HARMONY PROGRESSION ANALYZER FOR MIREX 2011

Yizhao Ni[1], Matt Mcvicar[1], Raul Santos-Rodriguez[2] and Tijl De Bie[1]

1. Intelligent Systems Lab
Department of Engineering Mathematics
University of Bristol, U. K.

2. Signal Theory and Communications Department
Universidad Carlos III de Madrid
Spain

ABSTRACT

We present a new system, *Harmony progression (HP) analyzer*, for simultaneous estimation of keys, chords, and bass notes from music audio. It makes use of a novel chromagram of audio that takes perception of loudness into account. Furthermore, it is fully based on machine learning (ML), such that it is potentially applicable to a wider range of genres as long as training data is available. As compared to other models, the proposed system is fast and memory efficient, while achieving state-of-the-art performance.

1. SYSTEM DESCRIPTION

1.1 Loudness based chromagram

Let $\mathbf{x} = [x_1, \ldots, x_T]$ be an audio signal with x_t indicating the sample data of the t-th frame, then the chromagram extraction assigns attributes (e.g. power or amplitude) $\mathbf{X} \in \mathbb{R}^{S\times T}$ to a set of frequencies $F = \{f_1, \ldots, f_S\}$ such that $\mathbf{X}$ reflects the energy distribution of the audio along these frequencies. In order to capture musically relevant information, the frequencies are selected from the equal-tempered scale, which may be tuned [7] and vary between songs. Popular implementations of chromagram extraction are *fixed bandwidth Fourier* [6] and *constant Q* [1] transforms.

The above two chromagrams and their variants represent the salience of pitch classes in terms of a power or amplitude spectrum. We note however that perception of loudness is not linearly proportional to the power or amplitude spectrum, and hence such chromagram representations do not accurately represent human perception of the audio's spectral content. Indeed, the empirical study in [5] showed that loudness is approximately linearly proportional to so-called *sound power level*, defined as $\log_{10}$ of power spectrum. Therefore, we developed a novel *loudness based chromagram*, which uses the $\log_{10}$ scale of power spectrum. Mathematically, a sound power level (SPL) matrix is of the form

$$\mathcal{L}_{s,t} = 10\log_{10}\left(\frac{\|X_{s,t}\|^2}{p_{ref}}\right),\ s = 1,\ldots,S, t = 1,\ldots,T,$$

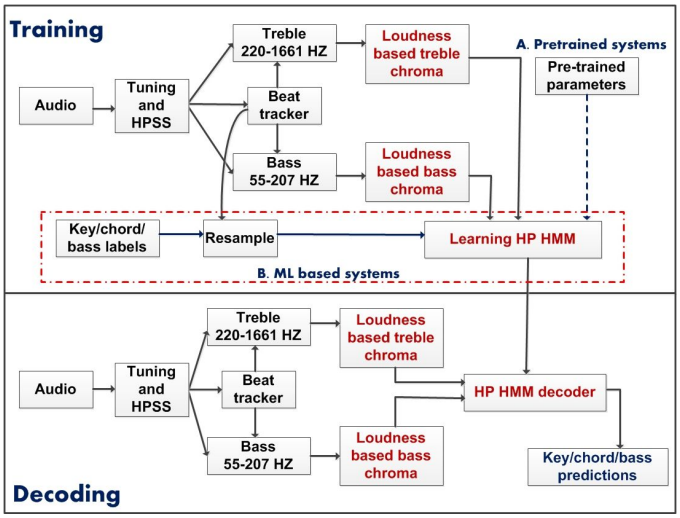

Figure 1. The learning procedure (via Approach B) of the proposed Harmony Progression (HP) system. The blocks in red show the novelties of the system.

where p_{ref} indicates the fundamental reference power and

$$X_{s,t} = \sum_{n=t-\frac{L_s}{2}}^{t+\frac{L_s}{2}} x_n w_n \exp\left(\frac{-2\pi s t}{L_s}\right)$$

is a constant Q transform with a frequency dependent bandwidth $L_s = Q\frac{SR}{f_s}$ [1] and the hamming window w_n [1].

Furthermore, low/high frequencies require higher sound power levels for the same perceived loudness as mid-frequencies [5]. To compensate for this, we propose to use *A-weighting* [14] to transform the SPL matrix into a representation of the perceived loudness of each of the pitches:

$$\mathcal{L}'_{s,t} = \mathcal{L}_{s,t} + A(f_s),\ s = 1,\ldots,S, t = 1,\ldots,T,$$

where

$$R_A(f_s) = \frac{12200^2\cdot f_s^4}{(f_s^2+20.6^2)\cdot\sqrt{(f_s^2+107.7^2)(f_s^2+737.9^2)}\cdot(f_s^2+12200^2)}$$
$$A(f_s) = 2.0 + 20\log_{10}(R_A(f_s)).$$

It is known that loudnesses are additive if they are not close in frequency [13]. This allows us to sum up loudness of sounds on the same pitch class, yielding:

$$X'_{p,t} = \sum_{s=1}^{S}\delta(M(f_s), p)\mathcal{L}'_{s,t},\ p = 1,\ldots,12, t = 1,\ldots,T.$$

[1] Q is the constant resolution fact and SR is the sampling rate of $\mathbf{x}$.

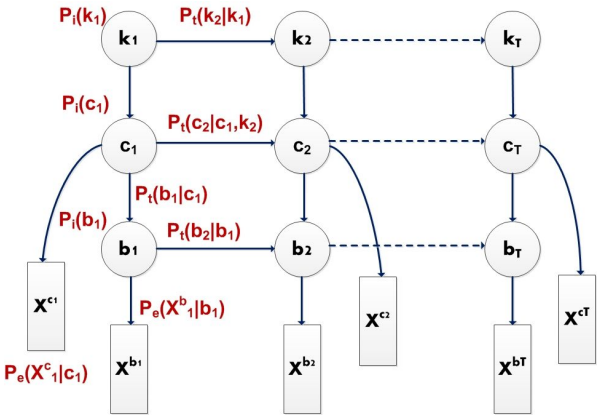

Figure 2. The HMM topology of the HP system. The probabilities in red are parameters of the system, which are learnt via maximum likelihood estimation (MLE).

Here δ denotes an indicator function and

$$M(f_s) = \left(\left\lfloor 12 \log_2 \left(\frac{f_s}{f_A} \right) + 0.5 \right\rfloor + 69 \right) \bmod 12$$

with f_A denoting the reference frequency of the pitch $A4$ (440Hz in standard pitch). Finally, our loudness-based chromagram, denoted $\bar{X}_{p,t}$, is obtained by normalizing $X'_{p,t}$ using:

$$\bar{X}_{p,t} = \frac{X'_{p,t} - \min_{p'} X'_{p',t}}{\max_{p'} X'_{p',t} - \min_{p'} X'_{p',t}}.$$

Note that this normalization is invariant to the reference power and hence a specific p_{ref} is not required.

1.2 HP HMM topology

The HP HMM topology consists of three hidden and two observed variables. The hidden variables correspond to the key $\mathcal{K} = \{\mathbf{k}^n \in \mathcal{A}_k^{1\times T_n}\}_{n=1}^N$, the chord $\mathcal{C} = \{\mathbf{c}^n \in \mathcal{A}_c^{1\times T_n}\}_{n=1}^N$ and the bass annotations $\mathcal{B} = \{\mathbf{b}^n \in \mathcal{A}_b^{1\times T_n}\}_{n=1}^N$. Under this representation, a chord is decomposed into two aspects: chord label and bass note. Take the chord A:maj/3 for example, the chord state is c = A:maj and the bass state is b = C#. Accordingly, the observed chromagrams are decomposed into two parts: the treble chromagram $\bar{\mathbf{X}}^{\mathbf{c}}$ which is emitted by the chord sequence $\mathbf{c}$ and the bass chromagram $\bar{\mathbf{X}}^{\mathbf{b}}$ which is emitted by the bass sequence $\mathbf{b}$. The reason of applying this decomposition is that different chords can share the same bass note, resulting in similar chromagrams in low frequency domain.

Under this framework, the set Θ of a HP HMM has the following parameters

$$\Theta = \begin{array}{l} \big\{p_i(k_1), p_i(c_1), p_i(b_1), p_t(k_t|k_{t-1}), p_t(c_t|c_{t-1}, k_t), \\ p_t(b_t|c_t), p_t(b_t|b_{t-1}), p_e(\bar{\mathbf{X}}_t^{\mathbf{c}}|c_t), p_e(\bar{\mathbf{X}}_t^{\mathbf{b}}|b_t)\big\}, \end{array}$$

where p_i, p_t and p_e denote the initial, transition and emission probabilities respectively. The joint probability of the feature vectors $\{\bar{\mathbf{X}}^{\mathbf{c}}, \bar{\mathbf{X}}^{\mathbf{b}}\}$ and the corresponding annotation sequences $\{\mathbf{k}, \mathbf{c}, \mathbf{b}\}$ of a song is then given by the formula

$$P(\bar{\mathbf{X}}^{\mathbf{c}}, \bar{\mathbf{X}}^{\mathbf{b}}, \mathbf{k}, \mathbf{c}, \mathbf{b}|\Theta) = p_i(k_1)p_i(c_1)p_i(b_1) \prod_{t=2}^{T} p_t(k_t|k_{t-1})$$
$$p_t(c_t|c_{t-1}, k_t)p_e(\bar{\mathbf{X}}_t^{\mathbf{c}}|c_t)p_t(b_t|c_t)p_t(b_t|b_{t-1})p_e(\bar{\mathbf{X}}_t^{\mathbf{b}}|b_t).$$

The initial probabilities $p_i(\star)$ can be learnt via *maximum likelihood estimation* (MLE). For example, $p_i(c) = \frac{\#(c_1=c)}{\#c_1}$ $\forall c \in \mathcal{A}_c$, where $\#$ indicates the number of.

For the transitions, $p_t(c|\bar{c}, k)$ represents the probability of a chord change under a certain key. Since the chord transition is strongly influenced by the underlying key [10], this probability is modelled as key dependent. Under the assumption that relative chord transitions are key independent, we transposed all sequences to a common key k and learn $p_t(c|\bar{c}, k)$ from the transposed sequences. This allowed us to get 12 times as much information from the data source and the MLE solution is

$$p_t(c|\bar{c}, k) = \frac{\#(c_t = c \;\&\; c_{t-1} = \bar{c} \;\&\; k_t = k)}{\sum_{c'} \#(c_t = c' \;\&\; c_{t-1} = \bar{c} \;\&\; k_t = k)}, \forall c, \bar{c}, k.$$

Similarly, $p_t(k|\bar{k})$ is applied to model key changes during a song. $p_t(b|c)$ models the probability of a bass note under a chord label so as to capture chord inversions. A transition link $p_t(b|\bar{b})$ is also added, with the purpose of modelling the continuity of bass notes and capturing ascending and descending bassline progressions. These parameters are learnt via MLE, e.g. $p_t(k|\bar{k}) = \frac{\#(k_t=k \;\&\; k_{t-1}=\bar{k})}{\sum_{k'} \#(k_t=k' \;\&\; k_{t-1}=\bar{k})}, \forall k, \bar{k} \in \mathcal{A}_k$.

Finally, emission probabilities $p_e(\bar{\mathbf{X}}_t^{\mathbf{c}}|c_t)$ and $p_e(\bar{\mathbf{X}}_t^{\mathbf{b}}|b_t)$ are modelled as 12-dimensional Gaussians, of which the mean vectors and covariance matrices are learnt via MLE as well.

1.3 Search space reduction

Given the optimal parameters Θ^* via MLE, the decoding task can be formalized as the computation of the key, chord and bass sequences $\{\mathbf{k}^*, \mathbf{c}^*, \mathbf{b}^*\}$ that maximize the joint probability $\{\mathbf{k}^*, \mathbf{c}^*, \mathbf{b}^*\} = \arg\max_{\mathbf{k},\mathbf{c},\mathbf{b}} P(\bar{\mathbf{X}}^{\mathbf{c}}, \bar{\mathbf{X}}^{\mathbf{b}}, \mathbf{k}, \mathbf{c}, \mathbf{b}|\Theta^*)$.

This task can be solved using the *Viterbi* algorithm [12], whose computational complexity is $O\big(|\mathcal{A}_k|^2|\mathcal{A}_c|^2|\mathcal{A}_b|^2|T|\big)$. This is a huge search space, especially when one would like to use a large chord vocabulary [9]. In order to reduce the decoding time, we propose three ML constraints:

1.3.1 Chord alphabet constraint (CAC)

It is unlikely that all chords will be used in a single song. Therefore, if it is possible to find out which chords are used in a song, we will be able to constrain the chord alphabet without loss of performance. One ML method is to utilize two-stage predictions. In particular, using a simple HMM with only chords as the hidden chain, we first apply a max-Gamma decoder [12] to a song and obtain the most probable chords $\mathcal{A}'_c$. Then, we force the HP HMM chord transition probability to be zero for chords that are absent in this output:

$$p'_t(c|\bar{c}, k) = \begin{cases} p_t(c|\bar{c}, k) & \text{if } c, \bar{c} \in \mathcal{A}'_c \\ 0 & otherwise \end{cases}.$$

1.3.2 Key transition constraint

Music theory dictates that not all key changes are equally likely. If a song does change key, the modulation is most likely to move to a related key [8]. Thus, we suggest to rule out a priori the key transition that are seen the least often in the training set. Formally, this can be done by constraining the key transition probability as

$$p'_t(k|\bar{k}) = \begin{cases} p_t(k|\bar{k}) & \text{if } \#(k_t = k \;\&\; k_{t-1} = \bar{k}) > \gamma \\ 0 & otherwise \end{cases},$$

where γ is a positive integer indicating the threshold.

1.3.3 Chord to bass transition constraint

Similar to the key transition constraint, we can also constrain the chord to bass transitions. A constraint is imposed on $p_t(b|c)$ such that the bass notes can only be one of τ ($\tau \leq 12$) candidates for a given chord. The frequencies of each chord-to-bass emission are ranked and only the most common τ are permissible. Mathematically:

$$p'_t(b|c) = \begin{cases} p_t(b|c) & \text{if } b \text{ is one of the top } \tau \text{ bass notes for } c \\ 0 & otherwise \end{cases}.$$

When $\tau = 3$, the constraint is equivalent to using root position, first and second inversions of a chord.

2. EXPERIMENTS

2.1 Audio dataset and ground truth annotations

The audio dataset used is the one used in the MIREX Audio Chord Estimation task 2010 [2], which contains 217 songs. The ground truth key and chord annotations were obtained from `http://isophonics.net`, while the bass notes are extracted directly from the ground truth chord annotations.

2.2 Preprocessing and chromagram feature extraction

As shown in Figure 1, we first converted our signals to mono 11025 Hz, and separated the harmonic and percussive elements with the Harmonic/Percussive Signal Separation algorithm (HPSS) [11]. After tuning [7] we computed loudness based chromagrams for each song. The frequency range of the bass chromagram was $A1$ to $G\sharp 3$ (55Hz - 207.65Hz), and that of the treble chromagram was $A3$ to $G\sharp 6$ (220Hz - 1661.2Hz). Finally, we estimated beat positions using the beat tracker presented in [3] and took the median chromagram feature between consecutive beats. We also beat synchronized our key/chord/bass annotations by taking the most prevalent labels between beats. The median feature vector with the corresponding beat-synchronized annotations is then regarded as one frame.

[2] `http://www.music-ir.org/mirex/wiki/2010:Audio_Chord_Estimation`

2.3 Major/minor chord prediction

In this experiment, we used a full key alphabet (12 major and 12 minor keys), but restricted ourselves to a chord alphabet of 25 chords (12 major, 12 minor and no-chord). There were 13 bass states corresponding to the 12 pitch classes as well as a 'no bass'. In accordance with the MIREX train-test setup, we randomly split $2/3$ of songs from each album to form the training set, while the remaining $1/3$ were used for testing. The same chord evaluation metric used in MIREX competition 2010 (denoted by 'OR' and 'WAOR' [3]) was applied to report chord prediction performance. The experiment was repeated 102 times to access variance.

To compare chord predictions, a standard HMM system (denoted as HMM-C) is taken as the baseline, where the observed variable is a concatenation of treble and bass chromagrams and the hidden states are 25 chords. Meanwhile, the performances of the LabROSA system [4] [4] (denoted as labROSA) and a key-specific HMM (denoted as K-HMM [5]) are also reported.

	HMM-C	labROSA	K-HMM	HP
OR	77.82^*	74.21^*	78.22^*	**79.37**
WAOR	77.22^*	73.22^*	77.62^*	**78.82**

Table 1. Performances [%] for the baseline, labROSA, key-specific HMM and HP systems on the major/minor chord prediction task. Bold numbers indicate the best results. The improvement of HP is significant at a level $< 10^{-45}$ over the performances marked by *.

Table 1 shows the results and the significance of the improvement of the HP system over the other systems assessed using a paired t-test. The first row shows the results of the standard HMM chord prediction system using loudness based chromagram. This simple system already outperforms labROSA that utilizes a more powerful discriminative HMM learning agent, verifying the effectiveness of the novel loudness based chromagram extraction. HP also outperforms K-HMM, indicating that separating treble and bass contents could help harmonic estimation.

We then investigated the proposed search space reduction techniques. Figure 3 (a) shows that using a reasonable cutoff γ can reduce the decoding time dramatically while retaining a high performance. The same trend is also observed when applying a reasonable τ to the chord to bass transition constraint (red dot curves in Figure 3 (b)). Furthermore, using a chord alphabet constraint (solid curves in Figure 3 (b)) did not decrease the performance (in fact it had a slight improvement), although the decoding time is also reduced significantly. To summarize, by applying all these ML techniques, we are able to speed up decoding

[3] 'OR' refers to *chord overlap ratio* in MIREX 2010 evaluation and 'WAOR' refers to *chord weighted average overlap ratio*.

[4] The system uses $12\times$ data transposition to enhance performance, which is equivalent to their *EW1* submission in MIREX 2010.

[5] Note that K-HMM is essentially equivalent to [2], except that it makes use of our loudness based chromagrams.

without decreasing the performance. Thanks to this, we can also apply HP to more complex chord representations (e.g. the 121 chord vocabulary used in [9]).

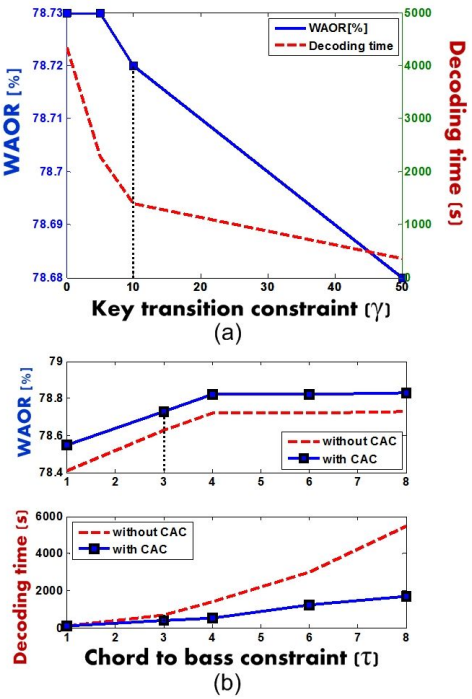

Figure 3. The performances and decoding times of HP using different search space reductions. The experiments in (a) were done without chord alphabet constraint and τ is fixed at 4. In (b), 'CAC' refers to chord alphabet constraint and the experiments were carried out with γ fixed at 10.

3. CONCLUSIONS AND MIREX 2011

We propose a novel key, chord and bass simultaneous recognition system – the HP system – that purely relies on ML techniques. The experimental results verify that the HP system can achieve the state-of-the-art performance on chord recognition, and it can be sped up significantly using the search space reduction techniques without severely decreasing the performance.

For the MIREX 2011 Audio Chord Estimation evaluation, we submitted four systems:

- HP_feng - a train-test system using all techniques described in Section 1. Since there is no key information provided, we assume the keys for all train/test songs are the same.
- HP_lin - a pre-trained system equivalent to HP-P in Table 1.
- HP_huo - a pre-trained system essentially equivalent to HP-P, except that it is trained on a complex chord vocabulary (i.e. 121 chord classes used in [9]) and the output is also in complex chord format.
- HP_shan - a pre-trained system on the basis of HP_lin, which uses a rule-based post-processing to refine chord predictions.

4. REFERENCES

[1] J. Brown. Calculation of a constant q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

[2] B. Catteau, J. Martens, and M. Leman. A probabilistic framework for audio-based tonal key and chord recognition. In *Proc. of GfKl*, pages 637–644, 2006.

[3] D. Ellis and G. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proc. of ICASSP*, pages 1429–1433, 2007.

[4] D. Ellis and A. Weller. The 2010 LABROSA chord recognition system. In *Proc. of ISMIR (MIREX)*, 2010.

[5] H. Fletcher. Loudness, its definition, measurement and calculation. *Journal of the Acoustical Society of America*, 5(2):82, 1933.

[6] T. Fujishima. Real time chord recognition of musical sound: a system using common lisp music. In *Proc. of ICMC*, pages 464–467, 1999.

[7] C. Harte and M. Sandler. Automatic chord identification using a quantised chromagram. In *Proc. of the Audio Engineering Society*, 2005.

[8] C. L. Krumhansl. *Cognitive foundations of musical pitch*. Oxford University Press, 1990.

[9] M. Mauch. *Automatic chord transcription from audio using computational models of musical context*. PhD thesis, Queen Mary University of London, 2010.

[10] K. Noland and M. Sandler. Key estimation using a hidden markov model. In *Proc. of ISMIR*, 2006.

[11] N. Ono, K. Miyamoto, J. Roux, H. Kameeoka, and S. Sagayama. Separation of a monaural audio signal into harmonic/percussive components by complimentary diffusion on spectrogram. In *Proc. of EUSIPCO*, 2008.

[12] L. R. Rabiner. A tutorial on hidden markov models and selected application in speech recognition. In *Proc. of the IEEE*, 1989.

[13] T. D. Rossing. *The science of sound (second edition)*. Addison-Wesley, 1990.

[14] M. T. Smith. *Audio engineer's reference book*. Focal Press, 1999.