調波的非負行列近似と階層的
隠れマルコフモデルによる多重ピッチ分析

ラチンスキ スタニスワヴ アンジェイ, 小野 順貴, 嵯峨山 茂樹

東京大学
大学院情報理工学系研究科
システム情報学専攻

Multipitch analysis by Harmonic Nonnegative Matrix Approximation and Hierarchical Hidden Markov Models

Stanisław Andrzej Raczyński, Nobutaka Ono, Shigeki Sagayama

The University of Tokyo
Graduate School of Information Science and Technology
Department of Information Physics and Computing

Abstract

We propose a new approach for dealing with multipitch analysis of musical signals that makes use of the fact that such signals are highly structured. This structure comes from the many musicological rules of the western tonal music, and we model it by using the recently developed method of Hierarchical Hidden Markov Models. We propose a model with four layers: song, key, chord, and note combination layer. One of the big advantage of this approach is that, besides from information about pitches, we get higher level musical information about chord progression and key modulation.

1 Introduction

Automatic music transcription of recorded music is usually a two stage process. The first stage is the event detection phase, where music events (note onsets, note offsets, pitch changes) are detected and identified. In the second stage, these events are transformed into a musical score. This paper focuses on the event detection stage, main part being multipitch analysis, which aims to uncover the fundamental frequencies of simultaneously played harmonic sounds.

Different approaches has been used to deal with the task of multipitch analysis, but recently some attention (e.g. [1]) is given to develop methods that would, as do human transcribers, use higher level musicological knowledge in the process. Researchers from other areas of music analysis share this tendency. The most popular models used for this purpose are Hidden Markov Models [2, 3, 4, 5, 1, 6] (to mention just a few), Probabilistic Context-Free Grammars [7] and, recently, Hierarchical Hidden Markov Models [8]. The latter is gaining popularity due to low complexity as compared to Probabilistic Grammars and ability to capture more signal structure than Hidden Markov Models.

This paper is organized as follows. Sections 5, 3 and 5 briefly describe the methods of Harmonic Nonnegative Matrix Approximation (HNNMA), Hidden Markov Models (HMMs) and Hierarchical Hidden Markov Models (HHMMs). Section 4 contains arguments against direct application of HMMs to the problem of multipitch analysis. Finally, section 6 describes an experiment of using the proposed method to analyze a simple piece of music. Conclusion is given in section 7.

2 Harmonic Nonnegative Matrix Approximation

Harmonic Nonnegative Matrix Approximation [9] is a modification of the Nonnegative Matrix Approximation (NNMA, described in [10]). NNMA is a method for decomposition of a nonnegative (having only nonnegative elements) matrix $\mathbf{X}$ (later referred to as the data matrix) into a multiplication of two, also nonnegative, matrices $\mathbf{X} \cong \mathbf{AS} = \widetilde{\mathbf{X}}$ (later referred to as the basis matrix and the activity matrix, respectively). The NNMA solves this problem by minimizing a Bregman divergence between the data matrix $\mathbf{X}$ and its approximation $\widetilde{\mathbf{X}}$. A special case of Bregman divergence is the I-divergence (generalized Kullback-Leibler divergence):

$$D_{KL}(\mathbf{P}, \mathbf{Q}) = \left| \mathbf{P} \odot \log \frac{\mathbf{P}}{\mathbf{Q}} - \mathbf{P} + \mathbf{Q} \right|, \tag{1}$$

where the logarithm, multiplication (denoted by $\odot$) and the division are calculated element-wise. Using the I-divergence leads to the Nonnegative Matrix Factorization (NMF), for which Lee and Seung [11] has proposed a very fast multiplicative update algorithm. A very similar algorithm exists for different Bregman divergences as well. Also, under particular assumptions, this algorithm can be extended to minimize an objective function containing additional penalizing terms [9]. The final algorithm takes form of two update rules performed alternately:

$$\mathbf{A} \leftarrow \mathbf{A} \odot \frac{\frac{\mathbf{X}}{\mathbf{AS}} \mathbf{S}^T}{\mathbf{1}\mathbf{S}^T + \nabla_{\mathbf{A}} \alpha(\mathbf{A})}, \tag{2}$$

$$\mathbf{S} \leftarrow \mathbf{S} \odot \frac{\mathbf{A}^T \frac{\mathbf{X}}{\mathbf{AS}}}{\mathbf{A}^T \mathbf{1} + \nabla_{\mathbf{S}} \beta(\mathbf{S})}, \tag{3}$$

where $\alpha(\mathbf{A})$ and $\beta(\mathbf{S})$ are the minimized penalizing functions.

Such update rules are used in HNNMA together with a special initialization of the basis matrix. Because zero-valued elements of basis vectors will remain zero-valued throughout the learning process (eq. 2 and 3), we can initialize the basis matrix to have zeros everywhere but at the positions of fundamentals of notes from a specific range of the equal temperament scale and their harmonics. That would guarantee that the basis vectors are sorted by their fundamental frequencies, and that corresponding rows in the activity matrix contain activities of consequent notes from that range, resulting in a harmonically-constrained NNMA. This would make analysis of the results of the algorithm straightforward – one would only have to analyze the note activities and find peaks corresponding to instances of these notes. This technique is a good tradeoff between full basis estimation methods (such as NMF and other NNMA-based approaches) and methods that use pre-learned basis vectors.

3 Hidden Markov Models

Hidden Markov Models (HMMs) are statistical models that belong to the big family of Dynamic Bayesian Networks. They are often thought of as a probabilistic generalization of the Finite State Machines. They consist of a finite number of unobserved (hidden) states, each of which can generate a single symbol (output) at a time. An extension to HMMs allows them to generate continuous output (that doesn't belong to a finite state of symbols), which is usually modelled using Gaussian Mixture Models. HMMs work very well for modeling time time sequences and have found applications in a huge number of fields, particularly in signal processing.

4 Feasibility considerations

Hidden Markov Models with continuous output have been successfully used for speech recognition, so they probably would yield good results in the task of music recognition. The big problem is, however, the number of states. In speech recognition, this number is defined by the number of phonemes in particular language, which, in most languages, is equal to about 30-50. In recognition of monoinstrumental music, one state corresponds to a single combination of notes. Number of all possible note combinations is enormous – for piano, for example, is equal to about 18 quintillions (assuming that each of the 10 fingers can either press one of the keys, different from the keys pressed by the other fingers, or stay in the air). The Hidden Markov Model cannot be directly applied to music recognition. However, we could reduce the complexity of problem by:

- allowing only musically correct transitions,
- allowing only the most probable note combinations,
- grouping note combinations, and treating combinations inside groups equally (assign them equal transition probabilities).

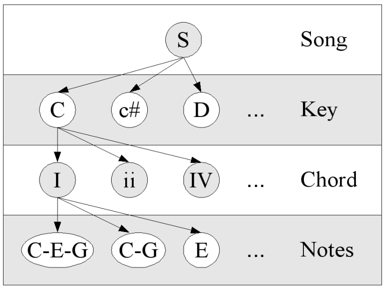

Figure 1: Proposed four-layer state hierarchy of musical signals.

The first point means that we should use higher level knowledge of music to reduce the overall number of possible transitions to musicologically correct minimum. A quite recently developed generalization of the regular Hidden Markov Models, called Hierarchical Hidden Markov Models (HHMMs), seems to be perfect for this task. It introduces state hierarchy, which happens to suit musical needs very well. All HHMMs can be transformed to a regular flat HMM, which is not fully connected (only some transitions are allowed). This model is further described in section 5. The second suggestion allows us to reduce the number of all possible note combinations to an implementable figure. This can be done by performing a pre-scanning of the analyzed data and selecting only those combinations that are most plausible. This is discussed in more detail in section 6. The third point aims at reducing the number of degrees of freedom of the model – only very few transition probabilities and initial probabilities will need to be learned. In our approach the following groups are used:

1. out-of-key combination (at least one note in the combination doesn't belong to the key),
2. non-chord combinations (at least one note in the combination doesn't belong to the chord),
3. chord combinations (all notes are in the key and belong to the chord),
4. a rest,
5. end state of the key.

However, non-harmonic combinations could be further split into such groups as anticipation note, neighbor note, passing note, escape note and pedal note, causing the HMM to better model musicological rules. All notes from within the same group are equally treated – they are equally probable, which seems like a very reasonable assumption. Including the state self-transition probability, it comes to 26 different transition probabilities, a number that allows model parameter learning even on a small amount of training data.

5 Hierarchical Hidden Markov Models

Hierarchical Hidden Markov Models (HHMMs) are a recently developed [12] generalization of regular Hidden Markov Models by introducing a hierarchical state structure. HHMMs are receiving more and more attention from researchers from various fields.

While in HMMs each state generates a single symbol, in HHMMs each state generates a whole sequence of symbols, and is in fact an HHMM on its own. Only the bottom-most level states (called production states) generate symbols.

We propose an HHMM with four layers (fig. 5): song layer (single master state), key layer (24 states corresponding to all the keys of the western tonal music), chord layer (7 triads for every scale degree, can also include higher chords), and note combination layer (one state for particular combination of notes played under the current chord-key pair), which is the production layer. Note combination states produce output, which can be a spectrum, or, in our case, single column of the coefficient matrix.

To learn parameters and to infere from a HHMM, the generalized Baum-Welch, and the generalized Viterbi algorithms [12] can be used. Unfortunately they are quite complex and difficult to implement, and the generalized Viterbi algorithm has $O(T^3)$ time complexity. A much faster ($O(T)$) algorithm has recently been developed [13] that uses standard methods of Dynamic Bayesian Networks, HHMM is a special case of which. For simple cases, an HHMM can be flattened to a regular HMM and standard Viterbi algorithm can be used.

	H	**NH**	**OOK**	**R**	**End**
H	0.625	0.125	0	0.125	0.125
NH	0.75	0	0	0.125	0.125
OOK	0	0	0	0	1
R	0.875	0	0	0	0.125

	I	**IV**	**V**
I	0	0.5	0.5
IV	1	0	0
V	1	0	0

Table 1: Group transition probabilities used in the experiment (left) and chord transition probabilities (right). H – in-chord combination, NH – non-chord combination, OOK – out-of-key combination, R – rest.

6 Experiment

To validate the proposed approach, we have tested the algorithm on a short, single-key, rhythmically-simple, artificially generated piece of music that uses the very common I-IV-I-V chord progression (see the bottom part of fig. 6).

The first processing stage involves calculating the Constant-Q Transform (CQT) of the input data in order to obtain the **X** matrix (topmost part of fig. 6). This matrix is then analyzed with the HNNMA method and the coefficient matrix **S** is generated (second from the top part of fig. 6). This matrix is first analyzed in the following manner to get the list of the most possible note combinations. For each time frame 5 peaks are located in the coefficient vector (a single column of the **S** matrix). This means that we assume that no more than 5 notes can be played simultaneously. Peaks are assigned to notes in the equal temperament scale and all possible combinations of them are created (1-, 2-, 3-, 4-, and 5-combinations) and added to the global note combination list. After removing duplicate items, this global list defines the production states of the HHMM.

Each chord-level HHMM state has the same set of production states, but different transition matrix and initial probability vector. Production states (note combination list) are analyzed in the context of the chord-key combination, and then grouped. The transition and initial probabilities are assigned according to the group the production state belongs to.

The output "probability" is calculated according to the following formula:

$$\mathbf{B}_{i,t} = \frac{\sum_{j \in combination_i} \mathbf{S}^2_{j,t}}{\sum_j \mathbf{S}^2_{j,t}} \cdot a^{-|combination_i|+1}. \tag{4}$$

It is a measure of fitness of particular note combination to a column of the coefficient matrix, penalized by the number of notes in the combination to avoid the overfitting problem. a is a trade-off factor between the goodness of fit and too complex note combinations.

In this experiment a simplified HHMM was used – it contained only 3 levels: song, chord (only 3 states: I, IV and V), and note combination (production level). A key of C-major was assumed. We used heuristically generated group and chord transition probabilities (table 6), however in future those parameters will be trained on reference data. This simple model can easily be flattened to a regular HMM, transition matrix of which is depicted on fig. 6. It is clearly visible that this matrix is highly structured – three squares along the main diagonal represent transitions inside the three chords, on the main diagonal there are the state self-transition probabilities, and outside of that are chord-to-chord transition probabilities (much lower than in-chord transition probabilities).

The results of using such an HHMM are presented on fig. 6 (third from the top). The are no spurious notes, but some of the notes are missing. This definitely could be improved by using better (trained) parameter values. The very big advantage of this method is the additional information about the chord and key throughout the piece of music. As we can see on fig. 6, the chords were identified correctly.

7 Conclusion and further work

Experimental results are promising, but answer only just the first very simple question about the proposed method. In order to truly evaluate this technique, tests need to be run using more complicated and fully-layered (including the key layer) HHMM with parameters learned on training data. The main problem is,

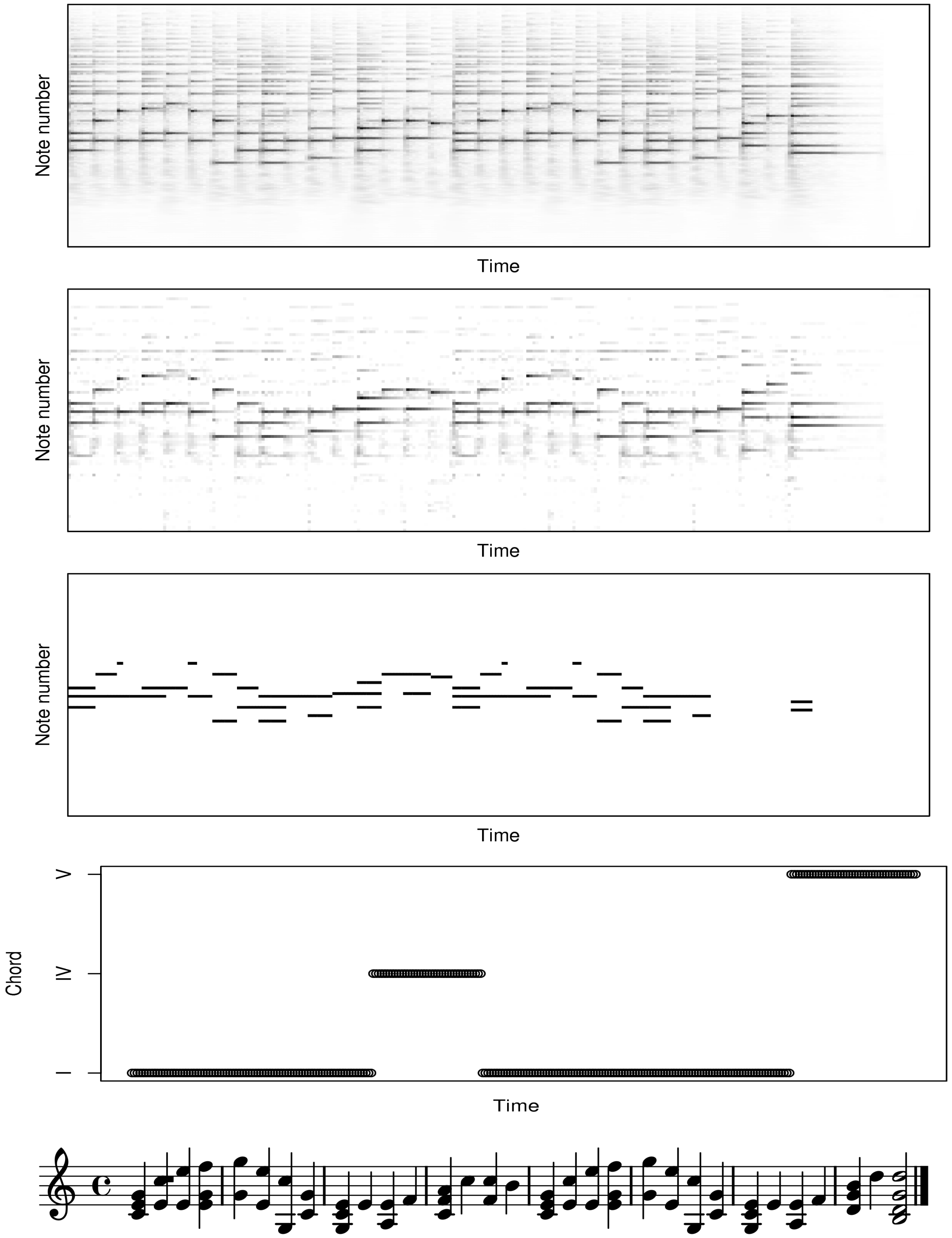

Figure 2: Analysis stages (from the top): Constant-Q-gram (**X**), activity matrix (**S**), the results of the HHMM algorithm, and the original score for reference.

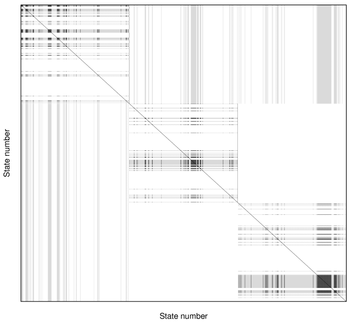

Figure 3: Transition probability matrix for the HHMM used in this experiment flattened to a regular HMM.

however, obtaining groundtruth data, since there currently is no database that stores notes and, at the same time, information about chord progression and key modulation.

References

[1] MP Ryynänen and A. Klapuri. Note Event Modeling for Audio Melody Extraction. *Proceedings of the 2005 Music Information Retrieval Exchange*, 2005.

[2] C. Raphael. Automatic segmentation of acoustic musical signals using hidden Markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):360–370, 1999.

[3] T. Kawakami, M. Nakai, H. Shimodaira, and S. Sagayama. Hidden Markov Model Applied to Automatic Harmonization of Given Melodies. *IEIC SIG Technical Reports*, 99(MUS-34):59–66, 2000.

[4] H. Takeda, N. Saito, T. Otsuki, M. Nakai, H. Shimodaira, and S. Sagayama. Hidden Markov model for automatic transcription of MIDI signals. *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 428–431, 2002.

[5] A. Sheh and D.P.W. Ellis. Chord Segmentation and Recognition using EM-Trained Hidden Markov Models. *Proc. Int. Conf. on Music Info. Retrieval ISMIR*, 3:185–191, 2003.

[6] H. Takeda, T. Nishimoto, and S. Sagayama. Automatic accompaniment system of MIDI performance using HMM-based score followng. *IPSJ SIG Technical Reports*, 90(MUS-66):109–116, 2006.

[7] T. Morooka, T. Nishimoto, and S. Sagayama. The automatic harmonic analysis using PCFG in consideration of nonharmoic tones. *Proc. of ASJ Spring Meeting*, 2-1-11:865–866, 2007.

[8] M. Weiland, A. Smaill, and P. Nelson. Learning Musical Pitch Structures with Hierarchical Hidden Markov Models. *Actes des Journées d'Informatique Musicale (JIM05)*, 2005.

[9] S.A. Raczyński, N. Ono, and S. Sagayama. Multipitch analysis with Harmonic Nonnegative Matrix Approximation. *Proc. 8th International Conference on Music Information Retrieval*, 2007.

[10] I.S. Dhillon and S. Sra. Generalized nonnegative matrix approximations with Bregman divergences. *Proc. Neural Information Processing Systems (NIPS) Conference*, 2005.

[11] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[12] S. Fine, Y. Singer, and N. Tishby. The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning*, 32(1):41–62, 1998.

[13] K.P. Murphy and M.A. Paskin. Linear Time Inference in Hierarchical HMMs. *Advances in Neural Information Processing Systems 14: Proceedings of the 2002 Conference*, 2002.