EXTENDED ABSTRACT FOR MIREX 2017 SUBMISSONS: CHORD RECOGNITION USING RANDOM FOREST MODEL

Junyan Jiang, Wei Li, Yiming Wu
School of Computer Science and Technology,
Fudan University of Shanghai
{jiangjy14, weili-fudan, yimingwu11}@fudan.edu.cn

ABSTRACT

This extended abstract presents two new algorithms for automatic chord estimation. Both of them follows a Chordino-like approach as a framework, but a new statistical model, the random forest model, was introduced. In the second submission, we suggest a new way to perform joint estimation of beat positions and chord sequences, aiming to replace the traditional beat-aligned chord estimation methods.

1. INTRODUCTION

We submitted two systems. In the first system, we introduce the random forest model to the automatic chord estimation task. Instead of mapping the chord types into smaller vocabulary, we perform training on a rich chord classes directly. After that, the HMM decoder is used for smoothing purpose.

In the second system, we introduce the beat positions to limit chord transitions. To prevent the advert effect when the beat-tracking algorithm outputs wrong annotations, we adopted a new CRF model to estimate beat positions and chord sequences jointly. Beats and chords are able to correct each other in the extracting process.

2. SUBMISSONS

2.1 RF-HMM Estimator

A random forest model is a collection of decision trees. A single decision tree functions similar to the one in figure 2. In a random forest, each decision tree is probabilistic, trained separately by a randomized subset of training samples and features. A single decision tree may not be accurate, so multiple trees are used to form a random forest for better estimation results.

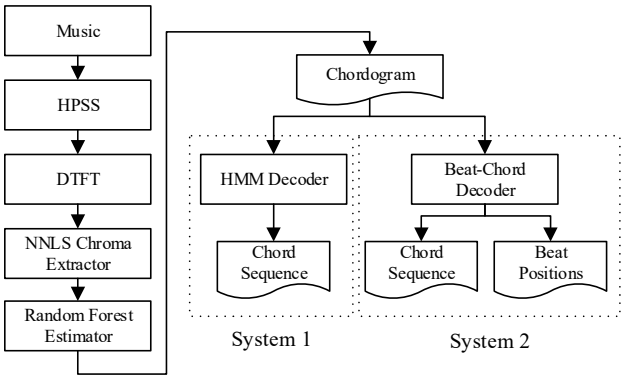

Figure 1. Overview of the submitted systems

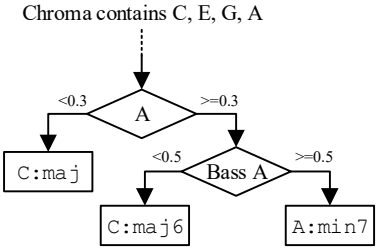

Figure 2. An (oversimplified) example of a decision tree

2.1.1 Preprocess And Feature Extraction

To process an audio file, the HPSS (Harmonic Percussive Source Separation) is performed first. Then we use NNLS-Chroma [1] to get the features. Although the NNLS-Chroma itself has the ability to reduce the influence of percussion components, an HPSS will make it work better in general.

2.1.2 Chord Sequence Decoding

A trained random forest model is used to estimate the emission strength of each chord label in each frame, forming a chordogram $\mathbf{H}$, where $H_n[c]$ represents for the emission possibility for chord c at n-th frame.

Then an HMM is used for decoding chord sequence. It's for smoothing propose only, similar to the one used by Chordino.

2.1.3 Large Chord Vocabulary And Class Reweighting

A chord vocabulary is the collection of different chords before or after chord mapping is performed. Most estimators working on the MajMin task tend to categorize chords into 25 classes (12 major, 12 minor and 1 no-chord). However, as errors and unbalanced subclasses (e.g. inverted chord subclasses) present in divided categories, the result is not always satisfactory. We adopted a different solution here: we regard different chord tags as distinct classes and no chord simplification is performed on complex chord types. Then the chord types with highest frequencies are reserved, where rare chord types are ignored and will be removed from the training set.

Within a certain chord type, the frequency of different chord roots varies. To solve this problem, a full shift of Chroma vector is performed to expand the training set. After expansion, all chord roots are on an equal footing.

As a response to the unbalance between chord types, a prior factor is attached to each type, and training samples are reweighted when building the random forest. More specifically, if the chord class Q_i has a prior factor of P_i, and N_i samples belong to the class, then the sample weights W_i are calculated as:

$$W_i = \begin{cases} \dfrac{P_i}{N_i} & \text{if } Q_i \text{ is no-chord} \\ \dfrac{P_i}{12N_i} & \text{otherwise} \end{cases}$$

2.2 The Beat-Chord Estimator

The relationship between chord change points and beat positions is commonly known, as chords tend to keep same during the beat period. In traditional beat-aligned chord estimators, the beat-tracking algorithm is performed first. The transition of chords is limited to the beat points, or they are readjusted during the post-processing. However, beat-tracking algorithms are not so reliable at the state of the art, leading to an advert effect on chord decoding sometimes. In this submission, we suggest a new way to extract beat positions and chord sequences at the same time using a CRF (Conditional Random Field) model, where beats and chords can correct each other during the joint estimation.

2.2.1 Model Details

The CRF model was mostly based on [2], and new hidden variables $\mathbf{C}$ (represent for the chord sequence) and new observation features $\mathbf{H}$ (represent for the chordogram) are introduced, as shown in figure 2.

And we use a Markovian form of CRF as in the original paper:

$$p(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})}\Phi(Y_1,\mathbf{X})\prod_{n=2}^{N}\Psi(Y_n,Y_{n-1})\,\Phi(Y_n,\mathbf{X})$$

The observation potentials $\Phi(Y_n,\mathbf{X})$ can be factorized into three terms:

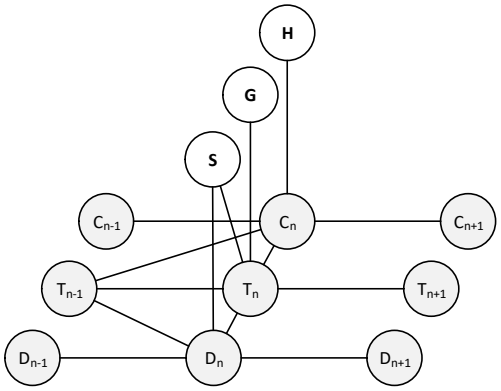

Figure 3. Graphical representation of the CRF model

$$\Phi(Y_n,\mathbf{X}) = \Phi_T(Y_n,G_n)\Phi_D(Y_n,\mathbf{S})\Phi_C(Y_n,H_n)$$

and

$$\begin{aligned} \Phi_T(Y_n,G_n) &= G_n[T_n] \\ \Phi_D(Y_n,\mathbf{S}) = D(T_n,D_n,\mathbf{S}) &= \mathrm{S}[n-D_n+1] * \Pi_{T_n}[n] \\ \Phi_C(Y_n,H_n) &= H_n[C_n]^{\gamma_c} \end{aligned}$$

where $\gamma_c > 0$ is the chord emission weight. A large γ_c indicates more confidence of chords information than beats information, and vice versa.

We made some slight changes to the convolution kernel $\Pi_T[n]$ in the second term than the original paper to make it less likely to sink into a wrong phase, and easier to calculate:

$$\begin{aligned} \Pi_T[n] = \delta[n] &+ \delta[n+T] + \delta[n-T] \\ &+ 1.5\gamma_h(\delta[n+T/2] + \delta[n-T/2]) \\ &+ 0.75\gamma_q(\delta[n+T/4] + \delta[n+3T/4] \\ &+ \delta[n-T/4] + \delta[n-3T/4]) \end{aligned}$$

where γ_h is the half-beat weight and γ_q is the quarter-beat weight, and $\delta[n]$ denotes the discrete impulse function.

The transition potentials $\Psi(Y_n,Y_{n-1})$, also, follows closely to the original paper, with the transition constraint of chord added:

$$\begin{aligned} &\Psi(Y_n,Y_{n-1}) \\ &= \begin{cases} h_t(T_{n-1},T_n)c_t(C_{n-1},C_n) & \text{if } D_{n-1}=T_{n-1} \text{ and } D_n=1 \\ 1 & \text{if } D_n=D_{n-1} \text{ and } T_n=T_{n-1} \text{ and } C_n=C_{n-1} \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

where $h_t(T_{n-1},T_n)$ and $c_t(C_{n-1},C_n)$ are the tempo transitions penalty term and chord transitions penalty term respectively. The first term complies with:

$$h_t(T_{n-1},T_n) = \begin{cases} \exp\left(-\gamma_t\left|\log\dfrac{T_{n-1}}{T_n}\right|^2\right) & \text{if } \dfrac{T_{n-1}}{T_n} < 2 \\ \exp(-\gamma_t|\log 2|^2) & \text{if } \dfrac{T_{n-1}}{T_n} > 2 \end{cases}$$

To get the optimal path of the hidden variables $\mathbf{X}$, the Viterbi algorithm is used. However, the naïve implementation of Viterbi algorithm has a time complexity of

$\Theta(NC^2T^3)$ and a space complexity of $\Theta(NCT^2)$, where N is the total frame number, C is the size of the chord vocabulary, and

$$\begin{aligned} T &= \max T_n - \min T_n \\ &= \frac{60 \cdot \mathrm{SR}}{\mathrm{win\_shift} \cdot \mathrm{min\_BPM}} - \frac{60 \cdot \mathrm{SR}}{\mathrm{win\_shift} \cdot \mathrm{max\_BPM}} \end{aligned}$$

The complexity is pretty unacceptable when the chord vocabulary is large. So we will briefly discuss some optimization in implementation here.

2.2.2 Algorithm Optimization

Let $\alpha_n(Y_n) = \alpha_n(T_n, D_n, C_n)$ denotes the forward variable in Viterbi algorithm. In implementation, the parameter D_n can be wiped out by only storing hidden states with $D_n = 1$. We define $g(n,t,c) \coloneqq \log \alpha_n(t,1,c)$ as the substate in a dynamic programming problem, and the transition formula can be rewritten as following:

$$\begin{aligned} g(n,t,c) &= \max_{t',c'} g(n-t',t',c') + \log h_t(t',t) + \log h_c(c',c) \\ &\quad + \sum_{i=n-t'}^{n-1} [\log G_i[t'] + \log D(t', i-n+t'+1, \mathbf{S}) \\ &\quad + \gamma_c \log H_i[c']] \\ &= \max_{t',c'} g(n-t',t',c') - \gamma_t \min\{(\log t' - \log t)^2, \log^2 2\} \\ &\quad + (\log h_c(c',c) + \gamma_c \sum_{i=n-t'}^{n-1} \log H_i[c']) \\ &\quad + \sum_{i=n-t'}^{n-1} [\log G_i[t'] \\ &\quad + \log D(t', i-n+t'+1, \mathbf{S})] \end{aligned}$$

As we can pre-calculate the prefix sum of two sigma terms respectively, both terms can be calculated in $\Theta(1)$ (e.g. if we define $H_s[n,c] \coloneqq \sum_{i=0}^{n} \log H_i[c]$, then we can get $\sum_{i=n-t'}^{n-1} \log H_i[c'] = H_s[n-1,c'] - H_s[n-t'-1,c']$), so the total time complexity is reduced to $\Theta(NC^2T^2)$.

Next, if we assume that for any different chord, transition penalty $c_t(c',c)$ is always a constant number γ_d (i.e. $c_t(c',c) = \gamma_d \quad \forall c' \neq c$), then it is not necessary to enumerate the c'. Define and pre-calculate

$$best\_c(n,t) \coloneqq \max_c g(n-t,t,c) + \log \gamma_d + \gamma_c \sum_{i=n-t}^{n-1} \log H_i[c]$$

Then, $g(n,t,c)$ is the maximum of the following two cases:

$$\begin{aligned} g_{diff}(n,t,c) &\coloneqq \max_{t'} best\_c(n,t') - \gamma_t \min\{(\log t' - \log t)^2, \log^2 2\} \\ &\quad + \sum_{i=n-t'}^{n-1} [\log G_i[t'] \\ &\quad + \log D(t', i-n+t'+1, \mathbf{S})] \\ g_{same}(n,t,c) &\coloneqq \max_{t'} g(n-t',t',c) - \gamma_t \min\{(\log t' - \log t)^2, \log^2 2\} \\ &\quad + (\log h_c(c,c) + \gamma_c \sum_{i=n-t'}^{n-1} \log H_i[c]) \\ &\quad + \sum_{i=n-t'}^{n-1} [\log G_i[t'] \\ &\quad + \log D(t', i-n+t'+1, \mathbf{S})] \\ g(n,t,c) &= \max\{g_{diff}(n,t,c), g_{same}(n,t,c)\} \end{aligned}$$

This is the version that the submission implemented. It has a time complexity of $\Theta(NCT^2)$ and a space complexity of $\Theta(NCT)$. However, the time complexity can be further optimized to $\Theta(NCT)$ due to the fact that log is a monotonous function and $\log h_c(c',c)$ is a segmented quadratic function of $\log t'$ and $\log t$, so a convex hull trick [3] can be used here to speed up dynamic programming.

2.2.3 Chord Transition Inside Beat Periods

In real-world cases, the change point of a chord can be inside a beat period sometimes. The most common circumstance is, the chord changes right in the middle of two beat positions. To allow this to happen, a new transition rule is added as a supplement. Chords can change in half-beat positions with a higher transition penalty given, to limit its occurrence.

Chord type	Sample count	Class Prior
maj	1443840	0.0784
min	367413	
N	141463	0.0065
7	257861	0.0523
min7	231326	
maj7	76010	
maj/5	50048	0.0261
maj/3	35608	
min/5	7999	
min/b3	7695	
7(#9)	34636	0.0261
maj(9)	34284	
maj6	26777	
min9	25543	
maj/2	23094	
maj9	20441	
min/b7	13999	
11	11567	
maj/4	10350	
maj/b7	9774	
13	9039	
9	7924	
maj6(9)	7852	
min11	6998	
min/4	6741	
min7/5	4877	
min6	4666	
7/3	4483	
7/5	4110	
7(b9)	3558	
maj/7	3537	
min/6	3222	

Table 1. Chord types that used for training, and their prior factors

3. TRAINING

3.1 Training Methods

As mentioned in section 2.1.3, we trained the random forest model on a full version of chord annotations.

To avoid any overlapping with the test datasets in MIREX contest, only 680 annotations from the Billboard dataset [4] (from No. 1 to No. 1000) and their Chroma features (provided by makers of Billboard dataset) are used for training. The chord types shown in Table 1 are reserved. Other chord types are discarded because of insufficient samples and will not appear in the output of the algorithm.

Because chord types like `sus4` or `dim` are not recognized in the MIREX contest, they are removed even though the samples are in large quantities.

4. RESULTS

We performed the evaluation of chord recognition accuracy on the RWC dataset [5]. The WAOR (Weighted Average Overlap Ratio) score is calculated when mapping to four main chord vocabularies in MIREX contest. The results are shown in Table 2.

Acc. %	Chordino	RF-HMM	**RF-BC**
MajMin	78.3	81.6	**82.1**
MajMinBass	72.3	79.0	**79.5**
Sevenths	53.1	69.7	**70.2**
SeventhsBass	47.7	67.4	**67.9**

Table 2. WAOR score of tested algorithms

The system will be further evaluated and optimized in future work, as the version we submitted for MIREX contest is still a prototype. The estimation of beat accuracy will also be added in the future.

5. REFERENCES

[1] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proceedings of the International Symposium on Music Information Retrieval 2010*, pp. 135–140, 2010.

[2] T. Fillon, C. Joder, S. Durand, and S. Essid. "A conditional random field system for beat tracking," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[3] Z. Galil and K. Park, "Dynamic programming with convexity, concavity and sparsity," in *Theoretical Computer Science, 92*(1), pp. 49-76, 1992.

[4] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga, "An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, ed. Anssi Klapuri and Colby Leider (Miami, FL, 2011), pp. 633–38.

[5] M. Goto, "Development of the RWC music database," in *Proceedings of the 18th International Congress on Acoustics (ICA 2004)*, pp. 553-556, 2004-April.