Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **2**, March **2018**

Sparse Representation Classification-Based Automatic Chord Recognition For Noisy Music

Zhongyang Rao[1,2], Chunyuan Feng[1]

[1]School of Information Science and Electrical Engineering,
Shandong Jiaotong University, Ji'nan
5001, Haitang Road, Changqing District 250357, Ji'nan, China
yaozhongyang@sohu.com; fengchunyuan818@163.com

[2]School of Electronic Information Engineering, Tianjin University, Tianjin
92, Weijin Road, Nankai District 30072, Tianjin, China
yaozhongyang@eyou.com

Received October, 2015; revised May, 2017

Abstract. *In this paper, Sparse Representation-based Classification(SRC) is used for automatic chord recognition in music signals. It extracts Pitch Class Profile (PCP) features from raw audio and achieve sparse representation of classes via ℓ_1-norm minimization on feature space and uses Viterbi algorithm to recognize 24 major and minor triads. But in the real word, the music usually is corrupted by noise. This recognition model is evaluated on MIREX09 dataset. And it compares the recognition rates when the music contains Gaussian white noise or not. Experimental results demonstrate that the method is robust to the Gaussian white noise.*
Keywords: Chord recognition, Noisy Music, PCP, Sparse Representation-based Classification, Viterbi algorithm

---

1. Introduction.

In music, a chord is a set of three or more notes that is played simultaneously. Chords are mid-level musical features which concisely describe the harmonic content of a piece. Automation labeling of chord is called chord recognition, which finds many applications such as music segmentation, cover song identification, audio matching, music similarity identification, and audio thumb nailing[1]. So automatic chord recognition is very important in musical information retrieval (MIR) in recent years.

In chord recognition, the features used may may not be identical. But in most cases, one of the most commonly used features is variants of the Pitch Class Profile (PCP) introduced by Fujishima (1999)[2]. PCP is also called chroma vector, which is often a 12-dimensional vector. It can convert pitch features into chroma features by adding up all values that belong to the same pitch class. The calculation of an audio file into a chroma representation is based either on the short-time Fourier transform (STFT) in combination with binning strategies [3-6] or on the constant Q transform (CQT) [7-11]. The musical content of audio musical signals can be well described with the chromagram.

The chord recognition is the chord labeling of each chord. Our chord recognition system is based on the sparse representation-based classification (SRC) [12] which has been proposed with amazing identification capability in recent years. Based on 12-dimensional PCP features, SRC discriminately selects the subset that most compactly expresses the input signal and rejects all other possible but less compact representations. Besides of these, we use the method to recognize the chords of noisy music, and compare the recognition rates of ideal music and noisy music.

The rest of this paper is organized as follows: Section 2 reviews previous the related work of this area; Section 3 gives a description of our construction of the feature vector; Section 4 describes the recognition method; Section 5 gives the results on MIREX09 datasets and a comparison between the recognition rates of ideal music and noisy music; Finally we will draw some conclusion and give possible developments for further work.

2. **Related Work.** In audio chord estimation, it mainly includes the feature extraction, modelling techniques, evaluation strategies and so on. Some features are used, such as non-negative least squares (NNLS)[13], chroma DCT-reduced log pitch(CRP)[14], loudness based chromagram (LBC)[15], Mel PCP (MPCP)[16]. But the most popular feature is a chromagram, also known as chroma vectors or Pitch Class Profile (PCP). Fujishima developed a real-time 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[2]. Lee also used binary chord templates[17]. He introduced a new feature called Enhanced Pitch Class Profile (EPCP) using the harmonic product spectrum. Gómez and Herrera [18] used Harmonic Pitch Class Profile (HPCP) as the feature vector.

In modelling techniques, it usually uses the templates-fitting methods [9, 19-23]. Besides templates-fitting methods, it is widely used machine-learning methods such as hidden Markov Model (HMM) [4, 24-30] and DBNs(Dynamic Bayesian Networks)[15, 31] for this recognition process. Sheh and Ellis proposed a statistical learning method for chord segmentation and recognition[24]. Bello and Pickens also used the HMMs with the EM algorithm, but they considered the inherent musicality of audio into the models for model initialization[26].

PCP feature vectors are very important in our recognition system. In the next section, we will describe the main steps for the calculation of log PCP.

3. **Feature Vectors.** First of all, the recognition system extracts a sequence of suitable feature vectors from the audio signal. In our system, the features are log PCP vectors. Mller and Ewert propose feature vectors 12-dimensional Quantized PCP[32, 33] which avoids a possible frequency resolution and is sufficient to separate musical notes of low frequency comparing with others. The calculation of feature vectors PCP can be divided into the following steps: (1) Calculating the 36-bin chromagram with the constant Q transform; (2)Mapping spectral chromagram to a particular semitone; (3)Segmenting the audio signal with beat tracking algorithm; (4)Reducing the 36-bin chromagram to 12-bin chromagram based on beat-synchronous segmentation; (5) Chromagram normalization. Refer to [26] for more detailed steps on how to calculate chromagram.

(1)36-bin chromagram calculation. Using the constant Q transform, it can get $X_{cqt}(k)$ of a audio signal $x(m)$:

$$X_{cqt}(k) = \frac{1}{N_k} \sum_{m=0}^{N_k-1} x(m) \cdot w_{N_k}(m) e^{-j2\pi mQ/N_k} \tag{1}$$

Where k is the bin position, $w_(N_k)(m)$ is the hamming window and its length $N_k = Q \cdot f_k/f_s$. And f_k is the center frequency of the k bin and f_s is the sample frequency. In this paper, the music signal is down-sampled to 11025Hz.

By adding all $X_{cqt}(k)$ that correspond to a particular frequencythen it get 36-bin chromagram of each frames. The specific formula is as follows:

$$QPCP(p) = \sum_{m=0}^{M-1} |X_{cqt}(p + mb)|, p = 1, 2, \cdot, 36 \tag{2}$$

Where M is the total number of octaves and b is the number of bins per octave.

(2)Chromagram tuning. In the 36-bin chromagram, 3 bins represent one note in the octave. Each spectral components of 36-bin is maped to a particular semitone. The mapping formula is as follows:

$$P(k) = 36 * [log_2(f_s/N_k * k/f_0)]mod36 \tag{3}$$

(3)Beat-synchronous segmentation. In our system, it use the beat tracking with dynamic programming method proposed by Daniel P.W. Ellis [34]. This approach has been found to work very well in in many types of music. Segmenting the audio signal with beat tracking algorithm has additional advantage that the chroma feature is a function of beat segments, rather than time.

(4)12-bin chromagram reduction. Finally, averaging the each spectral components of 36-bin in beat segments and summing them in semitones, thus the dimension of chromagram is reduced to 12 from 36. Then the chromagram of audio music can represented with these 12 dimensional vectors.

(5)Chromagram normalization. $QPCP_{12}(p)$ is the 12-bin chromagram. It can get the normalized value with p-norm. The formula is as follows:

$$QPCP_{log}(p) = log_{10}[C * QPCP_{12}(p) + 1] \tag{4}$$

$$QPCP_{norm}(p) = QPCP_{log}(p)/\|QPCP_{log}(p)\| \tag{5}$$

If it performs the logarithm and normalization, the chromagram is called Log PCP. In step (5) it has only normalization, it is called PCP.

As can be seen in Figure 1, the left picture shows a PCP of C major triad. The right one shows its Log PCP, as we can see, the strongest peaks are found at C, E, and G, since C major triad comprises three notes at C (root), E (third), and G (fifth). From the Figure 1, it can see that Log PCP is clear than PCP.

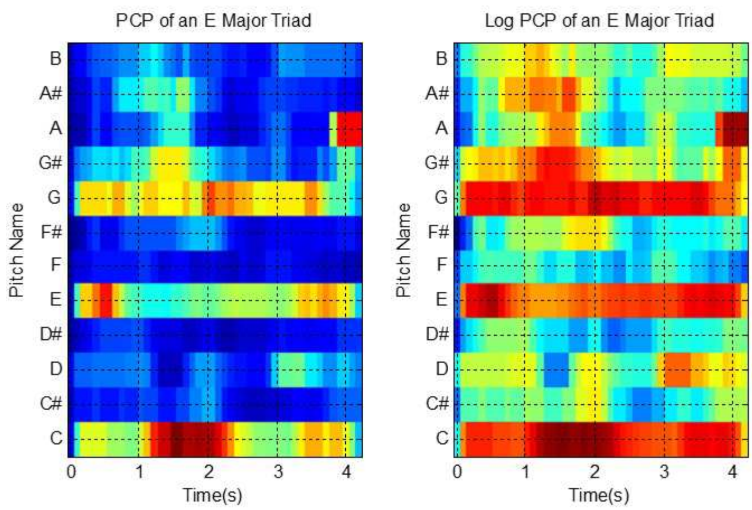

FIGURE 1. PCP and Log PCP of an E major triad

4. **Auto Chord Recognition.** In our chord recognition method, the system includes two sections: (1) Sparse representation-based classification (SRC); (2) Viterbi algorithm. Based on labeled musical fragments, the system uses SRC method and only relies on framewise classification. The method doesnt need amount of training data. If it has amount of training data, the system can add Viterbi algorithm by using transitions between chords to recognize chords.

4.1. **Sparse Representation-based Classification.** Template-based chord recognition methods used the chord definition to extract chord labels from a music piece. In fact, neither training data or extensive music theory knowledge is used[35]. The most HMM methods need amount of training data, parameters are learned from data. If labeled musical fragments are selected in template-based chord recognition, then the template is the PCP matrix of chords. So the basic problem in chord recognition is to use labeled training musical fragments from k distinct object chords to correctly determine the chord to which a new test musical fragments belongs. This problem can solved by sparse representation-based classification (SRC) [12, 36].

In recent years, the sparse representation become an important research focus in the field of pattern recognition, and has attracted wide attention in areas such as machine vision, machine learning, pattern recognition. The earliest in the field of sparse representation have been proposed[37, 38]. Its core idea is that the test sample is the linear representation of labeled training samples which the test sample belongs to. Obviously, only a few of the linear coefficient are zero, that is to say the coefficient vector is sparse.

Our chord recognition system is based on the sparse representation-based classification (SRC) [12]. Labeled samples by this algorithm can directly be used as the classifier training samples, saving lots of time and system resources. The following sections outline the method.

At first, we define a matrix $D = [D_1, D_2, , D_k] = [u_{1,1}, u_{1,2}, \cdots, u_{k,n_k}] \in \mathbb{R}^{m\times n}$ by collecting n classifier training samples of all k classes, where m is the dimension of the feature set. For a given test sample $y \in \mathbb{R}^m$ from subject i, can be rewritten in terms of all training samples as:

$$y = Dx_0 \in \mathbb{R}^m \tag{6}$$

Where x_0 is a coefficient vector, whose entries ideally the coefficient vector $x_0 = [0, \cdots, 0, a_{i,1}, a_{i,2}, \cdots, a_{i,n_i}, 0, \cdots, 0]$ are mostly zero except the values corresponding to the i-th class are non-zero and other coefficient values should be 0.

As coefficient vector x_0 can identify the test sample y, it can be obtained by solving the linear equation (6). Recent development in the emerging compressed sensing theory and sparse representation reveals that if the solution x_0 sought is sparse enough, the solution to the system of equation (6) is equivalent to the following ℓ_1-minimization problem:

$$\widehat{x}_1 = argmin\|x\|_1 \ subject \ to \ y = Dx \tag{7}$$

Since real music are noisy, it may not be possible to express the test sample exactly as a sparse representation of the training samples. Account for small noise, the model(6) can be modified to explicitly, as following

$$y = Dx_0 + E \in \mathbb{R}^m \tag{8}$$

Where E is a noise term with bounded energy $\|E\|_2 < \varepsilon$. The sparse solution x_0 can still be obtained by solving the following ℓ_1-minimization problem:

$$\widehat{x}_1 = argmin\|x\|_1 \ subject \ to \ \|y - Dx\|_2 \leq \varepsilon \tag{9}$$

According to these non-zero coefficient x_1, it can quickly know the test sample belongs to the class. Actually, because of noise and model errors, some of entries with multiple object classes is small nonzero values. For each class i, the given test sample y can be approximated as $\widehat{y}_i = D\delta_i(\widehat{x}_1)$, where $\delta_i : \mathbb{R}^n \to \mathbb{R}^n$ is the characteristic function which selects the coefficients associated with the ith class. We then calculate the residual between y and $\widehat{y}_i$:

$$r_i(y) = \|y - Dx\|_2 \tag{10}$$

At last, we classify y based on these approximations by assigning it to the object class that minimizes the residual, as follow:

$$identity(y) = \underset{i}{argmin}\ r_i(y) \tag{11}$$

The resulting SRC algorithm is summarized below.

Algorithm 1 Recognition via Sparse Representation Classification (SRC)

1: **Input:** D is a matrix of classifier training samples, $D = [D_1, D_2, \cdots, D_k] \in \mathbb{R}^{m*n}$ for k classes, a test sample $y \in \mathbb{R}^m$.
2: **Output:** $identity(y) = \underset{i}{argmin}\ r_i(y)$
3: Solve the ℓ_1-minimization problem: $\widehat{x}_1 = argmin\|x\|_1\ subject\ to\ \|y - Dx\|_2 \leq \varepsilon$
4: Compute the residuals $r_i(y) = \|y - Dx\|_2$, for $i = 1, \cdots, k$

If it selects a sample of **D** chord, using SRC solves its coefficients. Its residual of subset chord and coefficients of sparse linear combination are shown in figure 2. Many of these coefficients are zero. And the minimum residual of subset chord is the correct chord. When the sample contains Gaussian white noise and SNR is 10dB, its residual and coefficients are shown in figure 3. Through the sample contains noise, the SRC can recognize the correct chord. But the coefficient has many nonzero values. The proportion of the maximum residual and minimum value is reduced and the minimum increased.

4.2. **Viterbi Algorithm.** In SRC method, it uses the residuals $r_i(y)$ to recognize the chord. The method recognizes the chord on frame-wise classification. If it uses transitions between chords, it can improve the recognition rates of chord. Our system uses the Viterbi algorithm. Suppose the system has hidden N states, and we denote each state as $S_i, i \in [1 : N]$. The observed events are Q_t, $t \in [1 : T]$. The current observed events $Q = Q, Q_2, \cdots, Q_T, t \in [1 : T]$. A_{ij} represents the probability chord S_i jump to chord S_j. At an arbitrary time point t, for each of the states S_i, a partial probability $\delta_t(S_i)$ is defined to indicate the probability of the most probable path ending at the state S_i, given the current observed events $Q, Q_2, \cdots, Q_t$: $\delta_t(S_i) = \underset{j}{max}(\delta_{t-1}(S_j)A(S_j, S_i)P(Q_t|S_i))$.Here, we assume that we already know the probability $\delta_{t-1}(S_j)$ for any of the previous states S_j at time $t-1$. $P(Q_t|S_i)$ is the current observation probability. After having all the objective probabilities for each state at each time point, the algorithm seeks from the very end backwards to the beginning to find the most probable path of states for the given sequence of observation events $\Psi_t(i) = \underset{1\leq j\leq N}{argmax}[(\delta_(t-1)(S_j)A(S_j, S_i))]$.Where $\Psi_t(i)$ indicates which state is the most optimal state at time t based on the probability computed in the first stage.

The Viterbi algorithm is as follows:

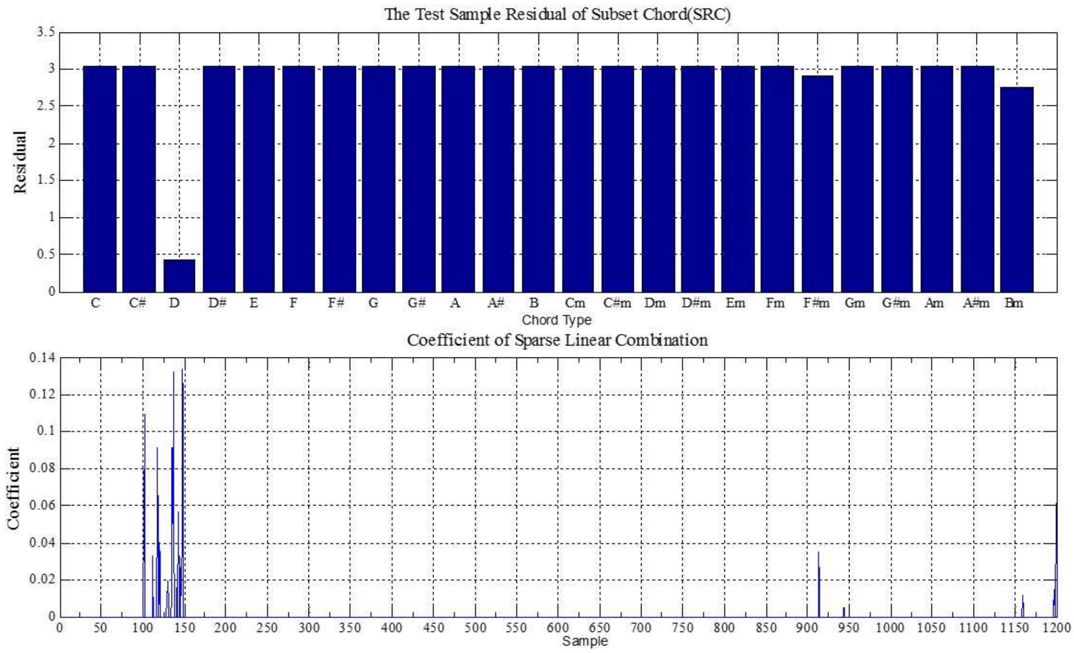

FIGURE 2. The residual and sparse linear coefficient of **D** chord sample

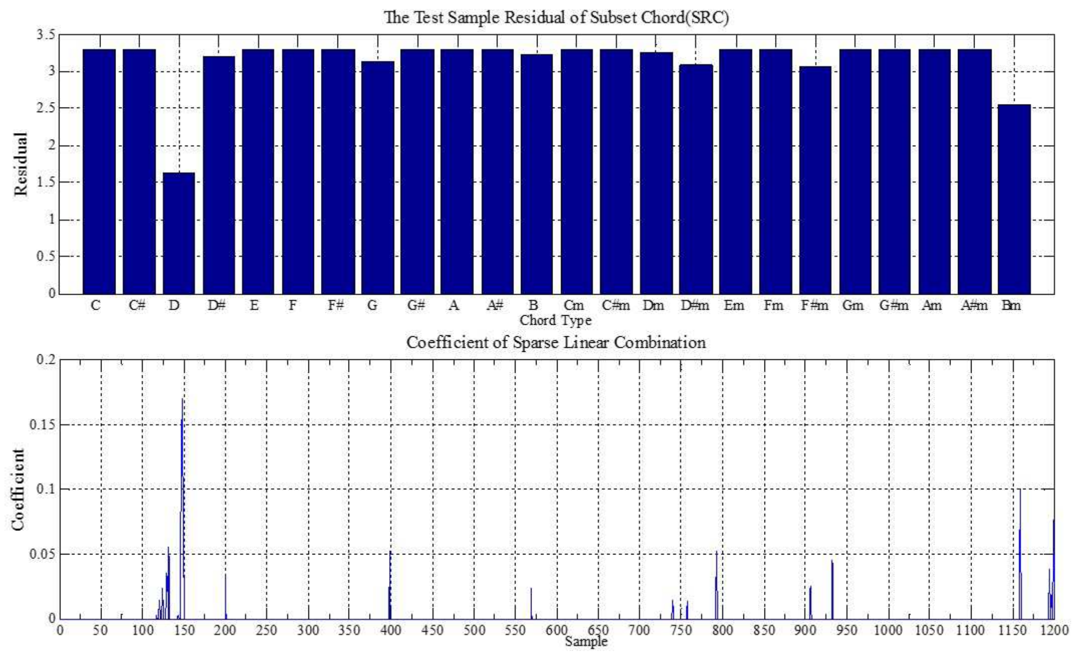

FIGURE 3. The residual and sparse linear coefficient of **D** chord sample when it contains noise

Algorithm 2 Recognition via Sparse Representation Classification (SRC)

Initialization: $\delta_t(S_i) = \prod_i P(Q_1|S_i), \Psi_t(i) = 0, 1 \leq i \leq N.$

Recursion:$\delta_t(S_i) = \underset{j}{max}(\delta_{t-1}(S_j) \cdot A(S_j, S_i) \cdot P(Q_t|S_i)),\ \Psi_t(i) = \underset{1\leq j\leq N}{argmax}[(\delta_{t-1}(S_j) \cdot A(S_j, S_i))]$.

Termination: $q_T^* = \underset{1\leq i\leq N}{max}[\delta_t(S_i)],\ P^* = \underset{i}{max}[\delta_t(S_i)].$

Path Backtracking: $q_t^* = \Psi_{t+1} q_{t+1}^*.$

In our method, the initialization observation probability $\prod_i$ is equal to 1/24. The observed events are y_t, where y_t is the PCP feature of t^{th} frame. And current observation probability is $r_i(y_t)$ and replaces the $P(Q_t|S_i)$ in Viterbi algorithm. S_i represents the chord $i \in [1:24]$, where N is the number of chord and set to 24.

The following figure 4 is the comparison of ground truth chord and estimated chord of the Beatles song **Misery**. In the top figure, it only uses the SRC method to recognize the chord and the bottom uses SRC and Viterbi decoding. The ground truth chord is represented in pink and the estimated chord labels are in blue. From the figure 4 it can see that the estimation is more stable when it uses the Viterbi than without.

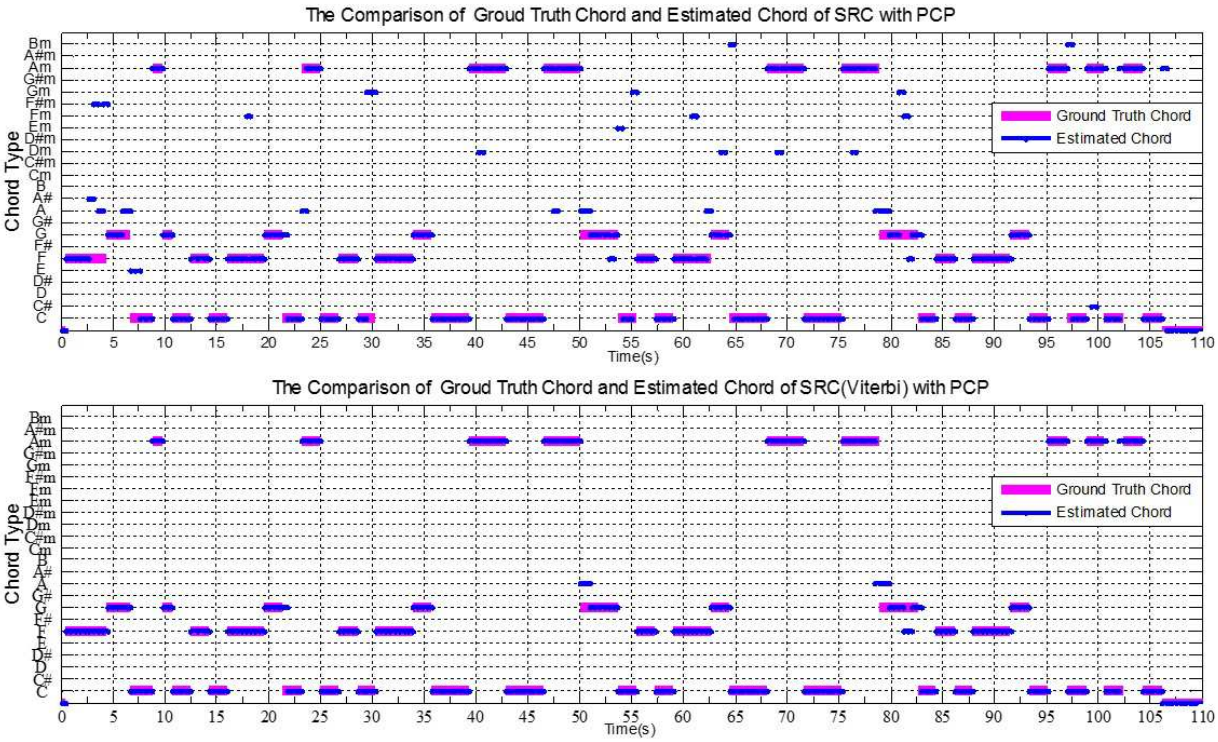

Figure 4. The comparison of ground truth chord and test chord

5. **Evaluation.** For evaluation, we use the MIREX09 dataset in Audio Chord Estimation task of MIREX. The dataset consists of 12 Beatles albums (180 songs, PCM 44 100Hz, 16 bits, mono). Besides the Beatles albums, in 2009, an extra dataset was donated by Matthias Mauch which consists of 38 songs from Queen and Zweieck.

This database based been extensively used for the evaluation of many chord recognition systems, in particular those presented at MIREX 2013, 2014 for the Audio Chord detection task. The evaluation is realized thanks to the chord annotations of the Beatles albums kindly provided by Harte and Sandler[39], and the chord annotations of Queen and Zweieck provided by Matthias Mauch.

The chord dictionary used in this work is the set of 24 major and minor triads, one each for all 12 members of the chromatic scales: C Major, C minor, C# Major, C# minor, $\cdots$, A# Major, A# minor, B Major, B minor. Each triad contains 50 labeled musical fragments which select from the Beatles albums.

To evaluate the quality of an automatic transcription, a transcription is compared to ground truth created by one or more human annotators. Since 2013, MIREX typically uses chord symbol recall (CSR) to estimate how well the predicted chords match the ground truth:

$$CSR = \frac{totaldurationofsegmentswhereannotationequalsestimation}{totaldurationofannotatedsegments} \tag{12}$$

Because pieces of music come in a wide variety of lengths, we will weight the CSR by the length of the song when computing an average for a given corpus. This final number is referred to as the weighted chord symbol recall (WCSR).

In order to verify the robustness of SRC, it first tests the algorithm of SRC adding different signal to noise ratio (SNR) noises. For the convenience of testing, the adding noise is white noise.

From the figure 5 and figure 6, it can see that the recognition rate of SRC with viterbi is higher than without, and SRC with LPCP higher than with PCP. When the noises add to the music, the recognition rates decrease hardly. When the noise is very large, for example SNR is 10dB, the rate decrease 8 percent.

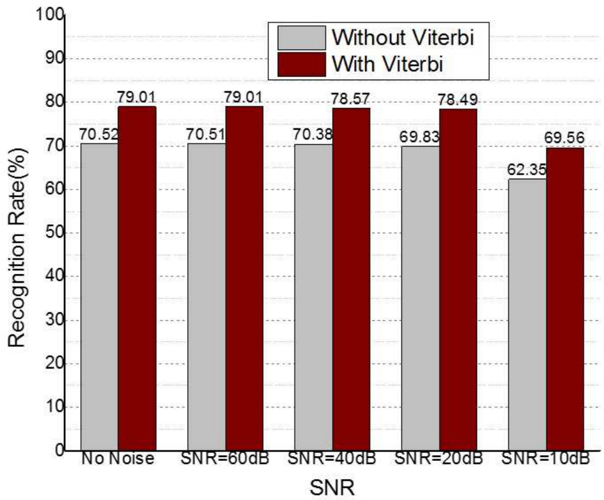

FIGURE 5. Recognition Rate with PCP

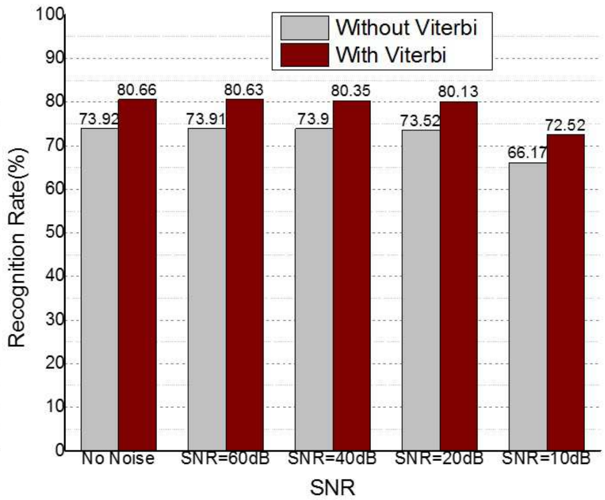

FIGURE 6. Recognition Rate with LPCP

6. **Conclusion.** In this paper, we have presented a new machine learning model-SRC for chord recognition. In comparison with different SNR, the method is robust to Gaussian white noise. When it uses the viterbi algorithm, the recognition rate can increases 9 percent with PCP feature, 6 percent with LPCP feature. The key part of our new method is the training chord samples, which are randomly cut down from the songs of Beatles.

Based on MIR development and combined our research, the following work is proposed. First, this paper only involved chord recognition which is a part of chord transcription task. Future work will consider adding recognition of more complex chords to our work. Chord recognition will find many applications in the field of MIR such as song identification, query by similarity or structure analysis. Second, in this work we take the effect of different features into account in SRC. We could add appropriate other features in the feature.

Acknowledgment. This work was supported by the national Natural Science Foundation of China (Grant no. 61101225 and 61601264) and PhD research startup foundation of Shandong Jiaotong University. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] M. McVicar, R. Santos-Rodrguez, Y. Ni, and T. De Bie, Automatic chord estimation from audio: A review of the state of the art, *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, pp. 556-575, 2014.

[2] T. Fujishima, Realtime chord recognition of musical sound: a system using common lisp music, *Proceedings of the International Computer Music Conference*, Beijing, China, pp. 464-467, 1999.

[3] M. A. Bartsch and G. H. Wakefield, Audio thumbnailing of popular music using chroma-based representations, *Multimedia, IEEE Transactions on*, vol. 7, pp. 96-104, 2005.

[4] A. Sheh and D. P. Ellis, Chord segmentation and recognition using EM-trained hidden Markov models, *ISMIR 2003*, pp. 185-191, 2003.

[5] E. Gómez, Tonal description of polyphonic audio for music content processing, *INFORMS Journal on Computing*, vol. 18, pp. 294-304, 2006.

[6] M. Khadkevich and M. Omologo, Use of Hidden Markov Models and Factored Language Models for Automatic Chord Recognition, *ISMIR*, pp. 561-566, 2009.

[7] J. C. Brown, Calculation of a constant Q spectral transform, *The Journal of the Acoustical Society of America*, vol. 89, pp. 425-434, 1991.

[8] J. P. Bello and J. Pickens, A Robust Mid-Level Representation for Harmonic Content in Music Signals, *ISMIR, 2005*, pp. 304-311, 2005.

[9] C. Harte and M. Sandler, Automatic chord identifcation using a quantised chromagram, *Audio Engineering Society Convention 118*, 2005.

[10] M. Müller and S. Ewert, Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features, *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, 2011.

[11] K. Lee, Automatic chord recognition from audio using enhanced pitch class profile, *Proc. of the International Computer Music Conference*, 2006.

[12] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, Robust face recognition via sparse representation, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, pp. 210-227, 2009.

[13] M. Mauch and S. Dixon, Approximate Note Transcription for the Improved Identification of Difficult Chords, *ISMIR, 2010*, pp. 135-140, 2010.

[14] M. Müller, S. Ewert, and S. Kreuzer, Making chroma features more robust to timbre changes, *Acoustics, Speech and Signal Processing, ICASSP 2009. IEEE International Conference on*, pp. 1877-1880, 2009.

[15] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie, An end-to-end machine learning system for harmonic analysis of music, *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, pp. 1771-1783, 2012.

[16] F. Wang and X. Zhang, Research on CRFs in Music Chord Recognition Algorithm, *Journal of Computers*, vol. 8, p. 1017, 2013.

[17] K. Lee, Automatic chord recognition from audio using enhanced pitch class profile, *International Computer Music Conference (ICMC)*, New Orleans, Louisiana, USA, 2006.

[18] E. Gómez, P. Herrera, and B. Ong, Automatic tonal analysis from music summaries for version identification, *Audio Engineering Society Convention 121*, San Francisco, CA, USA, 2006.

[19] L. Oudre, Y. Grenier, and C. Févotte, Template-based Chord Recognition: Influence of the Chord Types, *ISMIR, 2009*, pp. 153-158, 2009.

[20] T. Fujishima, Realtime chord recognition of musical sound: A system using common lisp music, *Proc. ICMC, 1999*, pp. 464-467, 1999.

[21] T. Rocher, M. Robine, P. Hanna, L. Oudre, Y. Grenier, and C. Fv́otte, Concurrent Estimation of Chords and Keys from Audio, *ISMIR, 2010*, pp. 141-146, 2010.

[22] T. Cho and J. P. Bello, A feature smoothing method for chord recognition using recurrence plots *Music Information Retrieval Evaluation eXchange(MIREX 2011)*, Miami, Florida, USA, 2011.

[23] L. Oudre, C. Févotte, and Y. Grenier, Probabilistic template-based chord recognition, *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 2249-2259, 2011.

[24] A. Sheh and D. P. Ellis, Chord segmentation and recognition using EM-trained hidden Markov models, *ISMIR 2003*, Library of Congress, Washington, D.C., USA, and Johns Hopkins University, Baltimore, Maryland, USA, pp. 185-191, 2003.

[25] H. Papadopoulos and G. Peeters, Large-scale study of chord estimation algorithms based on chroma representation and HMM, *Content-Based Multimedia Indexing, 2007. CBMI'07, International Workshop on*, pp. 53-60, 2007.
[26] J. P. Bello and J. Pickens, A Robust Mid-Level Representation for Harmonic Content in Music Signals,*ISMIR 2005*, London, UK, pp. 304-311, 2005.
[27] K. Lee and M. Slaney, Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio, *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, pp. 291-301, 2008.
[28] H. Papadopoulos and G. Peeters, Simultaneous estimation of chord progression and downbeats from an audio file, *Acoustics, Speech and Signal Processing, ICASSP 2008, IEEE International Conference on*, pp. 121-124, 2008.
[29] R. Scholz, E. Vincent, and F. Bimbot, Robust modeling of musical chord sequences using probabilistic N-grams, *Acoustics, Speech and Signal Processing, ICASSP 2009. IEEE International Conference on*, pp. 53-56, 2009.
[30] K. Yoshii and M. Goto, A Vocabulary-Free Infinity-Gram Model for Nonparametric Bayesian Chord Progression Analysis, *ISMIR, 2011*, pp. 645-650, 2011..
[31] M. Mauch, Automatic chord transcription from audio using computational models of musical context, *School of Electronic Engineering and Computer Science Queen Mary*, University of London, 2010.
[32] M. Müller and S. Ewert, Towards timbre-invariant audio features for harmony-based music, *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, pp. 649-662, 2010.
[33] M. Müller and S. Ewert, Chroma Toolbox: Matlab Implementations for Extracting Variants of Chroma-Based Audio Features, *ISMIR, 2011*, pp. 215-220, 2011.
[34] D. P. Ellis, Beat tracking by dynamic programming, *Journal of New Music Research*, vol. 36, pp. 51-60, 2007.
[35] L. Oudre, Template-based chord recognition from audio signals, *TELECOM ParisTech*, 2010.
[36] K. Huang and S. Aviyente, Sparse representation for signal classification, *Advances in neural information processing systems*, pp. 609-616, 2006.
[37] E. J. Candès, Compressive sampling, *Proceedings oh the International Congress of Mathematicians*, Madrid, Spain, pp. 1433-1452, 2006.
[38] D. L. Donoho, Compressed sensing, *Information Theory, IEEE Transactions on*, vol. 52, pp. 1289-1306, 2006.
[39] C. Harte, M. B. Sandler, S. A. Abdallah, and E. Gómez, Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations, *ISMIR*, London, UK, pp. 66-71, 2005.