AN AUGMENTED LAGRANGIAN METHOD FOR PIANO TRANSCRIPTION USING EQUAL LOUDNESS THRESHOLDING AND LSTM-BASED DECODING

Sebastian Ewert, Mark B. Sandler

Machine Listening Lab (MLLAB) and Centre for Digital Music (C4DM)
School of Electronic Engineering and Computer Science
Queen Mary University of London
United Kingdom

ABSTRACT

A central goal in automatic music transcription is to detect individual note events in music recordings. An important variant is instrument-dependent music transcription where methods can use calibration data for the instruments in use. However, despite the additional information, results rarely exceed an f-measure of 80%. As a potential explanation, the transcription problem can be shown to be badly conditioned and thus relies on appropriate regularization. A recently proposed method employs a mixture of simple, convex regularizers (to stabilize the parameter estimation process) and more complex terms (to encourage more meaningful structure). In this paper, we present two extensions to this method. First, we integrate a computational loudness model to better differentiate real from spurious note detections. Second, we employ (Bidirectional) Long Short Term Memory networks to re-weight the likelihood of detected note constellations. Despite their simplicity, our two extensions lead to a drop of about 35% in note error rate compared to the state-of-the-art.

Index Terms— Proximal Methods, Alternating Directions Method of Multipliers, Structured Sparse Coding, Instrument-dependent Transcription.

1. INTRODUCTION

Automatic music transcription (AMT) is often considered to be a key technology in music processing as it provides a link between the acoustic domain (in the form of audio recordings) and the symbolic music domain (capturing note events and higher level musical concepts) [1]. A central component in an AMT system is the detection of individual note events in an audio recording of a piece of music. However, despite ongoing research since the 1970s [2], the AMT problem remains unsolved in its most general form [1], i.e. for an unknown number of instruments of unknown type playing jointly under unknown acoustic conditions. In particular, a major challenge is that in music note events are usually highly correlated both in time and frequency – from a modelling point of view this often results in highly ill-conditioned systems of (non-)linear equations [3].

Several families of AMT methods have been proposed, each building on specific strategies to approach the AMT problem, see [1, 3, 4] for an overview. Currently, most state of the art methods either employ neural networks [5, 6] (typically using discriminative modelling) or factorization methods [7] (i.e. inference methods in generative models). Using the piano transcription task as an example, current methods typically yield f-measures (for correctly detected notes) of around 50-70%, leaving considerable room for improvement.

A typical approach to increase the transcription accuracy is to include recordings of the instrument to be transcribed in the training material – this is valid in a variety of scenarios where a calibration phase is possible (e.g. studio or home recordings). We will refer to this problem scenario as *instrument-dependent music transcription*. However, despite the availability of additional information, the f-measure for many methods improves only slightly to 60-80% – this range holds for both discriminative methods [5, 6] and generative models [8,9]. For example, in [5] Kelz et al. describe the current state-of-the-art based on neural networks (instrument-dependent training) – the final proposed method achieves an f-measure of $\approx$80%, which is achieved employing an extensive hyper parameter tuning process.

A first idea to improve ill-conditioned problems is to lower the 'noise', which means to keep the patterns used for identification as close as possible to the observations. In particular, for instruments such as the piano, a note is not a stationary sound but rather evolves in typical formations over time. Most factorization based methods, however, do not take this temporal progression into account and rather employ pure spectral templates. The idea in [10] is thus to model this note progression employing a graphical model that controls the temporal position in 88 spectro-temporal patterns, each associated with one piano key. The model is conceptually similar to Non-negative Matrix Deconvolution (NMD) [11] but employs, in contrast to NMD, patterns of variable length. There is also a close connection to non-negative factorial HMMs (NFHMM) [12,13] – the main difference in [10] being in the use of a specialized parameter estimation process to enable the use of 88 parallel Markov processes.

Overall, the system presented in [10] models the piano sound production process quite closely. Yet, this was not reflected in the evaluation results, with f-measure values around 80% on a standard dataset (MAPS [14]). A detailed analysis conducted in the context of [3] revealed that the signal model can be used to yield a higher transcription accuracy. However, for numerical reasons, the underlying parameter estimation process used in [10] was biased towards specific local minima of an objective function that are likely to cause misdetections. The design goal in [3] was thus to use a signal model similar to [10] but to replace the entire parameter estimation process. The resulting method consecutively switches from simple, convex regularizers (that stabilize the initial parameter estimation process) to more complex terms (to encourage a more meaningful structure as expressed by a graphical model). As a result, focusing only on the numerical properties of the parameter estimation process, the methods yields f-measure values of 95%.

While this is a step forward, the performance is still not high enough for all relevant applications – intuitively, still 5 in 100 notes

This work was funded by EPSRC grant EP/L019981/1.

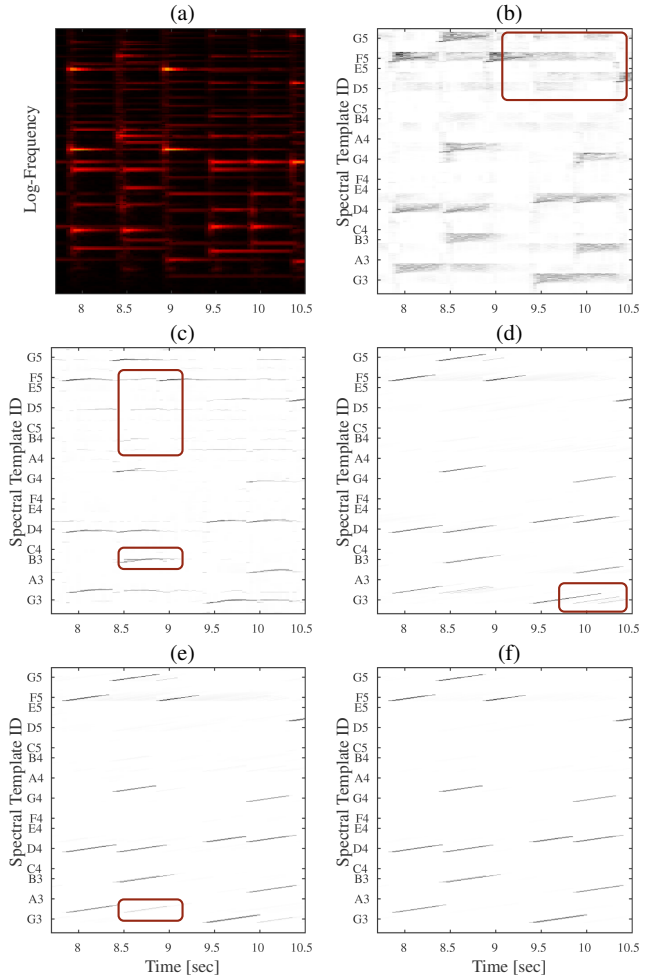

Figure 1: Illustration of the effect of using different combinations of regularizers. **(a)** Log-frequency spectrogram of a recording of Chopin's Nocturne No.2 (Op. 9). **(b)-(g)** Activity tensor estimated using different combinations of regularizers, see text for details.

are not correctly detected. The contribution in this paper is to investigate additional strategies for further increasing this accuracy. Such a high accuracy is particularly important in education systems that are used to give students feedback on their mistakes [15, 16]. In this context, a manual analysis revealed two important sources of errors: notes played with a low intensity and additive noise (moving chairs, coughing). With respect to the first problem, the system in [3] employs a single threshold for all pitches to differentiate real from spurious notes. Using only a single value, however, is problematic as loudness perception depends on the frequency. Therefore, a first simple extension is to make this threshold pitch-dependent. Here, as we will see, using simple schemes such as equal loudness contours that are based on sinusoidal tones did not give an advantage – instead we incorporated a method based on the Glasberg-Moore model for complex, non-stationary sounds [17].

To deal with additive noise, we can exploit that additive sounds as described above typically lead to activations that are harmonically unrelated to the music. That means we need a measure for how likely a certain constellation of notes is. This could be implemented using an HMM – however, since we are modeling constellations of notes, the corresponding state-space would at least have a size of 2^{88} (one state for each combination of active notes), which is practically infeasible. However, such large, complex joint distributions have recently been successfully approximated using neural networks [18] [6]. Therefore, as a second extension, we investigate here combining the method proposed in [3], which is adaptable to new acoustical conditions with minimal effort, with long short term memory (LSTM) neural networks for decoding, which essentially provide a simple musical language model on top.

The remainder is organized as follows. In Section 2, we describe our proposed extensions in more detail and report in Section 3 on our evaluation results. We conclude in Section 4 with a prospect on future work.

2. PROPOSED METHOD

2.1. Signal Model

Before we discuss our proposed extensions, we begin with a short summary of the model as presented in [3] and refer there for many of the details. The core of our signal model corresponds to a tensor product modeling a time-frequency representation $V \in \mathbb{R}_{\geq 0}^{M,N}$ of a recording to be transcribed:

$$V_{m,n} \approx (PA)_{m,n} := \sum_k \sum_\ell P_{m,\ell,k} \cdot A_{k,\ell,n}. \tag{1}$$

The *pattern dictionary tensor* $P \in \mathbb{R}_{\geq 0}^{M \times L \times K}$ contains K spectro-temporal patterns, each consisting of L frames. Here, $K = 88$ corresponds to the number of keys on a piano. That means each column $P_{:,\ell,k} \in \mathbb{R}_{\geq 0}^{M}$ for fixed ℓ and k contains a single *spectral template*; here we used the slicing notation $:$ to refer to all elements in an index dimension. Each pattern $P_{:,:,k}$ corresponds to a recording of a single note. The *activity tensor* $A \in \mathbb{R}_{\geq 0}^{K \times L \times N}$ encodes the activity of each template in each frame.

This basic signal model is relatively free and thus requires strong regularization. The following objective function employs several of the regularizers as proposed in [3]:

$$\begin{aligned} f(A) := & \sum_{m,n} d(V_{m,n}, (PA)_{m,n}) && (2)\\ & + \chi_{\mathbb{R}_{\geq 0}^{K \times L \times N}}(A) && (3)\\ & + \lambda_1 \left\| A \right\|_1 && (4)\\ & + \lambda_2 \left\| \Delta_D[A] \right\|_1 && (5)\\ & + \chi_{\mathcal{M}}(A) && (6)\\ & + \chi_{\mathcal{T}}(A) && (7) \end{aligned}$$

We illustrate the idea behind each term in Fig. 1. In Fig. 1a we see a time-frequency representation of the recording to be transcribed. Fig. 1b-f show activity tensors A obtained using different subsets of the terms (2)–(7). For illustrative purposes, each A is flattened out by placing the slices $A_{k,:,:}$ vertically on top of each other. With $d(a,b) := a \cdot \log\left(\frac{a}{b}\right) - a + b$ for $a, b > 0$ term (2) is a data fidelity term using the generalized Kullback-Leibler divergence. Term (3) encourages non-negativity of A, where χ_S is the characteristic function for some set S with $\chi_S(x) = 0$ if $x \in S$ and $\chi_S(x) = \infty$ otherwise. Fig. 1b shows the result of using only terms (2)–(3): As discussed in [3], the corresponding A is blurred and noisy – due to the structure of P we would expect diagonal lines that start at the position of note onsets. A transcription based on such a representation is likely to contain a larger number of errors.

To obtain more meaningful activations, term (4) introduces a sparsity inducing ℓ_1 regularizer. As shown in Fig. 1c, the results indeed clear up. However, instead of diagonal lines, we rather see horizontal lines. This is caused by using single note patterns in P whose individual templates are not normalized, i.e. we preserve the characteristic energy decay in the pattern. This causes here, however, that only the energy-rich, first templates in a pattern are activated. This activation of 'wrong' templates causes residual energy and thus spurious activity. As a remedy, term (5) discourages changes along diagonals using an anisotropic variant of the total variation operator. Here, Δ_D is essentially a simple high-pass filter along the diagonals as introduced in [3]. As shown in [3], terms (2)–(5) are jointly convex in A – when used to obtain a first initialization for A the convexity improves the robustness of the method considerably.

However, while convexity improves numerical stability, it often limits the expressiveness of terms. To improve upon remaining problems, additional non-convex terms are added. One remaining problem can be see in the G3 activations around 10 seconds (Fig. 1d): The G3 is activated twice, which is physically impossible and can lead to estimation errors. As a countermeasure, term (6) uses the characteristic function with a very specific set $\mathcal{M}$. This set contains only tensors A whose activations encode states in a specific graphical model. The model essentially encodes that a note has a minimum length and how it can progress in time. Including term (6) resolves the concurrency issues (Fig. 1e). Due to space restrictions, we refer to [3] for details.

A final problem is visible in Fig. 1e: a weak, incorrect activation of G#3 around 8.5 seconds (octave error). If the note energy is distributed across several weak activations, the correct activation can fall below the detection threshold. For this reason, term (7) specifies with $\mathcal{T} := \left([a_{\text{min}}, \infty) \cup \{0\}\right)^{K \times L \times N}$ that activations have to be zero or greater than a_{min}. This way, low intensity energy is 'pulled' into the main activation which in extreme cases pushes the activation above the detection boundary. In this context, see also [19] for a connection between hard thresholding and hard ℓ_0 sparsity.

2.2. Parameter Estimation using the Augmented Lagrangian

To obtain a meaningful A, we need to find a minimizing argument to our objective function f. However, such a function is difficult or even impossible to minimize with classical gradient or Newton-type optimization methods. It contains highly non differentiable terms, terms that yield infinity as value and strongly non-convex terms. In this context, Augmented Lagrangian methods have been found to be of high interest. In particular, the variant *Alternating Directions Method of Multipliers (ADMM)* [20] provides a scheme to split up the objective function, minimizing the terms individually and still provides convergence guarantees for the entire objective. As a result it is not only useful for complex objective functions as in our case but also in big data scenarios, as ADMM's splitting and merging operations fit perfectly into distributed computing schemes like Map-Reduce. Due to space constraints we refer to [3] for more details on ADMM and minimizing the objective function f.

2.3. Thresholding based on Glasberg-Moore Model

Before we describe a first extension to the method introduced in [3], we identify a potential weakness. In particular, the hard thresholding introduced by term (7) was found to be particularly useful to improve the detection of low intensity notes, i.e. those close to the decision boundary. In [3], the corresponding threshold a_{min} was derived from user input as this threshold depends on the recording level. More precisely, the user is asked to provide an example of a note having the lowest intensity to be expected in a recording session (during the evaluation this was one value for the entire dataset and not specific to recordings). The system processes this low-intensity recording and sets a_{min} using the computed activation values.

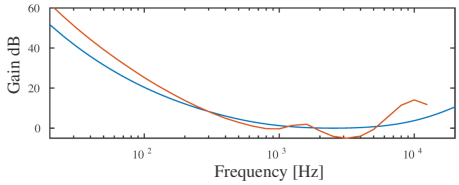

Figure 2: Equal loudness curves: Inverted A-weighting (Blue) and Fletcher et al. for 35 Phon (Orange).

To keep the user effort minimal, the system employs only a single low-intensity note. While including this type of thresholding in the optimization procedure led to measurable improvements, there is a problem: the perception of loudness is frequency dependent. That means, if the low-intensity note has a high pitch, its energy is likely to be different from a note having the same perceived loudness but with lower pitch. In other words, an energy-based threshold chosen based on one pitch is likely to be incorrect for another. Therefore, in a first extension we make the threshold a_{min} pitch dependent, without increasing the user effort.

A first idea to implement this change is to set the threshold based on equal loudness curves: Fig. 2 shows the widely used inverted A-weighting and Fletcher curves [21]. More precisely, we start by normalizing all single note recordings (that are used to create the pattern tensor P) to have the same root-mean-square (RMS) energy. Then, for each note, we calculate the difference in dB between the equal loudness value for that note and the one for the low-intensity note – for this lookup, we use the fundamental frequency associated with each note. Using this difference, we can derive an individual threshold for each pitch (which then hopefully corresponds to the energy level for a note of that pitch having the same loudness as the low-intensity note). Unfortunately, this procedure led to virtually no improvement in the results (f-measure improved by 0.2). There was no difference between the A-weighting and the Fletcher curve, which is not surprising given their similarity, compare Fig. 2.

A possible reason could be that there simply are not many low-intensity notes in our evaluation dataset and thus such a measure cannot have a stronger effect. Alternatively, the new thresholds might simply not be meaningful enough as both curves were based on listening tests involving stationary, sinusoidal sounds, while piano notes are harmonic and non-stationary. To test this hypothesis we employed a more complex loudness model as proposed by Glasberg and Moore [17], which was designed to provide a better fit to complex, non-stationary sounds. To this end, we derived for each RMS-normalized note a scaling factor such that the scaled note has the same perceived loudness as a reference note (C4 in our case) – loudness was measured as the local maximum over the entire note duration (Note: All normalized note recordings were pre-scaled to an assumed playback level of 30 db-SPL for the measurement). We can then use these scalers to convert the threshold obtained from the provided low-intensity note recording to all remaining piano keys. We will report on the results for this variant in Section 3.

2.4. LSTM-based Decoding

A second occasional problem we observed stems from non-musical interferences, including mechanical sounds from the instrument or

breathing sounds. These sometimes led to spurious activations, in particular, for pitched interferences. Most of these activations, however, are not strongly correlated with the music and typically occur as short, out-of-key activations. Graphical models such as an HMM could be used in this context to smooth over such unusual, musically often irrelevant activations (similar to a language model in speech recognition). However, even a simple frame-wise model would need to span a space consisting of 2^{88} states to model each combination of notes. While there exist sophisticated pruning techniques for such cases, they tend to be quite complex and often involve considerable trade-offs with respect to approximation quality and runtime performance. To approximate such complex joint probabilities, there has recently been considerable success using neural networks [18] [22] [6] (in the context of symbolic music representations). Following similar ideas, we have trained long short term memory (LSTM) based recurrent networks to decode the position of onsets in each key, given the activations obtained as above. While this could have been done with various other architectures as well (e.g. convolutional networks with time context) we chose LSTM networks as our input representation is relatively low-dimensional (convnets are often used to get around the difficulty of training RNNs with high dimensional inputs) and LSTM networks have (theoretically) the model capacity to represent very long temporal dependencies [18]. The latter is achieved by LSTM networks as they provide a more stable gradient flow in the backpropagation-through-time algorithm, which is essentially implemented through gated shortcut connections [22].

As a first step, we divided the activation values associated with each pitch by the corresponding pitch-dependent threshold. This way, all activations are normalized to a certain range of values and comparable statistics, which helps with the training process. In other words, most instrument and recording specific properties are eliminated from the input and the network can focus on musical aspects, which can be learned independently and do not need to be adapted to new acoustic conditions. The input for frame n consists of $A_{:,1,n} \in \mathbb{R}^{88}$, i.e. the activations for the onset part of each note pattern in P. The same representation was used in [3] for the final onset detection. We used LSTM networks in two different configurations. To take the entire recording into account the first configuration uses a bidirectional LSTM network [23]. In such a BLSTM network, one LSTM network operates on the original input sequence and the other one on the reverse sequence. The two networks are then trained jointly. For very long sequences, however, training BLSTM networks typically involves splitting the input sequence into chunks, which is necessary to deal with the limitations in computational resources but leads to questions whether the reversed LSTM network is actually necessary. Therefore, we trained in a second configuration a uni-directional LSTM. In this configuration, we simply delay the detection of notes by 400ms to allow the network to peak a little into the future.

3. EXPERIMENTS

To evaluate our proposed extensions, we employed the ENSTDkCl subset of the MAPS collection [14], which provides audio recordings of a Yamaha Disklavier and corresponding MIDI-based annotations. To evaluate a method, we employ precision (P), recall (R), and F-measure (F) as used in the MIREX evaluation campaigns. A detected note is considered correct if there is a note in the corresponding ground truth having the same MIDI pitch, with an onset position up to N ms apart from the detected note. As discussed in [3], we set N=100 to account for some temporal jitter in the ground truth annotations. Every ground truth note can validate up to one detected note. For the LSTM networks, we used two layers with 100 units each and a final dense layer containing 88 units with sigmoid activations. For the training we employed a subset of MAPS that was generated using a variety of software synthesizers, i.e. there was no overlap with the test dataset regarding the acoustic conditions. Besides a dropout of 0.5, which we apply only to the non-recurrent connections following [24], we employ no other unusual strategies [25][1]. The results for several methods, including our baseline [3] and our proposed extensions are given in Table 1.

Method	P	R	F
O'Hanlon et al. [8]	89	77	82
Cogliati et al. [9]	80	84	81
Ewert et al. 2015 [10]	76	83	79
Ewert et al. 2016 (Baseline) [3]	96	93	95
Extension 1	96	95	96
Extension 1+2 (LSTM)	97	96	97
Extension 1+2 (BLSTM)	97	96	97

Table 1: Precision, Recall and F-Measure in percent for various methods using the MAPS dataset.

The first extension led to an improvement of 0.7 in f-measure compared to the baseline. Given that there are typically not many notes close to the decision boundary, this might be what can be expected from such a simple extension. We were surprised, however, that this value seemed to be consistently higher than just using equal loudness contours. The use of a more complex loudness model made some difference here. The LSTM-based decoders improved the results by another 1.0 in f-measure to an overall improvement of 1.7. Again, a small improvement but given that the MAPS dataset is very clean and does not contain breathing or similar interferences, we would expect even bigger improvements in actual live recordings. We did not measure a difference in performance between the LSTM and the BLSTM networks, which might indicate that the delayed output solution used in the LSTM network is enough in this specific application scenario. Overall, the extensions increased the f-measure marginally from 95% to 97%. While this is an incremental rather than a major step forwards, it means that the expected number of wrong notes in a 100 has gone down from five to three. For feedback systems in education, this can make quite a difference.

4. CONCLUSIONS

We presented two extensions improving the accuracy of a state-of-the-art music transcription system. The first extension is based on specifying note-detection boundaries using the Glasberg-Moore loudness model for complex non-stationary sounds. The second extension employs an LSTM network to post-process the output of a dictionary-based method using variable-length spectro-temporal patterns. This way, the capacity to quickly adapt to new acoustic conditions (acoustical model) is combined with a decoder that can focus on music specific aspects such as the likelihood of specific note constellations. The f-measure increased from 95% to 97%, corresponding to a drop from five to three incorrect note detections per 100 notes.

[1] We employ Glorot weight initialization [26] and label smoothing [25]. To normalize the input variance [25], we measure the variance across both input samples and input dimensions to become invariant against pitch dependent velocity biases in the dataset. Loss is an element-wise cross-entropy. Optimizer is Adam using an initial stepsize set to 1/10 of the default [27].

5. REFERENCES

[1] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Breaking the glass ceiling," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 379–384.

[2] J. A. Moorer, "On the transcription of musical sound by computer," *Computer Music Journal*, vol. 1, no. 4, pp. 32–38, 1977.

[3] S. Ewert and M. Sandler, "Piano transcription in the studio using an extensible alternating directions framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 1983–1997, Nov 2016.

[4] A. P. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. New York: Springer, 2006.

[5] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 475–481.

[6] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, 2016.

[7] E. Benetos, S. Ewert, and T. Weyde, "Automatic transcription of pitched and unpitched sounds from polyphonic music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3107–3111.

[8] K. O'Hanlon, H. Nagano, N. Keriven, and M. D. Plumbley, "Non-negative group sparsity with subspace note modelling for polyphonic transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 530–542, March 2016.

[9] A. Cogliati, Z. Duan, and B. Wohlberg, "Piano transcription with convolutional sparse lateral inhibition," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 392–396, 2017.

[10] S. Ewert, M. D. Plumbley, and M. Sandler, "A dynamic programming variant of non-negative matrix deconvolution for the transcription of struck string instruments," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 569–573.

[11] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation (ICA)*, Grenada, Spain, 2004, pp. 494–499.

[12] G. Mysore and M. Sahani, "Variational inference in non-negative factorial hidden Markov models for efficient audio source separation," in *Proceedings of the International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, 2012, pp. 1887–1894.

[13] E. Benetos, S. Cherla, and T. Weyde, "An efficient shift-invariant model for polyphonic music transcription," in *Proceedings of the International Workshop on Machine Learning and Music*, 2013.

[14] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[15] E. Benetos, A. Klapuri, and S. Dixon, "Score-informed transcription for automatic piano tutoring," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2012, pp. 2153–2157.

[16] S. Ewert, S. Wang, M. Müller, and M. Sandler, "Score-informed identification of missing and extra notes in piano recordings," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York, USA, 2016, pp. 30–36.

[17] B. R. Glasberg and B. C. Moore, "A model of loudness applicable to time-varying sounds," *Journal of the Audio Engineering Society*, vol. 50, no. 5, pp. 331–342, 2002.

[18] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2012, pp. 1159–1166.

[19] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 298–309, 2010.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[21] H. Fastl and E. Zwicker, *Psychoacoustics, Facts and Models (3rd Edition)*. Springer, 2007.

[22] D. Eck and J. Schmidhuber, "Finding temporal structure in music: Blues improvisation with LSTM recurrent networks," in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, 2002, pp. 747–756.

[23] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[24] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[25] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade (2nd Ed)*. Springer, 2012, pp. 437–478.

[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 9, 2010, pp. 249–256.

[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference for Learning Representations (arXiv preprint arXiv:1412.6980)*, 2015.