification task, however, is on a wav file, which is a collection of frames. We now come up with a mixer model, which is a model for putting the results on the frame together. There are many ways of doing this, including looking at the middle frame, or counting the most frequent prediction across all of the frames.

Let us assume that a test wav file consists of a single chord being played. We will drop this assumption later, but for now, let us consider this simpler case. Now that we have constraint no chord changes within a wave file, we use the following model for predictions.

We assume that the prediction on each frame is independent of the prediction on other frames. Hence our prediction is $argmaxyP(y|X) = ProdP(y|x_i)$. We get a test accuracy of 1223

6. Improving the frame model

Softmax regression is a learning algorithm that models p(y —x), the conditional distribution of the chord given the extracted frame features. We now look at a model that tries to model p(x— y) and p(y): Gaussian

Discriminant Analysis (GDA). The model is: x—y = i N (i,) Since our aim is to make the system independent of any specific genre, we model the distribution of ys as unifrom $p(y) = 1/num_chords$. We also model p(x—y) using a multivariate normal distribution. Although we model each gaussian with different means, we assume they all share the same covariance matrix. With this new generative model, we can now expand our mixer model.

7. Improving the mixer model

Earlier, we had imposed the constraint that chords could not change in a wav file. However, in songs, chords do change frequently. Our next model allows us to loosen that constraint. We now use a hidden markov model to predict the chord sequence in wav files. The emission probabilities p(x—y) can be modelled by our GDA frame model, and we assume uniform chord transitions to remain flexible to any genre of music. We determine the most likely state sequence by the Viterbi algorithm.

With the new model, for a random sequence of frames, we achieve 1223

8. Improving features

The reason we are able to get very high accuracy was that our features were invariant to octave and inversions, and somewhat invariant to instrumentation. It would be possible to get even higher accuracy if features were invariant of instruments. CRP (Chroma DCT-Reduced log Pitch) is a chroma-based audio feature that boosts the degree of timbre invariance. The general idea is to discard timbre-related information similar to that expressed by MFCCs, in effect leaving the information related to pitch.

Graph showing chroma vs crp features.

With these feautres, on clean chords, training on instruments 1223, and testing on 1223 we get an accuracy of 1223. We found that training only on piano improved the performance even more.

9. Live system consideration

One of such differences is that the unit of speech that makes up a word, a phone, changes much more quickly than a chord does. A chord is usually sustained for a few centiseconds, if not a few seconds, before it changes. We make use of this knowledge in our live system to distinguish chords from noise.

In a live system, one of the most important considerations to take into account is noise. Even while running the system on songs, there are segments that consist of percussion, or silence. It is important for a system to not predict any chord in this timespan. Challenging is that we don't know what's about to come next.

References

Author, N. N. Suppressed for anonymity, 2011.

Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification.* John Wiley and Sons, 2nd edition, 2000.

Kearns, M. J. *Computational Complexity of Machine Learning.* PhD thesis, Department of Computer Science, Harvard University, 1989.

Langley, P. Crafting papers on machine learning. In Langley, Pat (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (eds.). *Machine Learning: An Artificial Intelligence Approach, Vol. I.* Tioga, Palo Alto, CA, 1983.

Mitchell, T. M. The need for biases in learning generalizations. Technical report, Computer Science Department, Rutgers University, New Brunswick, MA, 1980.

Newell, A. and Rosenbloom, P. S. Mechanisms of skill acquisition and the law of practice. In Anderson, J. R. (ed.), *Cognitive Skills and Their Acquisition*, chapter 1, pp. 1–51. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1981.

Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):211–229, 1959.