A System for Automatic Chord Transcription from Audio Using Genre-Specific Hidden Markov Models

Kyogu Lee

Center for Computer Research in Music and Acoustics
Stanford University, Stanford CA 94305, USA
kglee@ccrma.stanford.edu

Abstract. We describe a system for automatic chord transcription from the raw audio using genre-specific hidden Markov models trained on audio-from-symbolic data. In order to avoid enormous amount of human labor required to manually annotate the chord labels for ground-truth, we use symbolic data such as MIDI files to automate the labeling process. In parallel, we synthesize the same symbolic files to provide the models with the sufficient amount of observation feature vectors along with the automatically generated annotations for training. In doing so, we build different models for various musical genres, whose model parameters reveal characteristics specific to their corresponding genre. The experimental results show that the HMMs trained on synthesized data perform very well on real acoustic recordings. It is also shown that when the correct genre is chosen, simpler, genre-specific model yields performance better than or comparable to that of more complex model that is genre-independent. Furthermore, we also demonstrate the potential application of the proposed model to the genre classification task.

1 Introduction

Extracting high-level information of musical attributes such as melody, harmony, key or rhythm from the raw audio is very important in music information retrieval (MIR) systems. Using such high-level musical information, users can efficiently and effectively search, retrieve and navigate through a large collection of musical audio. Among those musical attributes, chords play a key role in Western tonal music. A musical chord is a set of simultaneous tones, and succession of chords over time, or chord progression, forms 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 at every beat or measure.

Recognizing the chords automatically from audio is of great use for those who want to do harmony 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 themes, phrases or forms can be defined.

Chord sequences with the timing of chord boundaries are also a very compact and robust mid-level representation of musical signals, and have many potential

N. Boujemaa, M. Detyniecki, and A. Nürnberger (Eds.): AMR 2007, LNCS 4918, pp. 134–146, 2008.

applications, which include music identification, music segmentation, music similarity finding, mood classification and audio summarization. Chord sequences have been successfully used as a front end to the audio cover song identification system in [1], where a dynamic time warping algorithm was used to compute the minimum alignment cost between two frame-level chord sequences. For these reasons and others, automatic chord recognition has recently attracted a number of researchers in the music information retrieval community.

Hidden Markov models (HMMs) are very successful for speech recognition, and they owe such high performance largely due to gigantic databases accumulated over decades. Such a huge database not only helps estimate the model parameters appropriately, but also enables researchers to build richer models, resulting in better performance. However, there is very few such database available for music applications. Furthermore, the acoustical variance in a piece of music is far greater than that in speech in terms of its frequency range, timbre due to instrumentation, dynamics, and/or tempo, and thus a even more data is needed to build the generalized models.

It is very difficult to obtain a large set of training data for music, however. First of all, it is nearly impossible for researchers to acquire a large collection of musical recordings. Secondly, hand-labeling the chord boundaries in a number of recordings is not only an extremely time consuming and laborious task but also involves performing harmony analysis by someone with a certain level of expertise in music theory or musicology.

In this paper, we propose a method of automating the daunting task of providing the machine learning models with a huge amount of labeled training data for supervised learning. To this end, we use symbolic music documents such as MIDI files to generate chord names and precise chord boundaries, as well as to create audio files. Audio and chord-boundary information generated this way are in perfect alignment, and we can use them to estimate the model parameters. In addition, we build a separate model for each musical genre, which, when a correct genre model is selected, turns out to outperform a generic, genre-independent model. The overall system is illustrated in Figure 1.

There are several advantages to this approach. First, a great number of symbolic music files are freely available, often the times categorized by genres. Second, we do not need to manually annotate chord boundaries with chord names to obtain training data. Third, we can generate as much data as needed with the same symbolic files but with different musical attributes by changing instrumentation, tempo, or dynamics when synthesizing audio. This helps avoid overfitting the models to a specific type of music. Fourth, sufficient training data enables us to build richer models for better performance.

This paper continues with a review of related work in Section 2; in Section 3, we describe the feature vector we used to represent the state in the models; in Section 4, we explain the method of obtaining the labeled training data, and describe the procedure of building our models; in Section 5, we present experimental results with discussions, and draw conclusions followed by directions for future work in Section 6.

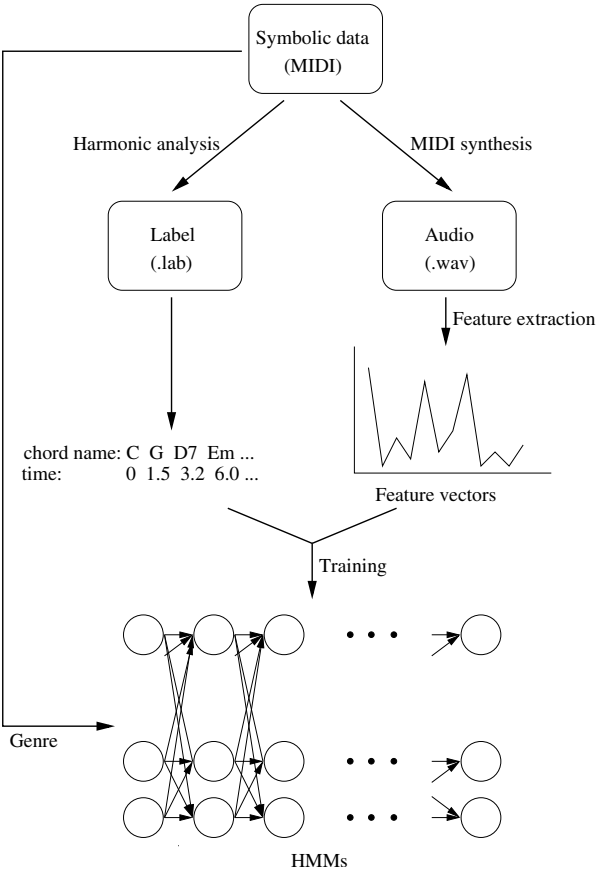

Fig. 1. Overview of the system

2 Related Work

Several systems have been proposed for chord recognition from the raw waveform. Some systems use a simple pattern matching algorithm [2,3,4] while others use more sophisticated machine learning techniques such as hidden Markov models or Support Vector Machines [5,6,7,8,9]. Our approach is closest to two previous works.

Sheh and Ellis proposed a statistical learning method for chord segmentation and recognition [5]. 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 algorithm to estimate the model parameters. The frame accuracy they obtained was about 76% for segmentation and about 22% for recognition, respectively. The poor performance for recognition may be due to insufficient training data

compared with a large set of classes (just 20 songs to train the model with 147 chord types). It is also possible that the flat-start initialization in the EM algorithm yields incorrect chord boundaries resulting in poor parameter estimates.

Bello and Pickens also used HMMs with the EM algorithm to find the crude transition probability matrix for each input [6]. What was novel in their approach was that they incorporated musical knowledge into the models by defining a state transition matrix based on the key distance in a circle of fifths, and avoided random initialization of a mean vector and a covariance matrix of observation distribution. In addition, in training the model's parameter, they selectively updated the parameters of interest on the assumption that a chord template or distribution is almost universal regardless of the type of music, 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. Even with the high recognition rate, it still remains a question if it will work well for all kinds of music.

The present paper expands our previous work on chord recognition [8,9,10]. It is founded on the work of Sheh and Ellis or Bello and Pickens in that the states in the HMM represent chord types, and we try to find the optimal path, *i.e.*, the most probable chord sequence in a maximum-likelihood sense using a *Viterbi* decoder. The most prominent difference in our approach is, however, that we use a *supervised learning* method; *i.e.*, we provide the models with feature vectors as well as corresponding chord names with precise boundaries, and therefore model parameters can be directly estimated without using an EM algorithm when a single Gaussian is used to model the observation distribution for each chord. In addition, we propose a method to automatically obtain a large set of labeled training data, removing the problematic and time consuming task of manual annotation of precise chord boundaries with chord names. Furthermore, this large data set allows us to build genre-specific HMMs, which not only increase the chord recognition accuracy but also provide genre information.

3 System

Our chord transcription system starts off by performing harmony analysis on symbolic data to obtain label files with chord names and precise time boundaries. In parallel, we synthesize the audio files with the same symbolic files using a sample-based synthesizer. We then extract appropriate feature vectors from audio which are in perfect sync with the labels and use them to train our models.

3.1 Obtaining Labeled Training Data

In order to train a supervised model, we need a large number of audio files with corresponding label files which must contain chord names and boundaries. To automate this laborious process, we use symbolic data to generate label files as well as to create time-aligned audio files. 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 harmony 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) is extracted. Sequence of chords are used as pseudo ground-truth or labels when training the HMMs along with proper feature vectors.

We used symbolic files in MIDI (Musical Instrument Digital Interface) format. For harmony analysis, we used the Melisma Music Analyzer developed by Sleator and Temperley [11]. 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, we can generate label files with sequence of chord names and accurate boundaries.

The symbolic harmony-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 yielded an accuracy of 83.7% and 87.4%, respectively [12].

We then synthesize the audio files using Timidity++. Timidity++ is a free software synthesizer, and converts MIDI files into audio files in a WAVE format.[1] It uses a sample-based synthesis technique to create harmonically rich audio as in real recordings. The raw audio is downsampled to 11025 Hz, and 6-dimensional tonal centroid features are extracted from it with the frame size of 8192 samples and the hop size of 2048 samples, corresponding to 743 ms and 186 ms, respectively.

3.2 Feature Vector

Harte and Sandler proposed a 6-dimensional feature vector called *Tonal Centroid*, and used it to detect harmonic changes in musical audio [13]. It is based on the Harmonic Network or *Tonnetz*, which is a planar representation of pitch relations where pitch classes having close harmonic relations such as fifths, major/minor thirds have smaller Euclidean distances on the plane.

The Harmonic Network is a theoretically infinite plane, but is wrapped to create a 3-D Hypertorus assuming enharmonic and octave equivalence, and therefore there are just 12 chromatic pitch classes. If we reference C as a pitch class 0, then we have 12 distinct points on the circle of fifths from 0-7-2-9-$\cdots$-10-5, and it wraps back to 0 or C. If we travel on the circle of minor thirds, however, we come back to a referential point only after three steps as in 0-3-6-9-0. The circle of major thirds is defined in a similar way. This is visualized in Figure 2. As shown in Figure 2, the six dimensions are viewed as three coordinate pairs $(x1, y1)$, $(x2, y2)$, and $(x3, y3)$.

Using the aforementioned representation, a collection of pitches like chords is described as a single point in the 6-D space. Harte and Sandler obtained a 6-D tonal centroid vector by projecting a 12-bin tuned chroma vector onto the three circles in the equal tempered Tonnetz described above. By calculating the

[1] http://timidity.sourceforge.net/

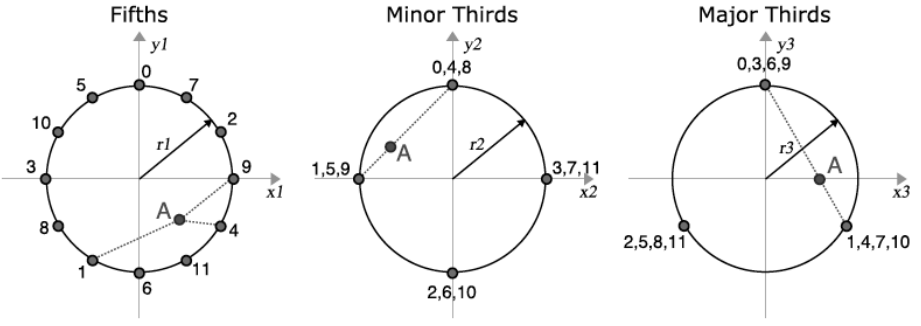

Fig. 2. Visualizing the 6-D Tonal Space as three circles: fifths, minor thirds, and major thirds from left to right. Numbers on the circles correspond to pitch classes and represent nearest neighbors in each circle. Tonal Centroid for A major triad (pitch class 9,1, and 4) is shown at point A (adapted from Harte and Sandler [13]).

Euclidean distance between successive analysis frames of tonal centroid vectors, they successfully detect harmonic changes such as chord boundaries from musical audio.

While a 12-dimensional chroma vector has been widely used in most chord recognition systems, it was shown that the tonal centroid feature yielded far less errors in [10]. The hypothesis was that the tonal centroid vector is more efficient and more robust because it has only 6 dimensions, and it puts emphasis on the interval relations such as fifths, major/minor thirds, which are key intervals that comprise most of musical chords in Western tonal music.

3.3 Hidden Markov Model

A hidden Markov model [14] 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 36-state HMMs. Each state represents a single chord, and the observation distribution is modeled by Gaussian mixtures with diagonal covariance matrices. 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.

In our model, we have defined three chord types for each of 12 chromatic pitch classes according to their sonorities – major, minor, and diminished chords – and thus we have 36 classes in total. We grouped triads and seventh chords with the same root into the same category. For instance, we treated E minor triad and E minor seventh chord as just E minor chord without differentiating the triad and the seventh. We found this class size appropriate in a sense that it lies between overfitting and oversimplification.

With the labeled training data obtained from the symbolic files, we first train our models to estimate the model parameters. Once the model parameters are learned, we then extract the feature vectors from the real recordings, and apply the Viterbi algorithm to the models to find the optimal path, *i.e.*, chord sequence, in a maximum likelihood sense.

3.4 Genre-Specific HMMs

In [10], when tested with various kinds of input, Lee and Slaney showed that the performance was greatest when the input audio was of the same kind as the training data set, suggesting the need to build genre-specific models. This is because not only different instrumentation causes the feature vector to vary, but also the chord progression, and thus the transition probabilities are very different from genre to genre.

We therefore built an HMM for each genre. While the genre information is not contained in the symbolic data, most MIDI files are categorized by their genres, and we could use them to obtain different training data sets by genres. We defined six musical genres including keyboard, chamber, orchestral, rock, jazz, and blues. We acquired the MIDI files for classical music – keyboard, chamber, and orchestral – from `http://www.classicalarchives.com`, and others from a few websites including `http://www.mididb.com`, `http://www.thejazzpage.de`, and `http://www.davebluesybrown.com`. The total number of MIDI files and synthesized audio files used for training is 4,212, which correspond to 348.73 hours of audio and 6,758,416 feature vector frames. Table 1 shows the training data sets used to train each genre model in more detail.

Figure 3 shows the 36×36 transition probability matrices for rock, jazz, and blues model after training. Although they are all strongly diagonal because the rate at which chord changes is usually longer than the frame rate, we still can observe the differences among them. For example, the blues model shows higher transition probabilities between the tonic (I) and the dominant (V) or subdominant (IV) chord than the other two models, which are the three chords almost exclusively used in blues music. This is indicated by darker off-diagonal lines 5 or 7 semitones apart from the main diagonal line. In addition, compared with the rock or blues model, we find that the jazz model reveals more

Table 1. Training data sets for each genre model

Genre	# of MIDI/Audio files	# of frames	Audio length (hours)
Keyboard	393	1,517,064	78.28
Chamber	702	1,224,209	63.17
Orchestral	319	1,528,796	78.89
Rock	1,046	1,070,752	55.25
Jazz	1,037	846,006	43.65
Blues	715	571,589	29.49
All	4,212	6,758,416	348.73

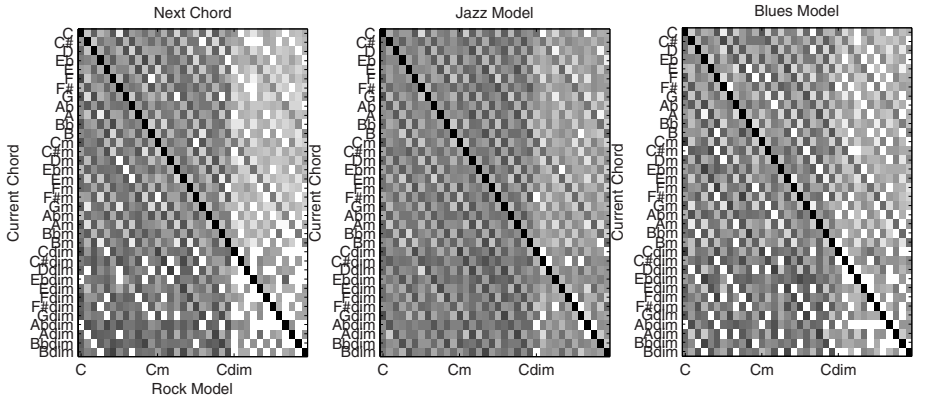

Fig. 3. 36×36 transition probability matrices of rock (left), jazz (center), and blues (right) model. For viewing purpose, logarithm was taken of the original matrices. Axes are labeled in the order of major, minor, and diminished chords.

frequent transitions to the diminished chords, as indicated by darker last third region, which are rarely found in rock or blues music in general.

We can also witness the difference in the observation distribution of the chord for each genre, as shown in Figure 4. Figure 4 displays the mean tonal centroid vectors and covariances of C major chord in the keyboard, chamber, and in the orchestral model, respectively, where the observation distribution of the chord was modeled by a single Gaussian.

We believe these unique properties in model parameters specific to each genre will help increase the chord recognition accuracy when the correct genre model is selected.

4 Experimental Results and Analysis

4.1 Evaluation

We tested our models' performance on the two whole albums of Beatles (CD1: *Please Please Me*, CD2: *Beatles For Sale*) as done by Bello and Pickens [6], each of which contains 14 tracks. Ground-truth annotations were provided by Harte and Sandler at the Digital Music Center at University of London in Queen Mary.[2]

In computing scores, we only counted exact matches as correct recognition. We tolerated the errors at the chord boundaries by having a time margin of one frame, which corresponds approximately to 0.19 second. This assumption is fair since the segment boundaries were generated by human by listening to audio, which cannot be razor sharp.

[2] http://www.elec.qmul.ac.uk/digitalmusic/

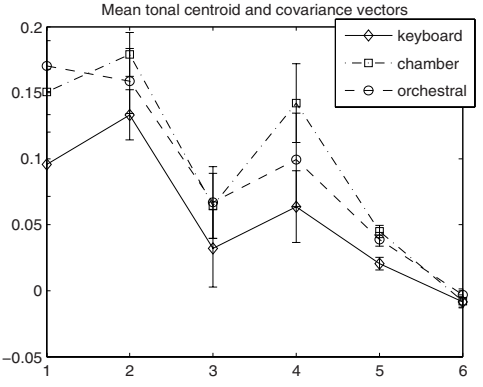

Fig. 4. Mean tonal centroid vectors and covariances of C major chord in keyboard, chamber, and orchestral model

To examine the dependency of the test input on genres, we first compared the each genre model's performance on the same input material. In addition to 6 genre models described in Section 3.4, we built a universal model without genre dependency where all the data were used for training. This universal, genre-independent model was to investigate the model's performance when no prior genre information of the test input is given.

4.2 Results and Discussion

Table 2 shows the frame-rate accuracy in percentage for each genre model. The number of Gaussian mixtures was one for all models. The best results are shown in boldface.

From the results shown in Table 2, we can notice a few things worthy of further discussions. First of all, the performance of the classical models – keyboard, chamber, and orchestral model – is much worse than that of other models. Second, the performance of the rock model came 2nd out of all 7 models, which proves our hypothesis that the model of the same kind as the test input outperforms the others. Third, even though the test material is generally classified

Table 2. Test results for each model with major/minor/diminished chords (36 states, % accuracy)

Model	Beatles CD1	Beatles CD2	Total
Keyboard	38.674	73.106	55.890
Chamber	30.557	73.382	51.970
Orchestral	18.193	57.109	37.651
Rock	45.937	77.294	61.616
Jazz	43.523	76.220	59.872
Blues	**48.483**	**79.598**	**64.041**
All	24.837	68.804	46.821

Table 3. Test results for each model with major/minor chords only (24 states, % accuracy)

Model	Beatles CD1	Beatles CD2	Total
Keyboard	43.945	73.414	58.680
Chamber	43.094	79.593	61.344
Orchestral	37.238	77.133	57.186
Rock	**60.041**	**84.289**	**72.165**
Jazz	44.324	76.107	60.216
Blues	52.244	80.042	66.143
All	51.443	80.013	65.728

as rock music, it is not striking that the blues model gave the best performance considering that rock music has its root in blues music. Particularly, early rock music like Beatles' was greatly affected by blues music. This again supports our hypothesis.

Knowing that the test material does not contain any diminished chord, we did another experiment with the class size reduced down to just 24 major/minor chords instead of full 36 chord types. The results are shown in Table 3.

With fewer chord types, we can observe that the recognition accuracy increased by as much as 20% for some model. Furthermore, the rock model outperformed all other models, again verifying our hypothesis on genre-dependency. This in turn suggests that if the type of the input audio is given, we can adjust the class size of the corresponding model to increase the accuracy. For example, we may use 36-state HMMs for classical or jazz music where diminished chords are frequently used, but use only 24 major/minor chord classes in case of rock or blues music, which rarely uses diminished chords.

Finally, we investigated the universal, genre-independent model in further detail to see the effect of the model complexity. This is because in practical situations, the genre information of the input is unknown, and thus there is no choice but to use a universal model. Although the results shown in Table 2 and Table 3 indicate a general, genre-independent model performs worse than a genre-specific model of the same kind as the input, we can build a richer model for potential increase in performance since we have much more data. Figure 5 illustrates the performance of a universal model as the number of Gaussian mixture increases.

As shown in Figure 5, the performance increases as the model gets more complex and richer. To compare the performance of a complex, genre-independent 36-state HMM with that of a simple, genre-specific 24-state HMM, overlaid is the performance of a 24-state rock model with only one mixture. Although increasing the number of mixtures also increases the recognition rate, it fails to reach the rate of a rock model with just one mixture. This comparison is not fair in that a rock model has only 24 states compared with 36 states in a universal model, resulting in less errors particularly because not a single diminished

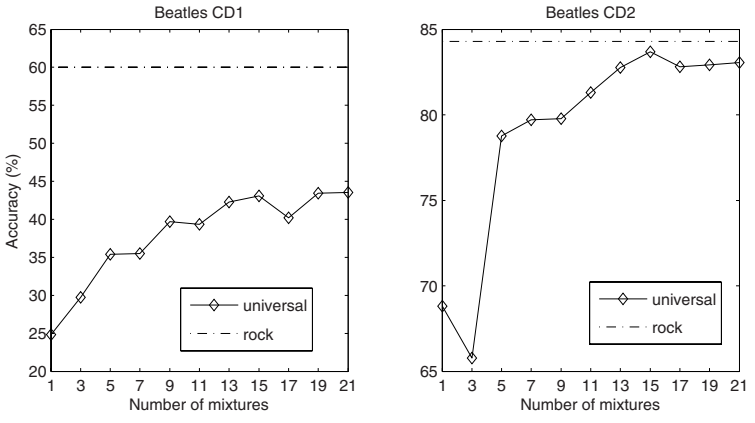

Fig. 5. Chord recognition performance of a 36-state universal model with the number of mixtures as a variable (solid) overlaid with a 24-state rock model with one mixture (dash-dot)

chord is included in the test material. As stated above, however, given no prior information regarding the kind of input audio, we can't take the risk of using a 24-state HMM with only major/minor chords because the input may be classical or jazz music in which diminished chords appear quite often.

The above statements therefore suggest that genre identification on the input audio must be preceded in order to be able to use genre-specific HMMs for better performance. It turns out, however, that we don't need any other sophisticated genre classification algorithms or different feature vectors like MFCC, which is almost exclusively used for genre classification. Given the observation sequence from the input, when there are several competing models, we can select the correct model by choosing one with the maximum likelihood using a forward-backward algorithm also known as a Baum-Welch algorithm. It is exactly the same algorithm as one used in isolated word recognition systems described in [14]. Once the model is selected, we can apply the Viterbi decoder to find the most probable state path, which is identical to the most probable chord sequence. Using this method, our system successfully identified 24 tracks as rock music out of total 28 tracks, which is 85.71% accuracy. What is noticeable and interesting is that the other four songs are all misclassified as blues music in which rock music is known to have its root. In fact, they all are very blues-like music, and some are even categorized as "bluesy".

Our results compare favorably with other state-of-the-art system by Bello and Pickens [6]. Their performance with Beatles' test data was 68.55% and 81.54% for CD1 and CD2, respectively. However, they went through a pre-processing stage of beat detection to perform a tactus-based analysis/recognition. Without a beat-synchronous analysis, their accuracy drops down to 58.96% and 74.78% for each CD, which is lower than our results with a rock model which are 60.04% and 84.29%.

5 Conclusion

In this paper, we describe a system for automatic chord transcription from the raw audio. The main contribution of this work is the demonstration that automatic generation of a very large amount of labeled training data for machine learning models leads to superior results in our musical task by enabling richer models like genre-specific HMMs. By using the chord labels with explicit segmentation information, we directly estimated the model parameters in HMMs.

In order to accomplish this goal, we have used symbolic data to generate label files as well as to synthesize audio files. The rationale behind this idea was that it is far easier and more robust to perform harmony analysis on the symbolic data than on the raw audio data since symbolic files such as MIDI files contain noise-free pitch and time information for every note. In addition, by using a sample-based synthesizer, we could create audio files which have harmonically rich spectra as in real acoustic recordings. This labor-free procedure to obtain a large amount of labeled training data enabled us to build richer models like genre-specific HMMs, resulting in improved performance with much simpler models than a more complex, genre-independent model.

As feature vectors, we used 6-dimensional tonal centroid vectors which proved to outperform conventional chroma vectors for the chord recognition task in previous work by the same author.

Each state in HMMs was modeled by a multivariate, single Gaussian or Gaussian mixtures completely represented by mean vectors and covariance matrices. We have defined 36 classes or chord types in our models, 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 Western tonal music. We reduced the class size down to 24 without diminished chords for some models – for instance, for rock or blues model – where diminished chords are very rarely used, and we could observe great improvement in performance.

Experimental results show that the performance is best when the model and the input are of the same kind, which supports our hypothesis on the need for building genre-specific models. This in turn indicates that although the models are trained on synthesized data, they succeed to capture genre-specific musical characteristics seen in real acoustic recordings. Another great advantage of present approach is that we can also predict the genre of the input audio by computing the likelihoods of different genre models as done in isolated word recognizers. This way, we not only extract chord sequence but also identify musical genre at the same time, without using any other algorithms or feature vectors.

Even though the experiments on genre identification yielded high accuracy, the test data contained only one type of musical genre. In the near future, we plan to expand the test data to include several different genres to fully examine the viability of genre-specific HMMs. In addition, we consider higher-order HMMs for future work because chord progressions based on Western tonal music theory show such higher-order characteristics. Therefore, knowing two or more preceding chords will help to make a correct decision.

Acknowledgment

The author would like to thank Moonseok Kim and Jungsuk Lee at McGill University for fruitful discussions and suggestions regarding this research.

References

1. Lee, K.: Identifying cover songs from audio using harmonic representation. In: Extended abstract submitted to Music Information Retrieval eXchange task, BC, Canada (2006)
2. Fujishima, T.: 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)
3. Harte, C.A., Sandler, M.B.: Automatic chord identification using a quantised chromagram. In: Proceedings of the Audio Engineering Society, Spain. Audio Engineering Society (2005)
4. Lee, K.: Automatic chord recognition using enhanced pitch class profile. In: Proceedings of the International Computer Music Conference, New Orleans, USA (2006)
5. Sheh, A., Ellis, D.P.: Chord segmentation and recognition using EM-trained hidden Markov models. In: Proceedings of the International Symposium on Music Information Retrieval, Baltimore, MD (2003)
6. Bello, J.P., Pickens, J.: A robust mid-level representation for harmonic content in music signals. In: Proceedings of the International Symposium on Music Information Retrieval, London, UK (2005)
7. Morman, J., Rabiner, L.: A system for the automatic segmentation and classification of chord sequences. In: Proceedings of Audio and Music Computing for Multimedia Workshop, Santa Barbar, CA (2006)
8. Lee, K., Slaney, M.: Automatic chord recognition using an HMM with supervised learning. In: Proceedings of the International Symposium on Music Information Retrieval, Victoria, Canada (2006)
9. Lee, K., Slaney, M.: Automatic chord recognition from audio using a supervised HMM trained with audio-from-symbolic data. In: Proceedings of Audio and Music Computing for Multimedia Workshop, Santa Barbara, CA (2006)
10. Lee, K., Slaney, M.: Acoustic Chord Transcription and Key Extraction From Audio Using Key-Dependent HMMs Trained on Synthesized Audio. IEEE Transactions on Audio, Speech and Language Processing 16(2), 291–301 (2008)
11. Sleator, D., Temperley, D.: The Melisma Music Analyzer (2001), `http://www.link.cs.cmu.edu/music-analysis/`
12. Temperley, D.: The cognition of basic musical structures. The MIT Press, Cambridge (2001)
13. Harte, C.A., Sandler, M.B.: Detecting harmonic change in musical audio. In: Proceedings of Audio and Music Computing for Multimedia Workshop, Santa Barbara, CA (2006)
14. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)