HAL open science

Multiple pitch transcription using DBN-based musicological models

Stanislaw A. Raczynski, Emmanuel Vincent, Frédéric Bimbot, Shigeki Sagayama

To cite this version:

Stanislaw A. Raczynski, Emmanuel Vincent, Frédéric Bimbot, Shigeki Sagayama. Multiple pitch transcription using DBN-based musicological models. 2010 Int. Society for Music Information Retrieval Conf. (ISMIR), Aug 2010, Utrecht, Netherlands. pp.363–368. inria-00544064

HAL Id: inria-00544064

https://inria.hal.science/inria-00544064v1

Submitted on 7 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MULTIPLE PITCH TRANSCRIPTION USING DBN-BASED MUSICOLOGICAL MODELS

Stanisław A. Raczyński* **Emmanuel Vincent+** **Frédéric Bimbot+** **Shigeki Sagayama***
raczynski@ emmanuel.vincent@ frederic.bimbot@ sagayama@
* The University of Tokyo, 7–3–1 Hongo, Bunkyo-ku, Tokyo 133-8656, Japan, email: *hil.t.u-tokyo.ac.jp
+ INRIA Rennes, Britagne Atlantique, 35042 Rennes Cedex, France, email: *inria.fr

ABSTRACT

We propose a novel approach to solve the problem of estimating pitches of notes present in an audio signal. We have developed a probabilistically rigorous model that takes into account temporal dependencies between musical notes and between the underlying chords, as well as the instantaneous dependencies between chords, notes and the observed note saliences. We investigated its modeling ability by measuring the cross-entropy with symbolic (MIDI) data and then proceed to observe the model's performance in multiple pitch estimation of audio data.

1. INTRODUCTION

The problem at hand is *musical note detection*, i.e. estimating pitches, onset and offset times, and, if desired, velocities of notes present, often simultaneously, in a recorded audio signal. Typically, this problem is solved by a two-step process [1]. First, pitch candidates are estimated within short time frames and confidence for each is quantified by a salience measure (see, for example, [2–4]). Then the salience is tracked over time in order to identify the musical notes.

The salience can be represented by a *note salience matrix* $\boldsymbol{S}$. Its rows contain estimated power envelopes of notes for different pitches, which typically correspond to frequencies of a diatonic scale, e.g. twelve-tone equal temperament scale. The activity of the underlying musical notes can be expressed by a *note activity matrix* $\boldsymbol{N}$, i.e. a binary matrix of the same dimensions as $\boldsymbol{S}$, elements of which indicate note presence at corresponding times and pitches.

A standard practice is to threshold the estimated saliences to detect notes. This step, although common, is quite problematic: there is no simple way to determine the threshold value and even an optimal value can lead to spurious detections and split notes. Some of the false positives and negatives can be removed by filtering, but it does not solve the problem completely and is not elegant.

Thresholding can in fact be interpreted as a maximum likelihood (ML) estimator of the note activities:

$$\hat{\boldsymbol{N}} = arg\,max_{\boldsymbol{N}}\, P(\boldsymbol{S}|\boldsymbol{N}) \,, \tag{1}$$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

If we assume that the detected saliences $S_{t,k}$ are mutually independent and only depend on whether a corresponding note was active at that moment, we get:

$$\hat{N}_{t,k} = arg\,max_{N_{t,k}} P(S_{t,k}|N_{t,k}) \,, \tag{2}$$

where k is the piano key number and t is the time frame number. If the probability distributions $P(S_{t,k}|N_{t,k}=1)$ and $P(S_{t,k}|N_{t,k}=0)$ have only one crossing point T, this procedure will be equivalent to thresholding with the threshold value equal to T.

Recently, some researchers have used more advanced musicological models in order to overcome the limitations of thresholding. Ryynänen and Klapuri [1] proposed a melody transcription method that uses a Hidden Markov Model (HMM) together with a simple musical key model. Their approach is limited in the sense that it models only a single voice at a time, and so it is not probabilistically rigorous. It also lacks modeling of instantaneous dependencies between estimated pitches. Raphael and Stoddard [5] proposed to use an HMM as a musicological model for harmonic analysis, i.e. estimating the chord progression behind a sequence of notes. Similar HMMs have also been successfully used for harmonic analysis of audio signals (for a recent paper see e.g. [10]). These approaches, however, lack note modeling and the temporal dependencies are only present between chords.

In this paper we have proposed a single, probabilistically rigorous framework based on the Dynamic Bayesian Networks (DBNs). We model both the instantaneous dependencies between notes (harmony) and the temporal dependencies between notes and chords. The notes our found with a maximum a posteriori (MAP) estimator:

$$\hat{\boldsymbol{N}} = arg\,max_{\boldsymbol{N}}\, P(\boldsymbol{S}|\boldsymbol{N})\, P(\boldsymbol{N}) \,. \tag{3}$$

The prior over the notes $P(\boldsymbol{N})$ models the temporal dependencies between the hidden variables (similar to those of an Hidden Markov Model) and includes a hidden layer of variables representing chords.

In our work we used a NMF-based front-end proposed in [6] to obtain note salience matrices with 88 rows that correspond to the full range of a piano: from A0 (27.5 Hz) to C8 (4186 Hz).

The model with its theoretical grounds and its practical aspects is described in section 2. Inference of the hidden notes is discussed in section 3. Experiments involving symbolic and audio data are described in section 4. and the conclusion is given in section 5.

2. THE MODEL

2.1 Structure

DBNs provide us with complete freedom as to what set of probabilistic variables and the relations between them can be, and so it is a perfect tool to solve the above formulated problem. We have chosen a network structure that, compared to thresholding, includes dependencies between hidden variables in neighboring time frames and an additional layer of hidden chords (see Fig. 1).

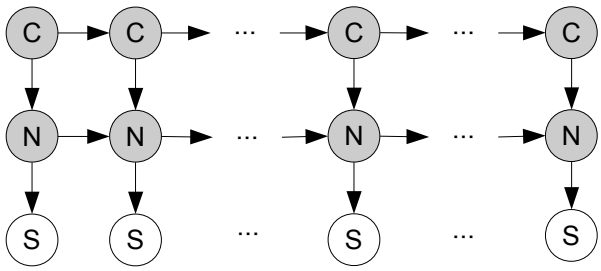

Figure 1: *Structure of the Bayesian network used in experiments*

The network consist of 3 layer of nodes: hidden *chord* layer C_t, hidden *note combination* layer N_t and an observed *note salience* layer S_t.. The prior distribution of notes is therefore given by:

$$P(\boldsymbol{N}) = \sum_{\boldsymbol{C}} P(C_0)\,P(N_0|C_0)\cdot \cdot\prod_{t=2}^{T} P(N_t|N_{t-1},C_t)\,P(C_t|C_{t-1}) \quad . \tag{4}$$

2.2 Chord level probabilities

Fig. 4 shows the chord transition probabilities that has been trained on the available dataset. A simple smoothing was used: each element was increased by 1 after counting the occurrences and before normalizing. Nevertheless, the data sparsity problem is visible (especially for the minor, rarer, chords). To deal with this problem, chord tying was used: each chord transition probability was assumed to be a function of only the interval between roots of the chords

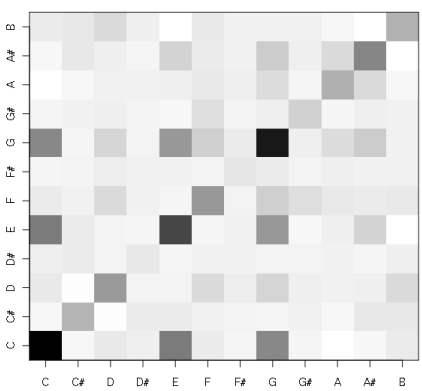

Figure 2: *Covariance matrix for the C-major chord. Note the positive high covariance between the root and the perfect fifth (harmonic interval) and weak covariance between root and minor second (inharmonic interval).*

R_t and R_{t-1}, and their types T_t and T_{t-1}:

$$P(C_t|C_{t-1}) = P(R_t - R_{t-1}, T_t, T_{t-1}) \quad . \tag{5}$$

The motivation behind this approximation is that the probability depends on relative chord positions rather than on the absolute ones. Because the tonal center is not modeled in our approach, it is reasonable to assume the same probability should be given to the transition from C-major chord to F-major (I→IV transition in C-major key) and the from A♭-major to D♭-major (I→IV transition in A♭-major key).

The same motivation led us to use a uniform distribution as the initial chord probability distribution:

$$P(C_0) = \text{const} \quad . \tag{6}$$

2.3 Note level probabilities

Another practical problem concerning the size of the note combination space is the problem of training the model's parameters. The note combination probability $P(N_t|N_{t-1},C_t)$ is a discrete distribution with $|L|^2|C|$ parameters to train, which, even for small values of L is computationally infeasible. To decrease the complexity of the problem, we again tie together some of the parameters: we replace that the note combination probability an approximation, in which it is factorized into the *note transition probability* $P(N_t|N_{t-1})$ and the *note emission probability* $P(N_t|C_t)$:

$$P(N_t|N_{t-1},C_t) \approx \frac{P(N_t|N_{t-1})\,P(N_t|C_t)}{\sum_{N_t} P(N_t|N_{t-1})\,P(N_t|C_t)} \quad . \tag{7}$$

The note probability distribution, as well as the note emission and transition distributions, was normalized over all unique note combinations in the reduced search space. In case of calculating joint likelihood from symbolic data, the sum is performed over all note combinations present in the analyzed data.

2.3.1 Note emission probabilities

There is commonly used multivariate parametric distribution over a discrete set, so to model the note emissions we chose a multivariate Gaussian distribution in the 12-tone chroma space.

$$Cr_{t,l} = \sum_{k \equiv l \,(\mathrm{mod}\,12)} N_{t,k} \tag{8}$$

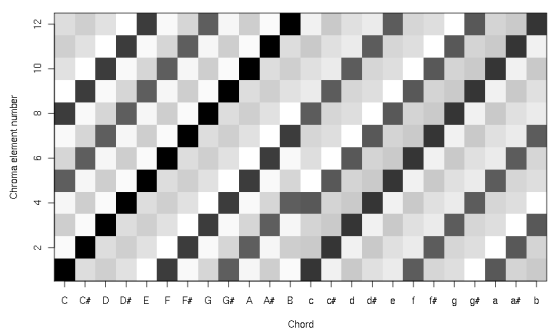

Figure 3: *Mean chroma vectors for different chords.*

$$P(N_t|C_t=m)=\frac{\mathcal{N}_{12}(Cr_t;\mu_m,\Sigma_m)}{\sum_{N_t}\mathcal{N}_{12}(Cr_t;\mu_m,\Sigma_m)} \quad (9)$$

The distribution parameters were estimated on the ground truth data and parameters corresponding to the same chord type were tied together as for the chord level probabilities. To avoid singular covariance matrices due to sparse training data, chroma vectors obtained from reference data were concatenated with a smoothing diagonal matrix $p\boldsymbol{I}$, where p is a control parameter ($p = 2$ was used). The chroma variance is modeled with a full-rank matrix because the pitch classes are not independent (see Fig. 2).

2.3.2 Note transition probabilities

The note transition probability $P(N_t|N_{t-1})$ is responsible for modeling note lengths. There are five basic kinds of changes in the note combination state that can occur in the data (depicted in Fig. 5): no change, insertion of notes, deletion of notes, voice movement (one note changes pitch) and harmony movement or other complex changes (many notes change pitches simultaneously). Because in real life situations note offsets are seldom aligned with other notes' onsets, the last two situations are very rare. In out training data they made up for only 0.2% of note transitions types, while transitions in which all notes stayed the same, if we don't count the insertions and deletions, made up for remaining 99.8% of situations. Motivated by this, in order to simplify the model, we assumed that only the first three kinds are allowed.

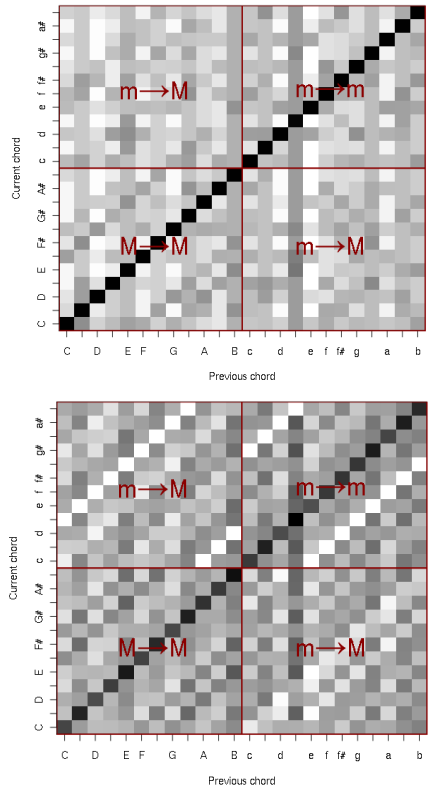

Figure 4: *Chord transition probability matrices: without state tying (top) and with state tying (bottom). Four quarters represent: the major-to-major (M→M), minor-to-major (m→M), major-to-minor (M→m) and minor-to-minor (m→m) transition.*

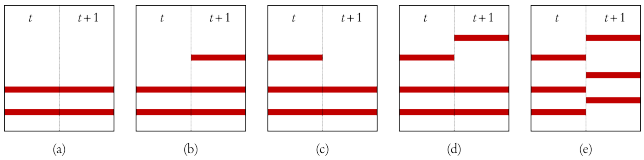

Figure 5: *Five basic note combination transition situations: (a) no change, (b) insertion, (c) deletion, (d) voice movement and (e) harmony movement or other complex changes.*

The note transition probability is therefore further approximated with the following factorization:

$$P(N_t|N_{t-1})\approx\frac{P_{len}(L_t|L_{t-1})P_{mov}(N_t|N_{t-1})}{\sum_{N_t}P_1(L_t|L_{t-1})P_2(N_t|N_{t-1})} \quad (10)$$

$$P_{mov}(N_t,N_{t-1})=\begin{cases}1 & \text{for no pitch movement}\\ 0 & \text{for pitch movement}\end{cases} \quad (11)$$

where L_t is the size of the current note combination (number of active notes). $P_1(L_t,L_{t-1})$ is presented in Fig. 6.

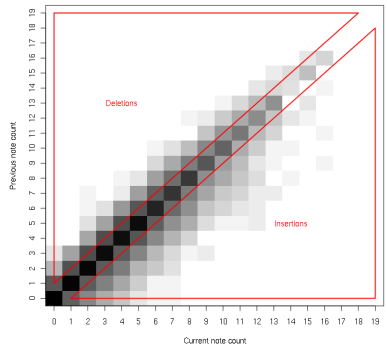

Figure 6: *Distribution of note combination length transitions. The probability matrix is "smeared" more in the area of simultaneous insertions of multiple notes (e.g. at beginnings of chords) are more probable than simultaneous deletions. The z-axis is logarithmic.*

2.3.3 Output probabilities

The observed note saliences are assumed to be mutually independent:

$$P(S_t|N_t)=\prod_{k\in 1}^{88}P(S_{t,k}|N_{t,k})\ . \quad (12)$$

Both obtained by measuring the histograms of the detected salience (see Fig. 8).

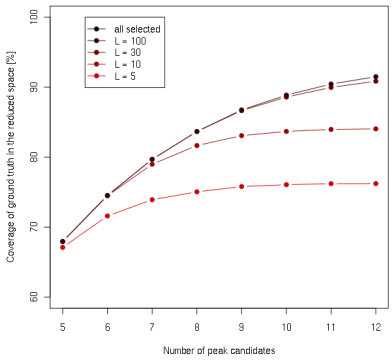

Figure 7: *Note recall for different values of N and L. Data obtained for a = 0.65 and R = 20.*

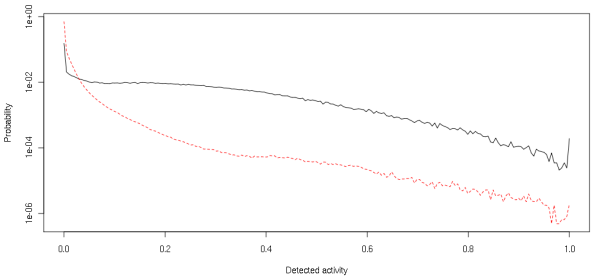

Figure 8: The estimated output probability . The black solid line depicts the distribution of observed note salience if the note was active ($P(S_{t,k}|N_{t,k}=1)$) and the red dashed line the distribution in case the note was inactive ($P(S_{t,k}|N_{t,k}=0)$). The lines cross at about -70 dB.

3. DECODING

3.1 Inference

The problem of multiple frequency estimation becomes a problem of inferring the hidden sequence of note combination states (and, as a side effect, the hidden chord progression). In other words, we need to find the most likely hidden state sequence (***C***, ***N***) given the model and the observed note saliences ***S***:

$$(\boldsymbol{C},\boldsymbol{N})=arg\ max_{(\boldsymbol{C},\boldsymbol{N})}P(\boldsymbol{C},\boldsymbol{N}|\boldsymbol{S})\ . \qquad (13$$

This problem is in fact directly related to the Viterbi decoding in Hidden Markov Models (HMMs).

As in with Viterbi decoding, a dynamic-programming-based algorithm can be used to solve this inference problem for DBNs. We refer to this algorithm as modified Frontier Algorithm. The original Frontier Algorithm was proposed by Murply in [7] to calculate the probability of a given observed sequence (equivalent of the HMM's Forward-Backward algorithm). Murphy noted that it can easily be modified to calculate the most probable sequence of hidden states in any finite-state DBN, i.e. solve our inference problem.

3.2 Reduced solution space

N_t is a variable that holds a list of notes active at a certain time (or, equivalently, a vector of binary note presence indicators). The number of all possible values (states) of N_t is enormous: 3.1×10^{26} if we limit the musical range to that of a 88-key piano. Even if we limit the number of simultaneously active notes to K=10 (if no sustain pedal is used this is the physical limit for a single piano player), it is still computationally infeasible: 5.2×10^{12} if K=10 and 4.2×10^{7} if K=5.

To deal with this problem, we reduce the solution prior to inferring the hidden sequence: for each time frame only the most probable note combinations are considered. To identify the most probable note combinations, first, for each time frame, we select K highest elements, or *note candidates*. Then, a list of all 2^K possible note combinations is created and each such combination is evaluated with a *fitness function*. Finally, the L fittest note combinations are selected and used for further analysis. Additionally, a rest (empty note combination) is always selected.

The fitness function was designed to penalize long note combinations (note combinations containing many active notes) while rewarding better explanation of the observed note saliences S_t:

$$F(N_t)=\frac{\sum_{k\in\{k:N_{t,k}=1\}}S_{t,k}}{\sum_{k=1}^{88}S_{t,k}}|N_t|^{-a} \qquad (14)$$

where N_t is the note combination for the current time frame and a is a control parameter. A similar fitness function was used by Klapuri in [8].

Limiting the solution space poses a threat to the note estimation process: if the real note combination is not selected due to fluctuations in the note salience, the language model will not be able to compensate for that. Therefore, to avoid some of the deletions, the observed note saliences are pre-filtered with a causative moving average (MA) filter:

$$\bar{S}_{t,k}=\sum_{\tau=0}^{R}S_{t-\tau,k} \qquad (15)$$

Additionally, this filtering removes short spurious peaks, e.g. the ones around onset times resulting from the wide-band onset noise. Unfortunately, it also smooths out the onsets.

We have analyzed how much of the ground truth is contained within the reduced solution space, depending on the chosen K, L and a, and on the chosen MA filter order (length), by measuring the note recall. The results are presented in Fig. 7. Optimal values were determined to be R = 20 (400 ms) and a = 0.65 (similar to Klapuri's [8]).

3.3 Fudging

To gain additional control over the behavior of the algorithm, a set of fudge factors was introduced:

$$P(N_t|N_{t-1},C_t)\approx\frac{P(N_t|N_{t-1})^{\alpha}P(N_t|C_t)^{\beta}}{\sum_{N_t}P(N_t|N_{t-1})^{\alpha}P(N_t|C_t)^{\beta}} \qquad (16)$$

Each factor controls the influence of individual probability distribution on the algorithm. The first factor controls mainly the ratio between the self-transition probability of N_t and the cross-transition probability, so smaller values are better for slower pieces and bigger values for higher tempo.

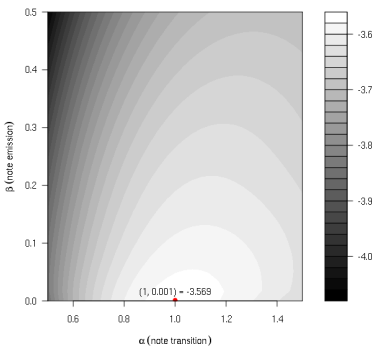

Figure 9: *Optimization of the fudge factors α and β.*

The values of first two factors were then optimized empirically by maximizing the joint likelihood of the hidden note variables $P(\boldsymbol{N})$ (see Fig. 9) and found to be $\alpha = 1.05$ and $\beta = 0.0015$. The fact that the first factor is close to one does not surprise, because.... A very small value of β is due to a very high sparseness of the note emission distribution, i.e. small number of note combinations are assigned significantly higher probability values than the others (which is a result of the curse of dimensionality).

	RWC	Composer	Instrument	Length
1	22	Brahms	2 pianos	2:25
2	23A	Ravel	Piano	1:20
3	23B	〃	Piano	2:45
4	23C	〃	Piano	3:25
5	23E	〃	Piano	4:09
6	24A	Bach	Harpsichord	1:26
7	24B	〃	Harpsichord	1:29
8	24C	〃	Harpsichord	0:52
9	25A	〃	Harpsichord	2:03
10	25B	〃	Harpsichord	2:11
11	25C	〃	Harpsichord	1:31
12	29	Schumann	Piano	2:25
13	30	Chopin	Piano	4:02
14	31	〃	Piano	4:16
15	32	〃	Piano	1:49
16	35A	Satie	Piano	3:49
17	35B	〃	Piano	3:01
18	35C	〃	Piano	2:46
19	40	Massanet	Piano + violin	5:06
			Total:	50:50

Table 1: *RWC pieces used in the experiments.*

4. EXPERIMENTS

4.1 The dataset

The data used in the experiments comes from the widely used RWC database [9]. We have used 19 pieces from the classical portion of the dataset (listed in Table 1).

As a joint effort of the University of Tokyo's Sagayama Laboratory and the Toho Gakuen School of Music (under the supervision of prof. Hitomi Kaneko), the classical pieces of the RWC database were annotated with detailed harmony labels that include: keys and modulations, and chords with their roots, inversions, types and modifications. This data uses abstract musical time (measures and beats), so, additionally, manual labeling of the RWC's audio data was performed.

Unfortunately, the RWC database's MIDI and audio files are not synchronized. What is more, it is not only a matter of linear time transformation, but rather a complex one. Further synchronization with the MIDI was needed for the purpose of training model parameters (note emission probabilities). This was done automatically with dynamic time warping (DTW).

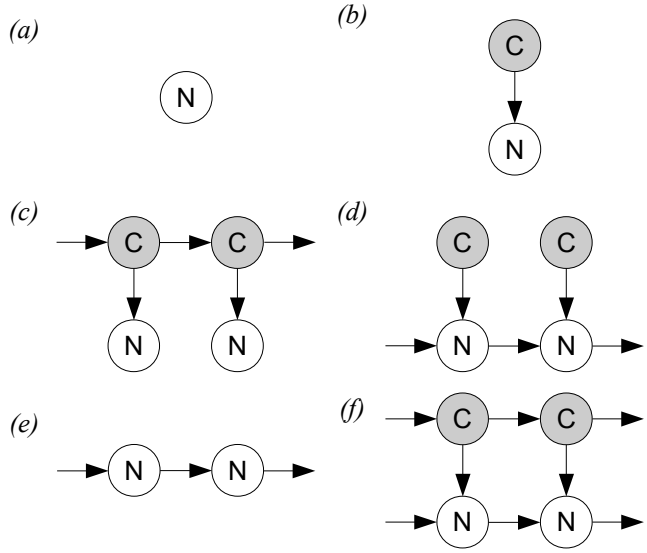

Figure 10: *Six variants of the model used in the evaluation.*

4.2 Symbolic data

A simple procedure to evaluate the proposed approach is to measure how well does our Bayesian network model the symbolic data. This can be assessed by calculating the likelihood of the data given the model $P(\boldsymbol{N})$.

Six variants of the proposed model were evaluated:

(a) Reference uniform model

$$P(\boldsymbol{N}) = \prod_{t=1}^{T} P(N_t) = A^T \tag{17}$$

(b) Harmony model only

$$P(\boldsymbol{N}) = \sum_{\boldsymbol{C}} \prod_{t=1}^{T} P(N_t | C_t) \tag{18}$$

(c) Harmony + chord progression

$$P(\boldsymbol{N}) = \sum_{\boldsymbol{C}} P(C_0) P(N_0 | C_0) \cdot \cdot \prod_{t=2}^{T} P(N_t | C_t) P(C_t | C_{t-1}) \tag{19}$$

(d) Note duration model only

$$P(\boldsymbol{N}) = P(N_0) \prod_{t=2}^{T} P(N_t | N_{t-1}) \tag{20}$$

(e) Duration + harmony

$$P(\boldsymbol{N}) = \sum_{\boldsymbol{C}} P(C_0) P(N_0 | C_0) \prod_{t=2}^{T} P(N_t | N_{t-1}, C_t) \tag{21}$$

(f) Duration + harmony + chord progression

$$P(\boldsymbol{N}) = \sum_{\boldsymbol{C}} P(C_0) P(N_0 | C_0) \cdot \cdot \prod_{t=2}^{T} P(N_t | N_{t-1}, C_t) P(C_t | C_{t-1}) \tag{22}$$

The frontier algorithm was used to evaluate the likelihood for model variants with hidden variables. Each model was evaluated by calculating the cross-entropy, i.e. the normalized log-likelihood of the data $\boldsymbol{N}$ given the model:

$$E(\boldsymbol{N}) = -\frac{1}{T}\log_2 P(\boldsymbol{N}) \tag{23}$$

4.3 Note detection

To evaluate the results of multiple frequency estimation, the F-measure was calculated by comparing the detected notes with the ground truth. A note was considered detected (*true positive*) if its onset was within 100 ms from a true note onset. By measuring the number of true positives, *false positives* (spurious notes) and *false negatives* (undetected notes), the *precision, recall* and F-measure were calculated.

Fig. 12 depicts preliminary note detection results obtained for 7 different models. The first two models were simple thresholding with -40 dB (optimal threshold, determined empirically) and -70 dB (crossing point between the output probability distributions). In the third model the note were detected based on the trained output probability, but only from the reduced solution space. This means that no prior on the notes was present (no language model) and this model was equivalent to the model (a) from subsection 4.2. The last 4 models correspond to the ones described in subsection 4.2, but with the note variables hidden and the note salience layer on the bottom. The proposed model performed not worse than thresholding and generally yielded better recall, but worse precision. The results for RWC-C24A were significantly improved over the thresholding, which can be attributed to the fact that this piece is played on a harpsichord, which has very strong overtones that were mistaken for pitches. The proposed model was able to remove most of these thanks to the prior distribution on the notes.

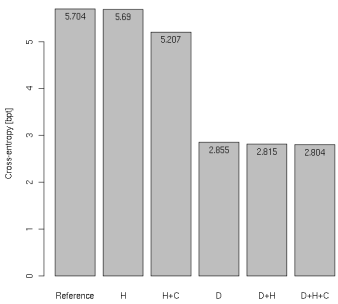

Figure 11: *Average cross-entropy for different variants of the model.*

5. CONCLUSION

We have proposed a uniform probabilistic framework that estimates note onsets and pitches from a salience matrix obtained by a pitch estimation front-end. The model was evaluated on symbolic (MIDI) data and, preliminarily, on audio signals. The results show significant improvement of the model over a reference model with uniformly and independently distributed notes, with the biggest improvement coming from using temporal dependencies.

Compared to the thresholding, the estimation was more robust and yielded higher precision, though the recall was sometimes lower.

In future we plan to focus on improving the accuracy and, therefore, the impact of the simultaneous pitch model $P(\boldsymbol{N}_t|C_t)$. We would also like to explore the possibilities of unsupervised training that would allow us to use a much larger training set, but also investigate the influence of the chosen chord dictionary size.

6. REFERENCES

[1] M. Ryynänen and A. Klapuri: "Modelling of Note Events for Singing Transcription," *Proc. ITRW*, 2004

[2] A. Klapuri: "Multiple fundamental frequency estimation by summing harmonic amplitudes," *in Proc. ISMIR*, pp. 216–221, 2006.

[3] A. Pertusa and J.M. Iñesta: "Multiple fundamental frequency estimation using Gaussian smoothness," *in Proc. ICASSP*, pp. 105–108, 2008.

[4] E. Vincent, N. Bertin, and R. Badeau: "Adaptive Harmonic Spectral Decomposition for Multiple Pitch Estimation," *in IEEE Trans. ASLP*, pp. 528–537, 2010

[5] C. Raphael and J. Stoddard: "Harmonic Analysis with Probabilistic Graphical Models," *in Proc. ISMIR*, pp. 177–181, 2003.

[6] S. Raczyński, N. Ono, S. Sagayama: "Extending Non-negative Matrix Factorization – a discussion in the context of multiple frequency estimation of musical signals," *in Proc. EUSIPCO*, pp.934–938, August 2009.

[7] K. Murphy: "Dynamic bayesian networks: representation, inference and learning," *PhD thesis,* 2002.

[8] D. Kawakami, H. Kaneko, S. Sagayama: "Developing functional harmony labeled data and its statistical analysis," *in Proc. ASJ Spring Meeting*, p. 2, March 2010. *(in japanese)*

[9] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Music Genre Database and Musical Instrument Sound Database," *in Proc. ISMIR*, pp. 229–230, 2003.

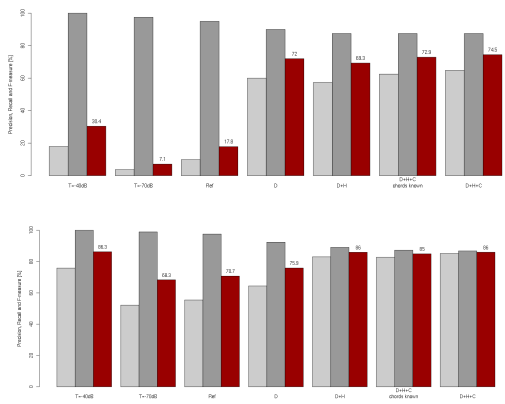

Figure 12: *Results for 7 different models obtained for RWC-C24A (top, α = 2.3, L = 12, N = 70) and RWC-C22 (bottom, α = 1.3, L = 12, N = 75).*

[10] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, S. Sagayama, "HMM-based Approach for Automatic Chord Detection Using Refined Acoustic Features," *in Proc. ICASSP*, March 2010.