MULTI-TEMPLATE SHIFT-VARIANT NON-NEGATIVE MATRIX DECONVOLUTION FOR SEMI-AUTOMATIC MUSIC TRANSCRIPTION

Holger Kirchhoff, Simon Dixon, Anssi Klapuri
Queen Mary University of London, Centre for Digital Music
{holger.kirchhoff, simon.dixon, anssi.klapuri}@eecs.qmul.ac.uk

ABSTRACT

For the task of semi-automatic music transcription, we extended our framework for shift-variant non-negative matrix deconvolution (svNMD) to work with multiple templates per instrument and pitch. A k-means clustering based learning algorithm is proposed that infers the templates from the data based on the provided user information. We experimentally explored the maximum achievable transcription accuracy of the algorithm and evaluated the prospective performance in a realistic setting. The results showed a clear superiority of the Itakura-Saito divergence over the Kullback-Leibler divergence and a consistent improvement of the maximum achievable accuracy when each pitch is represented by more than one spectral template.

1. INTRODUCTION

Automatic music transcription describes the process of transforming a recording of a piece of music into a score or an intermediate score-like representation. It has been an active area of research over the last decades and a multitude of approaches has been proposed. An overview of the main computational techniques for music transcription can be found in [1]. Despite this long research history, the accuracy of fully automatic music transcription systems is still considerably below the accuracy achieved by trained musicians.

As a step towards a more accurate transcription system, we address the task of *user-assisted* or *semi-automatic music transcription*. These terms refer to systems in which the user provides a certain amount of information about the recording under analysis which can then be used to guide the transcription process. In this paper, we assume that the user labels a certain number of notes for each instrument, which is then used to build instrument models that are tailored to the specific instruments in the mixture. In a practical application, the user could either be presented with a magnitude spectrogram and be asked to graphically mark a few fundamental frequency trajectories, or — if a more musical approach is desired — with the result of a fully-automatic transcription system for which he is asked to assign some of the detected notes to the instruments.

We address this task by means of a non-negative matrix deconvolution framework. Since the introduction of non-negative matrix factorisation (NMF) [2] which was first applied to music analysis by Smaragdis and Brown [3], a number of modifications to this algorithm have been proposed. In this work, we build on our shift-variant non-negative matrix deconvolution (svNMD) framework [4] which is itself a modification of Schmidt and Mørup's NMF2D model [5]. In the svNMD framework, a single spectral template for each pitch of each instrument is estimated which is then used to detect fundamental frequencies in the constant-Q magnitude spectrogram of the recording. Here, we extend the model to work with multiple templates per pitch. The motivation for having multiple templates per pitch is given by the fact that the spectral shape of a particular note can vary based on dynamics or playing style and to model a time-varying spectral envelope of a note.

Other related work can be found in NMF-based approaches to score-informed source separation, where mid-level score representations are used to infer models for the source instruments. Hennequin et al. [6] modify the NMF model to work with parametric spectral templates. The model allows templates to be shifted in frequency while preserving the overtone amplitudes. The parameters are learned by initialising the NMF gain matrix and successively applying update functions for the template parameters and the gains. In [7], Ganseman et al. use a synthesised and time-aligned score as priors for the PLCA system proposed in [8]. In addition to note information, this approach requires knowledge about the timbre of each source in order to facilitate a fast convergence.

The remainder of this paper is organised as follows: In the following section we present our extension to the svNMD framework that works with multiple templates per pitch (Sect. 2.1) and illustrate the algorithm for learning these templates (Sect. 2.2). In Sect. 3 we evaluate the proposed algorithm in two different experiments and discuss the results. Conclusions are finally drawn in Sect. 4.

2. MULTIPLE-TEMPLATE SHIFT-VARIANT NON-NEGATIVE MATRIX DECONVOLUTION

In this section we present our non-negative matrix deconvolution framework which decomposes a constant-Q spectrogram into a structured dictionary of instrument templates and corresponding gain values (see Sect. 2.1). The framework represents each pitch of each instrument by a predefined number of spectral templates. Furthermore, in Sect. 2.2 we describe a procedure that allows us to extract multiple templates for each note previously labelled by the user. This procedure is applicable to polyphonic material where partials might overlap.

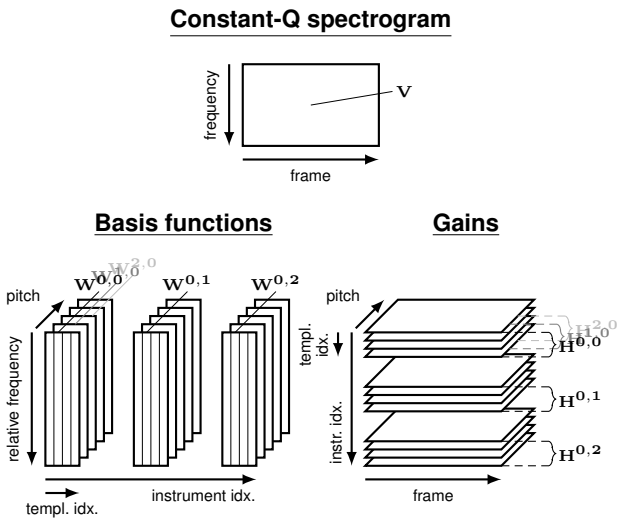


Figure 1: svNMD framework with multiple templates per instrument and pitch.

2.1 Framework

The proposed non-negative matrix deconvolution framework decomposes a constant-Q spectrogram into 4-dimensional structures for the basis functions and the gains, respectively. Figure 1 illustrates the framework graphically. Each instrument in the mixture under analysis is represented by a 3-dimensional structure (tensor) that contains a fixed number of basis functions for each pitch. The pitch resolution is determined by the frequency resolution of the constant-Q spectrogram under analysis and the number of templates per pitch can be chosen arbitrarily. Likewise, for each instrument a 3-dimensional structure contains the corresponding gains for the spectral templates. Each layer displayed on the right-hand side of Fig. 1 contains the gain trajectories at a fixed template index over time. In order to arrive at a single pianoroll-like representation for each instrument, the gains of the layers can be summed up vertically.

In mathematical terms, we denote the constant-Q magnitude spectrogram by $\mathbf{V} \in \mathcal{R}_+^{N \times M}$, where N is the number of frequency bins and M the number of frames. The matrix $\mathbf{W}^{\phi,i} \in \mathcal{R}_+^{N \times T}$ contains in its columns the spectral templates of instrument i at pitch ϕ (see Fig.1). T denotes the specified number of spectral templates. All templates have their first partial at the first row index of $\mathbf{W}^{\phi,i}$ and likewise all other partials appear each roughly at their corresponding row index due to the use of the constant-Q spectrogram. $\mathbf{H}^{\phi,i} \in \mathcal{R}_+^{T \times M}$ on the other hand denotes the matrix that contains the corresponding gains for the templates of instrument i at pitch ϕ over time. Note that in Fig. 1, this matrix corresponds to a slice through one of the banks of layers, as shown in the figure.

Given these matrices we approximate our original spectrogram $\mathbf{V}$ by

$$\mathbf{V} \approx \mathbf{\Lambda} = \sum_{i=0}^{I-1} \sum_{\phi=0}^{\Phi-1} \overset{\phi\downarrow}{\mathbf{W}}^{\phi,i} \mathbf{H}^{\phi,i}, \tag{1}$$

where $\mathbf{\Lambda} \in \mathcal{R}_+^{N \times M}$ has the same dimensions as $\mathbf{V}$. Here, I denotes the number of instruments in the mixture and Φ the number of pitches. Φ and N do not necessarily need to be the same, in our case, however, they are. The operator $\phi\downarrow$ denotes a downward shift of the matrix elements by ϕ rows while the upper ϕ rows are filled with zeros. This mixture model shifts each spectral template to the correct frequency position and scales them by the corresponding gains at each frame.

Update equations were derived for both $\mathbf{W}^{\phi,i}$ and $\mathbf{H}^{\phi,i}$ by computing the gradient of the β-divergence between $\mathbf{V}$ and $\mathbf{\Lambda}$. The β-divergence is given by

$$C_\beta = \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{[\mathbf{V}]_{n,m}^{\beta}}{\beta(\beta-1)} + \frac{[\mathbf{\Lambda}]_{n,m}^{\beta}}{\beta} - \frac{[\mathbf{V}]_{n,m}\,[\mathbf{\Lambda}]_{n,m}^{\beta-1}}{\beta-1}, \tag{2}$$

for $\beta \in \mathcal{R} \backslash \{0,1\}$ and

$$C_0 = \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{[\mathbf{V}]_{n,m}}{[\mathbf{\Lambda}]_{n,m}} - \log\left(\frac{[\mathbf{V}]_{n,m}}{[\mathbf{\Lambda}]_{n,m}}\right) - 1 \tag{3}$$

$$C_1 = \sum_{n=1}^{N} \sum_{m=1}^{M} [\mathbf{V}]_{n,m} \log\left(\frac{[\mathbf{V}]_{n,m}}{[\mathbf{\Lambda}]_{n,m}}\right) + [\mathbf{\Lambda}]_{n,m} - [\mathbf{V}]_{n,m}\,. \tag{4}$$

The update equations are given by

$$\mathbf{W}^{\phi,i} \leftarrow \mathbf{W}^{\phi,i} \bullet \frac{\left(\overset{\phi\uparrow}{\mathbf{V}} \bullet \overset{\phi\uparrow}{\mathbf{\Lambda}}^{\beta-2}\right) \left[\mathbf{H}^{\phi,i}\right]^T}{\overset{\phi\uparrow}{(\mathbf{\Lambda}^{\beta-1})} \left[\mathbf{H}^{\phi,i}\right]^T} \tag{5}$$

$$\mathbf{H}^{\phi,i} \leftarrow \mathbf{H}^{\phi,i} \bullet \frac{\left[\overset{\phi\downarrow}{\mathbf{W}}^{\phi,i}\right]^T \left(\mathbf{V} \bullet \mathbf{\Lambda}^{\beta-2}\right)}{\left[\overset{\phi\downarrow}{\mathbf{W}}^{\phi,i}\right]^T \mathbf{\Lambda}^{\beta-1}} \tag{6}$$

In these equations, $\bullet$ denotes an elementwise multiplication and all divisions and power operations are likewise carried out per element. We can obtain the well-known least squares (LS), Kullback-Leibler (KL) and Itakura-Saito (IS) cost functions by setting $\beta = 2$, $\beta = 1$ and $\beta = 0$, respectively. The derivation of Eqs. 5 and 6 is provided in a supplementary document [9].

2.2 Learning the basis functions

Figure 2 illustrates the iterative procedure of learning a number of templates for a single note labelled by the user. The user provides information about the start frame, the end frame and the pitch ϕ_0 of a note of a particular instrument i_0. This information can be illustrated by a pianoroll that contains a single line representing the note, as shown on the left-hand side of panel (a). Given this information, we can identify the matrix $\mathbf{W}^{\phi_0,i_0}$ in which the learned templates will be stored and the matrix $\mathbf{H}^{\phi_0,i_0}$ that contains the gains for each of the templates over time (grey-shaded matrices on the right-hand side of panel (a)). Since only those two matrices $\mathbf{W}^{\phi_0,i_0}$ and $\mathbf{H}^{\phi_0,i_0}$ are relevant for learning the templates from the labelled note, we isolate them from their tensors when illustrating the learning algorithm in panels (b)–(f).

Panels (b)–(f) display the algorithmic steps for estimating the spectral templates. This procedure is in fact very

similar to applying *k-means clustering* to the spectra of a note at all time frames within the spectrogram $\mathbf{V}$. In this analogy, each spectral template corresponds to a cluster mean and thus represents a set of spectra at different time frames. Since the learning procedure is carried out within the nonnegative framework, the correponding k-means clustering steps might not be obvious. For that reason, we illustrate these on the right hand side of panels (b)–(f). In these graphs, each data point corresponds to a spectrum of the note at a particular time frame in the N-dimensional space which is here for the sake of illustration reduced to 2 dimensions.

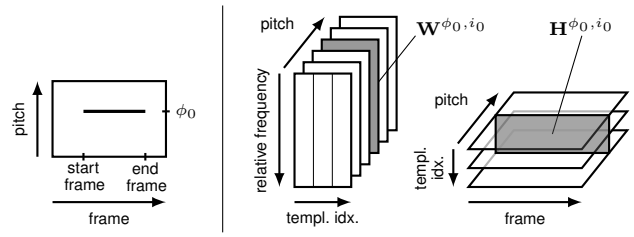
1. Initialisation: The algorithm starts by initialising the spectral templates in $\mathbf{W}^{\phi_0,i_0}$ with nonnegative random values (panel (b)). In the gain matrix $\mathbf{H}^{\phi_0,i_0}$ each frame of the note is randomly assigned to exactly one spectral template by setting the corresponding gains to a value of 1 while all other entries of the matrix are set to 0. In the k-means example, this corresponds to assigning the data points randomly to one of the three clusters: crosses, circles and squares.

2. Update: In the second step (panel (c)), we update the spectral templates in $\mathbf{W}^{\phi_0,i_0}$ based on the gains that were set in the previous step. This modifies the spectral templates in such a way that the resulting templates minimise the β-divergence at the assigned frames. Thus, each resulting spectral template can be seen as an average of the instrument spectra at the time frames that were assigned to it. In k-means clustering terms, this is equivalent to computing the average of the data points that were assigned to the same class. Note that in order to eliminate scale-ambiguities in the nonnegative framework, all spectral templates in $\mathbf{W}^{\phi_0,i_0}$ are scaled to have a power of 1 and the gains are adjusted accordingly.

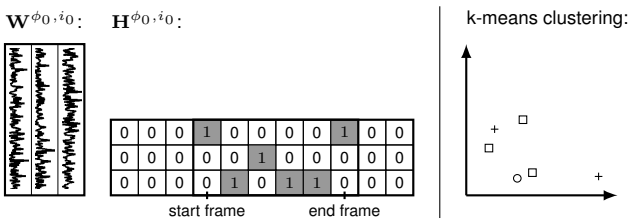
3. Assignment: In order to assign the spectra of the note at all frames to the template that best resembles their spectral shape, we set the template gains at each note frame to equal values (panel (d)) and update the gains based on the given spectral templates (panel (e)). This way, the gain matrix contains the contributions of each template to the audio spectra of each time frame when linearly combining the templates. This can be seen as a similarity measure between the templates and the spectra. We assign each frame to the template with the highest gain value, here indicated by the grey-shaded entries. In the k-means clustering example, this corresponds to the assignment step, in which each data point is assigned to the closest mean. We setup a new matrix $\mathbf{H}^{\phi_0,i_0}$ (panel (f)) that contains at each frame and each assigned template index the gains from step 2 (cf. panel (d)).

The algorithm iterates over steps 2 and 3.

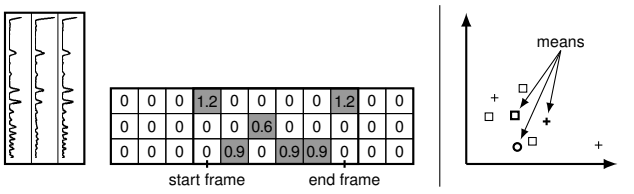
The reason for assigning each frame to just a single spectral template in step 1 and 3 is that we want to avoid the partials of a note to be split among the different templates. A template that only contains a subset of partials might be used by the algorithm to explain partials of other notes from the same or another instrument. An intuitive example for this case would be a spectral template that only contains a single partial (i.e. a single spectral peak) which can be used by the algorithm to approximate a partial of any note



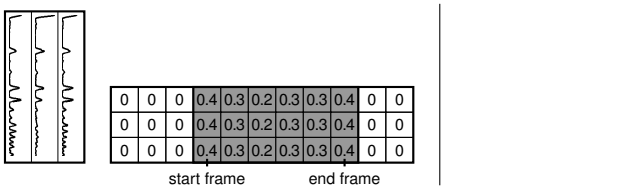
(a) Piano roll and svNMD framework



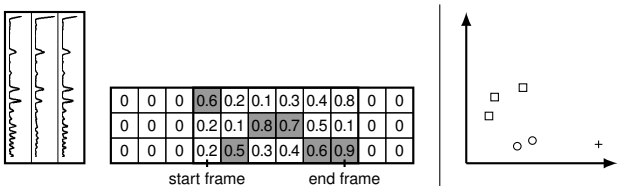
(b) Initialisation



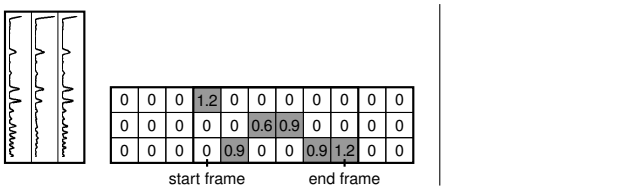
(c) Update



(d) Assignment (1)



(e) Assignment (2)



(f) Assignment (3)

Figure 2: Learning algorithm

at that position of the same or another instrument. This would produce a gain value either at the wrong fundamental frequency or the wrong instrument or both and thereby adulterate the transcription accuracy.

In k-means clustering, there is a chance of producing empty clusters when assigning the data points to the new means. The same problem applies to our proposed learning algorithm. In our algorithm this problem can occur in

panel (e), when for a certain template none of the frames contains the largest gains. In this case, we detect the largest cluster (i.e. the template with the largest number of assigned frames) and randomly assign half of its frames to the empty cluster. The spectral template of the empty cluster is then discarded and replaced by a duplicate of the spectral template of the largest cluster.

Although the learning procedure was here illustrated by an individual note of a single instrument, the procedure is applicable to and intended for polyphonic audio. A MATLAB implementation of the learning algorithm is available at `http://code.soundsoftware.ac.uk/projects/svnmdmt`.

3. EVALUATION

The evaluation of the proposed framework and the template learning algorithm was carried out in two experiments. In the first experiment (Sect. 3.3) we explored the upper limit of performance of the algorithm when used for semi-automatic transcription. The results of this experiment provide some intuition about the potential of the framework to accurately approximate a spectrogram. The second experiment (Sect. 3.4) looked at a more realistic semi-automatic transcription setting in which only a part of the notes are employed for learning the templates which are then applied to transcribe the remainder of the recording.

3.1 Dataset

For both experiments described below, the same dataset as in [4] was used. The dataset was based on monophonic recordings of musical phrases from 12 different instruments, each with a length of approximately 30s. Mixtures of 2 to 5 instruments were produced by combining the monophonic signals. For each polyphony level (2 to 5 instruments), 50 different combinations were generated. At the same time, the hand-annotated notes of the 12 monophonic files were available in MIDI format. Those MIDI files acted as the ground truth for the evaluation.

In addition to that, we evaluated the algorithm on more harmonically related instrument parts and computed results for a wind quintet excerpt (cf. [10]). This example had a length of 54s and for each instrument part hand-annotated MIDI ground-truth was available.

3.2 Accuracy

In order to measure the transcription accuracy, we refrained from using the common measures *precision*, *recall* or *F-score*. Those measures are used to compare detected note events to ground truth notes. Combining gains into note objects, however, would require a subsequent note-tracking algorithm which will have an influence on the results. Our aim is here to study the performance of the proposed algorithm in isolation.

As an accuracy measure, we therefore compute the percentage of energy in the gain matrices that is concentrated in the ground truth fundamental frequencies. This is done for each instrument individually. In order to achieve that, a summary gain matrix $\mathbf{G}^i$ is computed for each instrument i in the mixture by

$$\left[\mathbf{G}^i\right]_{\phi,n} = \sum_{t=1}^{T} \left[\mathbf{H}^{\phi,i}\right]_{t,n}. \tag{7}$$

Intuitively, in Fig. 1 this corresponds to summing all the displayed gain layers for each instrument. Based on the summary gain matrices $\mathbf{G}^i$, the per-instrument accuracies Acc_i are computed by

$$\mathrm{Acc}_i = \frac{\sum_{n=1}^{N} \sum_{\phi \in \mathcal{F}_n} \left(\left[\mathbf{G}^i\right]_{\phi,n}\right)^2}{\sum_{n=1}^{N} \sum_{\phi'=1}^{\Phi} \left(\left[\mathbf{G}^i\right]_{\phi',n}\right)^2}. \tag{8}$$

In this equation, $\mathcal{F}_n$ denotes the set of frequency bins of the annotated pitches in the n-th frame. Since the test set only contains monophonic instruments, $\mathcal{F}_n$ only contains the bins of at most one note at each time frame. Ideally, we would like to see all energy concentrated in the fundamental frequencies which would make it easy to detect notes within the gain matrices. This case would correspond to an accuracy Acc_i of 1.

3.3 Experiment 1: Exploring the upper performance limit

In the first experiment we explored the upper performance limit of the nonnegative framework when used for a semi-automatic transcription task. The upper performance limit is given when a user labels *all* notes of *all* instruments in the mixture under analysis. Although this scenario may seem trivial, because no transcription algorithm would be required if all notes were known beforehand, this evaluation provides an intuition about the expressivity of the algorithm and reveals any methodological flaws.

3.3.1 Experimental setup

For each file in the dataset, we extracted $T = 1$, 3 and 5 templates per pitch, by running 50 iterations of the template learning algorithm described in Sect. 2.2. The user information was given by the ground truth MIDI files of the instruments contained in the mixture which contained onset, offset and pitch information of the notes of the instruments. Once the basis functions were learned from the constant-Q magnitude spectrogram of the recording, the gain matrices were computed. This was done by randomly initialising all matrices $\mathbf{H}^{\phi,i}$ with nonnegative values and applying 10 iterations of the update equation for the gains (Eq. 6). Transcription accuracies were computed as described in Sect. 3.2. The experiment was conducted for the IS-divergence ($\beta = 0$) and the KL-divergence ($\beta = 1$).

3.3.2 Results

The results of this experiment are displayed in Fig. 3. The upper panels display the results obtained by using the Itakura-Saito (IS) divergence, the lower panels the results of the Kullback-Leibler (KL) divergence. From left to right, the panels show the results of the different polyphony levels — from 1 to 5 instruments — and on the right-hand side the results of the wind quintet. In each panel, we compare the per-instrument transcription accuracies of all instruments of all files when represented with different numbers of templates per pitch. The results are displayed as boxplots: the upper and lower edges of the box represent the first (Q_1) and third quartile (Q_3), the median is displayed in between. The whiskers extend to the data points that

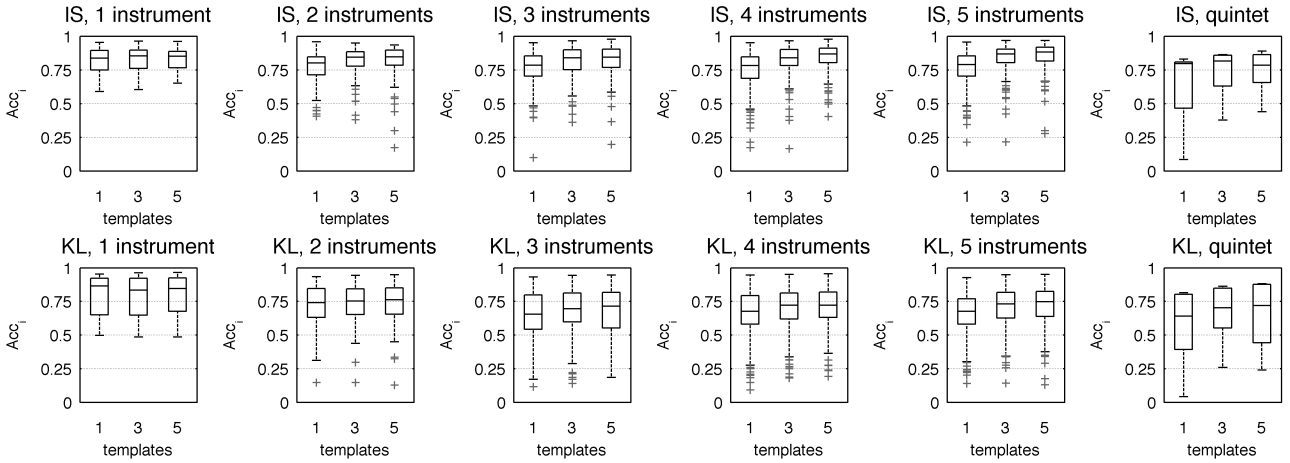


Figure 3: Results of experiment 1. The upper and lower rows display the per-instrument accuracies for the IS-divergence and KL-divergence, respectively. From left to right, the panels contain the accuracies for different polyphony levels and for the wind quintet. Within each panel the results for different numbers of templates per pitch are presented as boxplots.

are furthest away from the median, but within the interval $[Q_1 - 1.5 \cdot (Q_3 - Q_1) \ldots Q_3 + 1.5 \cdot (Q_3 - Q_1)]$. All data points outside that range are marked by crosses and considered as *outliers*.

When comparing the different cost functions for the random instrument mixtures, it becomes obvious that the Itakura-Saito divergence outperforms the Kullback-Leibler divergence in all cases. A possible explanation for the good performance of the IS-divergence is its scale-invariance property (cf. [11]) which is in compliance with Weber's law applied to the perception of loudness. An interesting aspect we found here is that by using the IS-divergence, the accuracies do not even noticeably decay when the number of instruments is increased.

When we compare the results for different numbers of spectral templates per pitch, a clear tendency towards higher accuracies can be observed when more templates are learned for each note. The improvement is consistent when the number of templates is increased from 1 to 3 and ranges between 2% and almost 10% for different poyphony levels when considering the median accuracies for the IS-divergence. Increasing the number of templates from 3 to 5 improves the accuracy even further, but not in the same consistent way as from 1 to 3.

The results of the wind quintet generally confirm the above findings, particularly the increasing accuracy when multiple templates are used. The median accuracy is however slightly lower than for the data set of random instrument mixtures, which can be attributed to the larger number of overlapping partials.

3.4 Experiment 2: Real case scenario

In the second experiment, we estimated the performance of a semi-automatic transcription system in a more realistic environment. We assumed that the user had labelled a certain number of notes for each instrument, which we use to estimate template spectra at the corresponding pitches. These template spectra are then used to build complete models for the instruments which are then applied to the remainder of the piece in order to obtain the transcription.

3.4.1 Experimental setup

For this experiment, we split each file in the dataset in two halves, each containing approx. 15 s of audio. We assumed that the user had labelled all notes of all instruments in the first half and used these to learn the basis functions as described above. The basis functions were then replicated at the surrounding pitches to cover the whole pitch range and were applied to estimate the gains of the second half of the audio.

As in the first experiment, we applied all combinations of cost functions (IS-divergence and KL-divergence), number of instruments (1–5) and number of templates per pitch (1,3 and 5). We again ran 50 iterations of the learning algorithm and 10 iterations for the estimation of the gain matrices.

3.4.2 Results

Figure 4 shows the results for the second experiment. The order of the results is the same as for the previous results.

For the random instrument mixtures, the results of this experiment differ from the results of the previous experiment. In general, there is a considerably larger variance in the results for each configuration. Several trends are clearly visible in the diagram: For both cost functions, the accuracy decreases when the number of instruments in the mixture is increased. The impression from the first experiment that the IS-divergence generally yields better results than the KL-divergence is here confirmed, the only exception being the polyphony level of one instrument. However, since the results for the monophonic audio files are only based on 12 accuracies, this fact needs to be put in perspective.

In terms of the different numbers of templates per pitch, the results for 1, 3 and 5 templates consistently stay in the same range and no clear trend can be found. It has to be considered here that the results of this experiment are not only influenced by the number of templates, but also by the fact that templates of non-annotated pitches were estimated by replicating adjacent pitches. It seems that the error introduced by this rough assumption outweighs the gain of having multiple templates per pitch.

The results for the quintet recording only show a small loss in accuracy to the previous experiment. The reason for

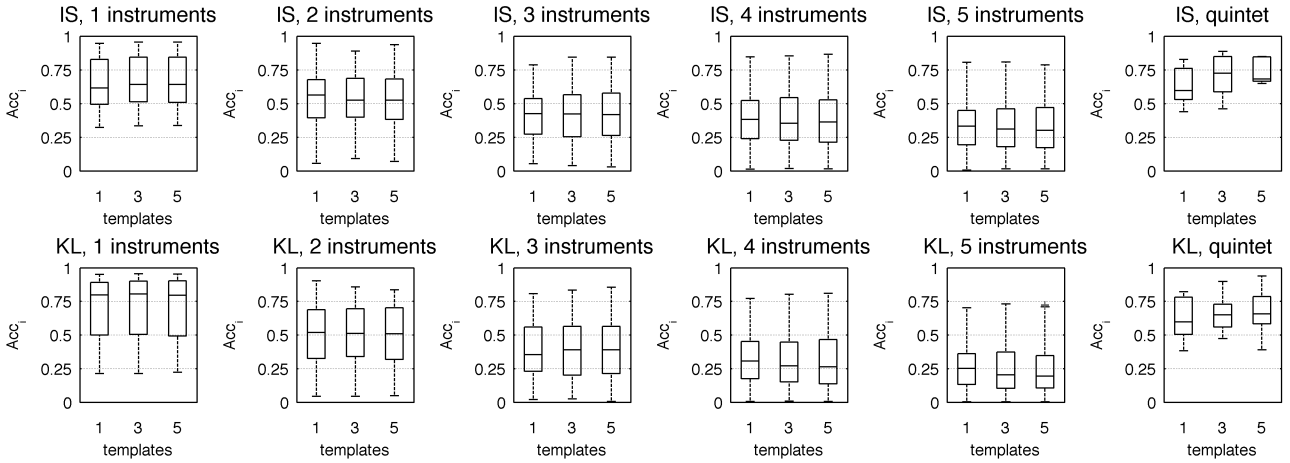


Figure 4: Results of experiment 2. The results are displayed in the same order as the ones in Fig. 3.

this can be seen in the fact that in this excerpt large parts of the first half are repeated in the second half, so that almost the same pitch range was covered for training and testing.

The experiments show a certain discrepancy between the maximum achievable accuracy and the accuracies that can be expected in a more realistic setting. There are several explanations for the fact that the accuracy of the second experiment is decreased: First, there was twice more training data in experiment 1. Second, in the first experiment the basis functions will have been better adjusted to the spectra of the second half of the audio files, which were not used in the learning process in the second experiment. And third, as indicated above, filling the gaps in the basis function tensors by merely replicating the estimated basis functions in the second experiment leads to a loss in accuracy.

4. CONCLUSION

We presented a shift-variant non-negative matrix deconvolution (svNMD) framework that represents each note of each instrument by multiple spectral templates. A learning algorithm was presented that allows the different templates to be estimated within the svNMD framework. The steps of this algorithm are comparable to a k-means clustering algorithm. We investigated the use of the framework for the task of semi-automtic music transcription in which the user provides a priori information about some notes in the mixture under analysis. Two experiments were carried out. In the first experiment, the upper performance limit of the algorithm was investigated which is given when the user provides information about all notes of all instruments. The results showed the superiority of the IS-divergence over the KL-divergence and a consistent improvement when more than one template per pitch was used. The second experiment expoited a more realistic use case in which the user merely labels a subset of the notes. Here, the superiority of the IS-divergence could be confirmed. In this experiment, however, no improvement could be found by using multiple templates per pitch.

5. REFERENCES

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

[2] D. D. Lee and H. S. Seung: "Learning the parts of objects by non-negative matrix factorization," *Nature*, Vol. 401, Nr. 6755, pp. 788-–791, 1999.

[3] P. Smaragdis and J.C. Brown: "Non-negative matrix factorization for polyphonic music transcription," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 177–180, 2003.

[4] H. Kirchhoff, S. Dixon, and A. Klapuri: "Shift-variant non-negative matrix deconvolution for music transcription," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012.

[5] M. Schmidt and M. Mørup: "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation," *6th International Conference on Independent Component Analysis and Blind Source Separation*, pp. 700–707, Charleston, USA, 2006.

[6] R. Hennequin, B. David, and R. Badeau: "Score informed audio source separation using a parametric model of non-negative spectrogram,". *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 45-–48, 2011.

[7] J. Ganseman, G. J. Mysore, J. S. Abel, and P. Scheunders: "Source separation by score synthesis," *International Computer Music Conference*, pp. 462–465, 2010.

[8] P. Smaragdis and G. J. Mysore: "Separation by humming: User-guided sound extraction from monophonic mixtures," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2009.

[9] H. Kirchhoff, S. Dixon, and A. Klapuri: "Derivation of update equations for multiple-template shift- variant non-negative matrix deconvolution based on β-divergence," Tech. Rep. C4DM-TR-06-12, Queen Mary University of London, 2012, `http://www.eecs.qmul.ac.uk/~holger/C4DM-TR-06-12`.

[10] E. Benetos and S. Dixon: "Joint multi-pitch detection using harmonic envelope estimation for polyphonic music transcription," *IEEE Journal of Selected Topics in Signal Processing*, Vol. 5, No. 6, pp. 1111–1123, 2011.

[11] C. Févotte, N. Bertin, and J. L. Durrieu: "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural Computation*, Vol. 21, No. 3, pp. 793–830, 2009.