

Automatic Chord Recognition from Audio Using an HMM with Supervised Learning

Kyogu Lee

Center for Computer Research in Music and Acoustics
Department of Music, Stanford University
kglee@ccrma.stanford.edu

Malcolm Slaney

Yahoo! Research
Sunnyvale, CA94089
malcolm@ieee.org

Abstract

In this paper, we propose a novel method for obtaining labeled training data to estimate the parameters in a supervised learning model for automatic chord recognition. To this end, we perform harmonic analysis on symbolic data to generate label files. In parallel, we generate audio data from the same symbolic data, which are then provided to a machine learning algorithm along with label files to estimate model parameters. Experimental results show higher performance in frame-level chord recognition than the previous approaches.

Keywords: Chord recognition, hidden Markov model, supervised learning

1. Introduction

A musical chord is a set of simultaneous tones. Succession of chords over time, or chord progression, form the core of harmony in a piece of music. Hence analyzing the overall harmonic structure of a musical piece often starts with labeling every chord. Automatic chord labeling is very useful for those who want to do harmonic analysis of music. Once the harmonic content of a piece is known, a sequence of chords can be used for further higher-level structural analysis where phrases or forms can be defined. Chord sequences are also a good mid-level representation of musical signals for such applications as music search, music segmentation, music similarity identification, and audio thumbnailing. For these reasons and others, automatic chord recognition has recently attracted a number of researchers in the Music Information Retrieval field.

Hidden Markov models (HMMs) are very successful for speech recognition, and gigantic databases with labels accumulated over decades play an important role in estimating the model parameters appropriately. However, there is no such database available for music. Furthermore, the acoustical variance in a piece of music is even greater than that in speech in terms of its frequency range, instrumentation,

dynamics, or duration, and thus a lot more data is needed to train the models for generalization.

Hand-labeling the chord boundaries in a number of recordings is an extremely time consuming and tedious task. In this paper, we propose a method of automating this daunting task to provide the models with labeled training data. To this end, we use symbolic data such as MIDI data to generate chord names and boundaries as well as to create audio. Audio and chord boundary information generated this way are in perfect sync, and we can use them to estimate the model parameters.

There are several advantages to this approach. First, we do not need to manually annotate chord boundaries with chord names to obtain training data. Second, we can generate as much data as needed with the same notes but different musical attributes by just changing instrumentation, tempo, or dynamics when synthesizing audio. This helps avoid overfitting the models to a specific type of music. Third, sufficient training data enable us to include more chord types such as 7th, augmented, or diminished.

This paper continues with a review of related work in Section 2; in Section 3, we describe how we extract the feature vectors, and explain the model and the method of obtaining the labeled training data; in Section 4, we present empirical results followed by discussions, and draw conclusions in Section 5.

2. Related Work

Sheh and Ellis proposed a statistical learning method for chord segmentation and recognition [1]. They used the hidden Markov models (HMMs) trained by the Expectation Maximization (EM) algorithm, and treated the chord labels as hidden values within the EM framework. In training the models, they used only the chord sequence as an input to the models, and applied the forward-backward or Baum-Welch algorithm to estimate the model parameters. The frame accuracy in percent they obtained was about 76% for segmentation and about 22% for recognition, respectively. The poor performance for recognition may be due to insufficient and unlabeled training data compared with a large set of classes (20 songs for 147 chord types).

Bello and Pickens also used HMMs with the EM algorithm [2]. They incorporated musical knowledge into the models by defining a state transition matrix based on the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.
© 2006 University of Victoria

key distance in a circle of fifths, and avoided random initialization of a mean vector and a covariance matrix of observation distribution, which was modeled by a single multivariate Gaussian [2]. In addition, in training the model’s parameter, they selectively update the parameters of interest on the assumption that a chord template or distribution is almost universal, thus disallowing adjustment of distribution parameters. The accuracy thus obtained was about 75% using beat-synchronous segmentation with a smaller set of chord types (24 major/minor triads only). In particular, they argued that the accuracy increased by as much as 32% when the adjustment of the observation distribution parameters is prohibited. Again, this may be because they not only used an unsupervised model, but also the training data were insufficient to appropriately estimate the model parameters.

Our approach is based on the work of Sheh and Ellis or Bello and Pickens in that the states in the HMM represent chord types, and try to find the optimal path, *i.e.*, chord sequence in a maximum likelihood sense. The most prominent difference in our approach is, however, that we use labeled training data by which model parameters can be directly estimated. Furthermore, we propose a method of automatically obtaining the labeled training data, which removes the problematic and time consuming task of manual annotation.

3. System

Our system starts with extracting suitable feature vectors from the raw audio. Like most chord recognition systems, a chroma vector or a PCP vector is used as the feature vector.

3.1. Chroma Features

A chromagram or a Pitch Class Profile (PCP) is the choice of the feature set in automatic chord recognition or key extraction since introduced by Fujishima [3]. Perception of musical pitch involves two dimensions – *height* and *chroma*. Pitch height moves vertically in octaves telling which octave a note belongs to. On the other hand, chroma tells where it stands in relation to others within an octave. A chromagram or a pitch class profile is a 12-dimensional vector representation of a chroma, which represents the relative intensity in each of twelve semitones in a chromatic scale. Since a chord is composed of a set of tones, and its label is only determined by the position of those tones in a chroma, regardless of their heights, chroma vectors appear to be an ideal feature to represent a musical chord or a musical key.

Fujishima developed a realtime chord recognition system, where he derived a 12-dimensional pitch class profile from the DFT of the audio signal, and performed pattern matching using the binary chord type templates [3]. Gomez and Herrera proposed a system that automatically extracts from audio recordings tonal metadata such as chord, key, scale and cadence information [4]. They used as the feature vector, a Harmonic Pitch Class Profile (HPCP), which is based on Fujishima’s PCP, and correlated it with a chord

or key model adapted from Krumhansl’s cognitive study [5]. Similarly, Pauws used the maximum-key profile correlation algorithm to extract key from the raw audio data, where he averaged the chromagram features over variable-length fragments at various locations, and correlate them with the 24 major/minor key profile vectors derived by Krumhansl and Kessler [6]. Harte and Sandler used a 36-bin chromagram to find the tuning value of the input audio using the distribution of peak positions, and then derived a 12-bin, semitone-quantized chromagram to be correlated with the binary chord templates [7].

There are some variations when computing a 12-bin chromagram, but it usually follows the following steps. First, the DFT of the input signal $X(k)$ is computed, and the constant-Q transform X_{CQ} is calculated from $X(k)$, using a logarithmically spaced frequencies to reflect the frequency resolution of the human ear [8]. The frequency resolution of the constant-Q transform follows that of the equal-tempered scale, which is also logarithmically based, and the k th spectral component is defined as

$$f_k = (2^{1/B})^k f_{min}, \quad (1)$$

where f_k varies from f_{min} to an upper frequency, both of which are set by the user, and B is the number of bins in an octave in the constant Q transform. Once $X_{CQ}(k)$ is computed, a chromagram vector CH can be easily obtained as:

$$CH(b) = \sum_{m=0}^{M-1} |X_{CQ}(b + mB)|, \quad (2)$$

where $b = 1, 2, \dots, B$ is the chromagram bin index, and M is the number of octaves spanned in the constant Q spectrum. For chord recognition, only $B = 12$ is needed, but $B = 24$ or $B = 36$ is also used.

In our system, we used as feature vectors 12-bin Quantized chromagram proposed by Harte and Sandler [7], which compensates a possible mistuning present in the recordings by reallocating the peaks based on the peak distribution.

3.2. Hidden Markov Model

A hidden Markov model [9] is an extension of a discrete Markov model, in which the states are *hidden* in the sense that an underlying stochastic process is not directly observable, but can only be observed through another set of stochastic processes.

We recognize chords using a 36-state HMM. Each state represents a single chord, and the observation distribution is modeled by a single multivariate Gaussian in 12 dimensions defined by its mean vector μ_i and covariance matrix Σ_i , where i denotes i th state. We assume the features are uncorrelated with each other, and thus use diagonal covariance matrix. State transitions obey the first-order Markov

property; *i.e.*, the future is independent of the past given the present state. In addition, we use an ergodic model since we allow every possible transition from chord to chord, and yet the transition probabilities are learned.

Once the model parameters – initial state probabilities, state transition probabilities, and mean vector and covariance matrix for each state – are learned, the Viterbi algorithm is applied to the model to find the optimal path, *i.e.*, chord sequence, in a maximum likelihood sense given an input signal.

In our model, we have used 36 classes or chord types – major, minor, and diminished triads for each pitch class. We treated major and dominant seventh chords as belonging to major triads, minor sevenths to minor triads, and diminished sevenths to diminished triads. We found this class size appropriate in a sense that it lies between overfitting and oversimplification.

3.3. Labeled Training Data

In order to train a supervised model, we need label files with annotated chord boundaries. To automate this laborious process, we use symbolic data to generate label files as well as audio data. To this end, we first convert a symbolic file to a format which can be used as an input to a chord analysis tool. Chord analyzer then performs harmonic analysis and outputs a file with root information and note names from which complete chord information (*i.e.*, root and its sonority - major, minor, or diminished triad/seventh) is extracted. Sequence of chords are used as ground-truth or labels when training the HMM. In parallel, we use the same symbolic files to generate audio files using a sample-based synthesizer. Audio data generated this way are in sync with chord label files obtained above, and are enharmonically rich as in real acoustic recordings. Figure 1 illustrates the overview of the system.

4. Implementation and Experiments

As shown in Figure 1, our system for generating labeled training data has two main blocks running in parallel. First, harmonic analysis is performed on symbolic data. We used symbolic files in Humdrum data format. Humdrum is a general-purpose software system intended to help music researchers encode, manipulate, and output a wide variety of musically-pertinent representations.¹ For harmonic analysis, we used the Melisma Music Analyzer developed by Sleator and Temperley.² The Melisma Music Analyzer takes a piece of music represented by an event list, and extracts musical information from it such as meter, phrase structure, harmony, pitch-spelling, and key. By combining harmony and key information extracted by the analysis program, a complete Roman-numeral analysis is performed,

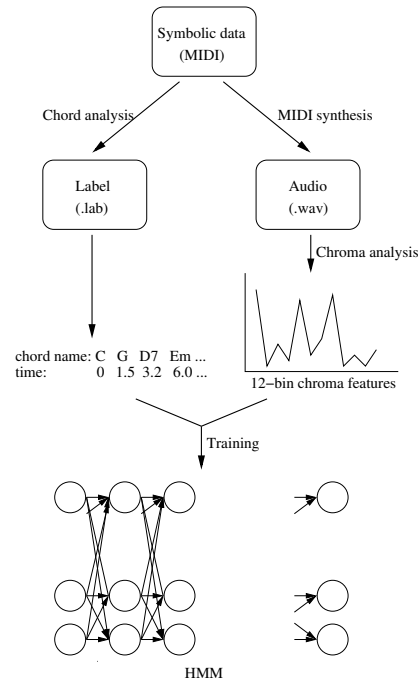


Figure 1. Overview of the system.

from which we can generate label files with sequence of chord names.

The analysis program was tested on a corpus of excerpts and the 48 fugue subjects from the *Well-Tempered Clavier*, and the harmony analysis and the key extraction yield the accuracy of 83.7% and 87.4%, respectively [10].

In the near feature extraction block in our system, MIDI files are synthesized using Timidity++ using a GUS (Glavis UltraSound) instrument patch. Timidity++ is a free software synthesizer, and converts MIDI files into audio files in a WAVE format.³ It uses a sample-based synthesis technique to generate enharmonically rich audio with upper partials as in real acoustic recordings. The raw audio is downsampled to 11025 Hz, and 12-bin chroma features are extracted from it with the frame size of 8192 samples and the hop size of 2048 samples. The chroma vectors are then used as input to the HMM along with the label files obtained above.

As a training data set, we used 175 files of Haydn’s String Quartets in a Humdrum data format at the Center for Computer Assisted Research in the Humanities at Stanford University.⁴ These files were converted to a format which can be used in the Melisma Music Analyzer as well as to a MIDI format using the tools developed by Craig Sapp.⁵ We used acoustic piano samples to generate audio. The audio data synthesized from these MIDI files is about 7.5 hours long, and contains about 145,000 frames in total.

Figure 2 shows a transition probabilities matrix and tran-

¹ <http://dactyl.som.ohio-state.edu/Humdrum/>

² <http://www.link.cs.cmu.edu/music-analysis/>

³ <http://timidity.sourceforge.net/>

⁴ <http://www.ccarh.org/>

⁵ <http://extras.humdrum.net/>

sition probabilities for C major chord estimated from the training data set. It can be observed that the transition matrix is strongly diagonal since chord duration is usually longer than the frame length, and thus the state does not change for several frames. However, chord progression based on music theory can also be found in transition probabilities, for example, in the case of C major chord. As mentioned, it has the largest probability of staying within the same state, *i.e.*, within C major chord, but has comparably higher probabilities for making a transition to specific chords like F major, G major, or D minor chord than to others, as shown in the right figure. F major and G major have fifth-tonic relationships with C major, and transitions between them happen very often in Western tonal music. C major chord is also a dominant chord of F minor, and therefore a C major to F minor transition is frequent as well.

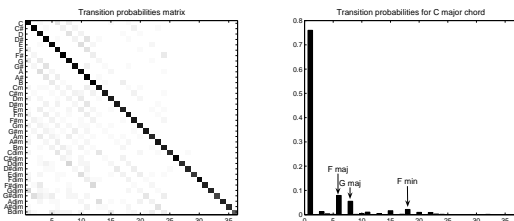


Figure 2. 36x36 transition probabilities matrix and transition probabilities for C major chord. Axes are numbered in the order of major, minor, and diminished chords.

Figure 3 exemplifies the observation distribution parameters estimated from the training data for C major chord. On the left is the mean chroma vector for C major chord. It is obvious that it has three largest peaks at chord tones or at C, E, and G, as expected. In addition, we can also see relatively large peaks at D and B, which come from the third harmonics of G and E, respectively. Covariance matrix for C major chord is also consistent with what is expected from the music theoretical knowledge. Chord tones or C, E, and G are strongly correlated with themselves whereas very low correlation was found with D#, F#, or G#.

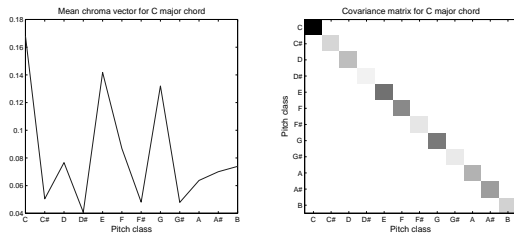


Figure 3. Estimated mean chroma vector and covariance matrix for C major chord

4.1. Empirical Results

We tested our model on the actual recording of Bach’s Prelude in C major performed by Glenn Gould. It is approximately 140 seconds long, and contains 753 frames. Test

data first goes through the chroma analysis which outputs 12-bin quantized chroma feature vectors. These feature vectors are then fed into the trained HMM. Recognition is accomplished as the Viterbi algorithm finds the optimal path given the model parameters and the input observation vectors. We compared the output of the model, which is a sequence of frame-level chord names, with the hand-marked ground-truth to make scores for frame rate accuracy.

In computing scores, we only counted exact matches as correct recognition. We tolerated the errors at the chord boundaries by having some time margins of a few frames around the boundaries. This assumption is fair since the ground-truth was generated by human by listening to a piece, which can’t be razor sharp. Figure 4 shows a recognition example from the test data.

As can be seen in Figure 4, estimated chord boundaries are very closely aligned with the ground-truth boundaries. Furthermore, all chord names are also correctly recognized. As mentioned in Section 3.2, dominant seventh chords were recognized as their root triads, which we treated as correct recognition. The overall frame-level accuracy was about 93.35%.

Except for some sporadic errors, most consistent errors in the test data came from the confusion between A minor seventh chord and C major chord. A minor seventh is composed of four notes – A, C, E, and G – in which C, E, and G are also chord tones of C major triad. Since we treated A minor triad and A minor seventh as one class, it is highly likely that A minor seventh is misrecognized as C major triad in the presence of a G note, which was the case.

Furthermore, the preceding chord was C major triad, and thus it is a most likely decision for the system to stay in the same state of C major chord rather than jumping to another state unless there is a large change in the observation vectors. We expect that the system will be less sensitive to this sort of confusion if we increase the class size to include seventh chords and train our model on more data.

It is hard to directly compare performance of our system with previous work since we are using different type of music for testing as well as for training. But we believe our high performance, when training on synthetic pieces and testing on live recordings, will only get better as we add more pieces to our training collection and add additional instrumentations.

5. Conclusion

The main contribution of this work is the automatic generation of labeled training data for a machine learning model for automatic chord recognition. By using the chord labels with explicit segmentation information, we directly estimate the model parameters in an HMM.

In order to accomplish this goal, we have used symbolic data to generate label files as well as to create audio files. The rationale behind this idea was that it is far easier and

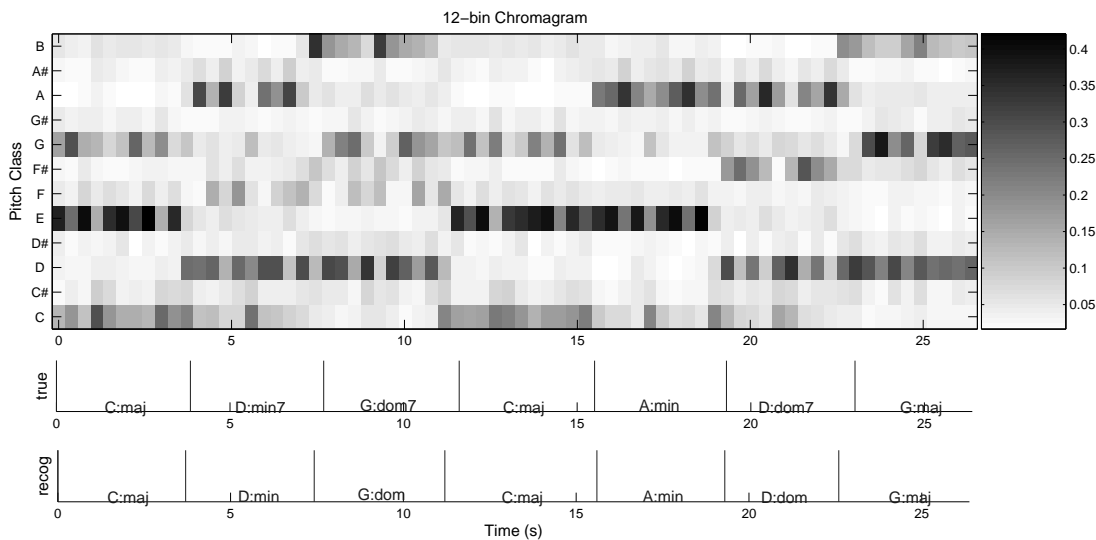


Figure 4. 12-bin chromagram of an excerpt from Bach’s *Prelude in C Major* (BWV 846) performed by Glenn Gould. At the bottom chord labels with boundaries can be observed: “true” corresponds to the ground-truth annotation, and “recog” corresponds to the system output.

more robust to perform harmonic analysis on the symbolic data than on the raw audio data since symbolic files such as MIDI files contain noise-free pitch information. In addition, by using a sample-based synthesizer, we could create audio files which have enharmonically rich spectrum as in real recordings.

As feature vectors, we used 12-bin tuned chroma vectors which have been successfully used by others for the chord recognition application. We have defined 36 classes or chord types in our model, which include for each pitch class three distinct sonorities – major, minor, and diminished. We treated seventh chords as their corresponding root triads, and disregarded augmented chords since they very rarely appear in tonal music.

After the model parameters were estimated from the training data, unseen test input of real recording was fed to the model, and the Viterbi algorithm was applied to find the best probable state path, *i.e.*, chord sequence, at the frame rate. Experiments showed very promising results.

In this paper, we trained our model only on piano music, and tested with piano music. In the near future we plan to include more training data with different instrumentation and genre to make our system more general to all kinds of music.

6. Acknowledgments

The authors would like to thank Craig Sapp and Jonathan Berger for fruitful discussions and suggestions regarding this research.

References

- [1] A. Sheh and D. P. Ellis, “Chord segmentation and recognition using EM-trained hidden Markov models,” in *Proceedings of the International Symposium on Music Information Retrieval*, Baltimore, MD, 2003.
- [2] J. P. Bello and J. Pickens, “A robust mid-level representation for harmonic content in music signals,” in *Proceedings of the International Symposium on Music Information Retrieval*, London, UK, 2005.
- [3] T. Fujishima, “Realtime chord recognition of musical sound: A system using Common Lisp Music,” in *Proceedings of the International Computer Music Conference*. Beijing: International Computer Music Association, 1999.
- [4] E. Gomez and P. Herrera, “Automatic extraction of tonal metadata from polyphonic audio recordings,” in *Proceedings of the Audio Engineering Society*. London: Audio Engineering Society, 2004.
- [5] C. L. Krumhansl, *Cognitive Foundations of Musical Pitch*. Oxford University Press, 1990.
- [6] S. Pauws, “Musical key extraction from audio,” in *Proceedings of the International Symposium on Music Information Retrieval*, Barcelona, Spain, 2004.
- [7] C. A. Harte and M. B. Sandler, “Automatic chord identification using a quantised chromagram,” in *Proceedings of the Audio Engineering Society*. Spain: Audio Engineering Society, 2005.
- [8] J. C. Brown, “Calculation of a constant-Q spectral transform,” *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1990.
- [9] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [10] D. Temperley, *The cognition of basic musical structures*. The MIT Press, 2001.