Nonlinear Denoising, Linear Demixing

Rainer Kelz[1] **Gerhard Widmer**[1,2]
[1]Institute of Computational Perception, Johannes Kepler University, Linz, Austria
[2]LIT Artificial Intelligence Lab, Linz Institute of Technology, Austria
rainer.kelz@jku.at

Abstract

We cast the combinatorial problem of polyphonic piano transcription as a two stage process. A nonlinear denoising stage maps spectrogram representations of arbitrary piano music with unknown timbral characteristics onto a canonical spectrogram representation with known timbral characteristics. A subsequent linear demixing stage aims to exploit the knowledge about the canonical timbral characteristics. The idea behind this two stage process is to try to elegantly sidestep any musical bias inherent in the training dataset that is easily picked up by a single stage, nonlinear (neural) transcription system (with large capacity). The two stage process tries not to force the nonlinear system to solve a combinatorial problem, which is more amenable to being solved by a linear decomposition method that has the superposition property. Using the simplest setup we could think of, we obtain (rather mixed (pun intended)) results on a standard polyphonic piano transcription dataset — the two stage process still suffers from generalization problems after the first stage, which the second stage is unable to compensate.

1 Idea and Motivation

Polyphonic piano transcription is a specific instance of the more general automatic music transcription problem — given an audio recording of polyphonic music, produce a symbolic representation that describes the pitch of each note, as well as its start and end times. We can model this as a sequence labeling problem: we discretize the audio recording in time into so-called *frames* of fixed length, and output label indicator vectors for each such frame, that describe what pitches are currently sounding. Polyphonic piano transcription focuses only on the piano instrument class, which in turn encompasses many different physical designs, each with their own (subtly) different structural details, that may lead to pronounced differences in timbral characteristics. Pianos are capable of producing music with a high degree of polyphony, meaning many notes with different pitches can be played at the same time. Pianos also have extensive tonal and dynamic range. This means there are many possible pitches ($88+$), there are many intermediate levels possible between softest and loudest note, and soft notes sound quite different from loud notes for the same pitch. Current state-of-the-art systems for solving this problem are highly nonlinear and involve convolutional or recurrent neural networks, or both.

Due to the *combinatorial nature* of the polyphonic transcription problem, *all nonlinear systems* trained on a corpus of musical pieces will suffer from being biased towards the note combinations (or chords) that sound simultaneously most often. *Linear systems*, on the other hand, for example non-negative matrix decomposition methods, are not affected by this bias due to their very nature. Linear systems all have two desirable properties when it comes to such combinatorial problems, namely *additivity* $[f(a + b) = f(a) + f(b)]$ and *homogeneity* $[f(\lambda a) = \lambda f(a)]$. From these two properties, it is straightforward to conclude that a linear transcription function $f(\cdot)$ is (in principle) able to transcribe *any* arbitrary mixture of notes $[a + b + c + \dots]$, regardless of whether it has encountered such a combination during training. In fact, one of the simplest linear methods for solving the

I (Still) Can't Believe It's Not Better Workshop at NeurIPS 2021.

polyphonic transcription problem is non-negative matrix decomposition, where the dictionary matrix is usually trained with spectrograms of individual notes only. In the following, subscripts $\cdot_S$ or $\cdot_T$ denote non-negative matrices or vectors from a source domain (S) or a target domain (T) respectively. The source domain refers to the original set of (different) piano models that make up the training and testing datasets for our system. The target domain refers to an altogether different piano model, with different (but known) spectral characteristics from all the others in the dataset. We will also call this the *canonical* representation in the following.[1] Given a spectrogram $\mathbf{V}_S$ of a musical piece from an unseen piano, and a dictionary matrix $\mathbf{D}_T$ derived from known piano sounds, we find a non-negative activity matrix $\mathbf{A}$, representing the activity of the different dictionary entries over time, by minimizing some notion of reconstruction error (cf. Equation (1)).

$$\hat{\mathbf{A}}_S = \arg\min_{\mathbf{A}} \|\mathbf{D}_T\mathbf{A} - \mathbf{V}_S\|, \mathbf{A}_{ij} \geq 0 \quad (1) \qquad \hat{\mathbf{A}}_{T'} = \arg\min_{\mathbf{A}} \|\mathbf{D}_T\mathbf{A} - \mathbf{V}_{T'}\|, \mathbf{A}_{ij} \geq 0 \quad (2)$$

The main shortcoming of linear systems that are applied to the polyphonic transcription problem is their small modeling capacity. The capacity is directly limited by the entries in the dictionary matrix $\mathbf{D}_T$. In the simplest case, $\mathbf{D}_T$ consists of only one *spectral profile* for each individual note. If the spectral characteristics of the mixture of notes in the input spectrogram $\mathbf{V}_S$ are too dissimilar from the spectral profiles in $\mathbf{D}_T$, then the activity matrix $\mathbf{A}$ will be rather dense and contain many spurious entries — the opposite of a clean and sparse sequence labeling.

It would be very beneficial to know the dictionary $\mathbf{D}_S$ that belongs to the piano that produced the source domain spectrogram $\mathbf{V}_S$. Unfortunately, in all but a select few cases it is unknown. The basic idea is now to train a "denoising autoencoder" or "domain transfer function" $f_\theta : \mathcal{V}_S \to \mathcal{V}_T$, $f_\theta(\mathbf{V}_S) = \mathbf{V}_{T'}$, on matching spectrogram pairs $(\mathbf{V}_S, \mathbf{V}_T)$. This way, a previously unseen input $\mathbf{V}_S$ can be mapped onto a canonical, "denoised" version $\mathbf{V}_{T'}$, whose characteristics are better known and mostly contained in the matching dictionary $\mathbf{D}_T$, which ought to lead to much better decomposition performance (cf. Equation (2)).

Our initial expectation was that by setting the learning task up in this way, we *would not force the nonlinear parts* of the system to solve the combinatorial problem of polyphonic piano transcription on its own. The hope was that the network would focus only on implementing a "denoising" function, making it easier to apply a low capacity, linear decomposition method on the "denoised" or "domain transferred" inputs.

2 Realization and Experimental Setup

The denoising function f_θ takes on the form of a UNet [8][2]. The network is trained to minimize the mean squared reconstruction error on temporally aligned, matched pairs of spectrogram snippets $(\mathbf{V}_S, \mathbf{V}_T)$. All spectrogram snippets $\mathbf{V}_S$ originate from 160 different classical piano pieces, played by 9 different (virtual) pianos, and are taken from the MAPS dataset [3]. The exact number of pieces in the two train / validation / test splits can be found in Figure 2b. For each $\mathbf{V}_S$, the MIDI data that generated it is known, and was used to synthesize the corresponding, canonical spectrogram $\mathbf{V}_T$. We did so by using an open source software sampler called Fluidsynth[3] together with the soundfont "Fluid R3 GM"[4], where we selected the zeroth preset to render the audio. Each new audio recording was then converted into the spectrogram $\mathbf{V}_T$ with the exact same parameters used to compute the spectrogram from the source domain, $\mathbf{V}_S$. We compute three different decompositions $\hat{\mathbf{A}}_{S|T|T'}$ for two different train/test splits of musical pieces, that have different attributes and overlap relationships (cf. Figure 2b), and evaluate them against the groundtruth in the next section. All decompositions use the *same dictionary* $\mathbf{D}_T$, obtained by computing the mean spectral profile of individual notes produced by the Fluidsynth preset.

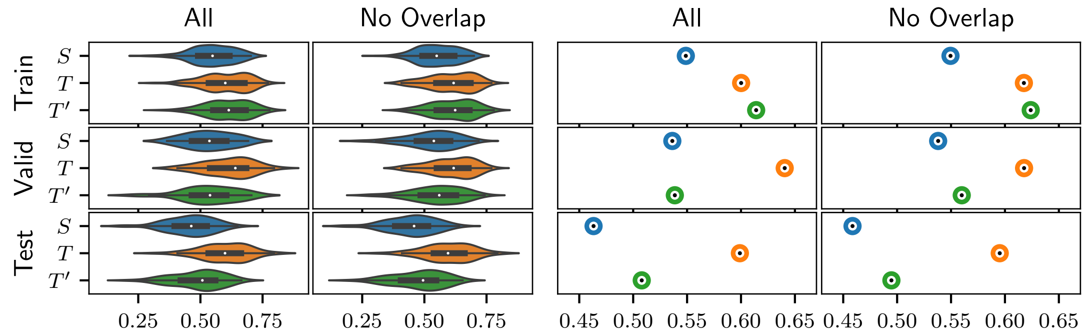
One quantity of interest for polyphonic transcription systems is framewise F-measure. To compute it, we first need to obtain binary label indicators $\mathbf{Y}_{S|T|T'}$ from the decompositions $\hat{\mathbf{A}}_{S|T|T'}$. The optimal detection threshold for each pitch is determined by concatenating all decompositions and

[1] Please see Appendix E for visual comparisons between the domains.

[2] For details and different variants of the objective function that were tried, please see Appendix B

[3] `www.fluidsynth.org`

[4] `http://www.musescore.org/download/fluid-soundfont.tar.gz`

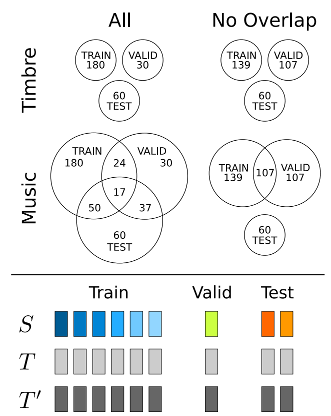


(a) Approximate F-Measure distribution (b) Medians of F-Measures

Figure 1: (a) shows the approximate distribution, (b) shows only the medians of F-Measures for the two different train / test splits called “All” and “No Overlap”.

Split		All $\mathcal{P}$	All $\mathcal{R}$	All $\mathcal{F}_1$	No Overlap $\mathcal{P}$	No Overlap $\mathcal{R}$	No Overlap $\mathcal{F}_1$
Train	S	0.572	0.585	0.549	0.555	0.610	0.554
	T	0.597	0.660	0.598	0.610	0.659	0.606
	T'	0.606	0.674	0.609	0.607	0.680	0.614
Valid	S	0.563	0.557	0.536	0.605	0.526	0.531
	T	0.612	0.666	0.611	0.605	0.664	0.606
	T'	0.611	0.531	0.535	0.609	0.563	0.551
Test	S	0.412	0.559	0.458	0.387	0.578	0.448
	T	0.592	0.663	0.596	0.594	0.657	0.595
	T'	0.416	0.649	0.489	0.403	0.651	0.478

(a) Median performance measure values over individual pieces



(b) Split Attributes and number of pieces per split

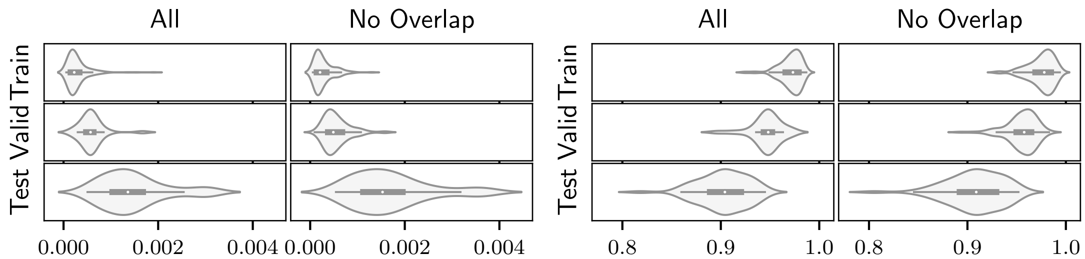
Figure 2: (a) shows the median, framewise precision, recall and F-measure values over individual pieces for all sets and all splits. (b) shows a sketch that tries to depict the different attributes of the different splits used. Set variants labeled S contain pieces with 9 different timbral characteristics (6 train, 1 valid, 2 test). Set variants labeled T contain exactly one, canonical, timbral characteristic, and set variants labeled T' contain exactly one, approximated, timbral characteristic.

groundtruths in the train and validation set, and finding the threshold that maximises framewise F-Measure between groundtruth and binary label indicators $\mathbf{Y}_{S|T|T'}$. These binarization thresholds are then used to obtain the results for all sets. This means that the results for train and validation sets are (very close to) optimal, with respect to the dictionary $\mathbf{D}_T$[5].

3 Related Work

In principle, any image-to-image translation method, such as [6] can be used at the “denoising stage”. Parts of the `TimbreTron` approach [5] come to mind. Were it not for the arguably *very close* appearance of source and target domain spectrograms in our case, we might have opted for adversarial losses in the first place — this could very well turn out to be the missing ingredient, in order to make the two stage approach work. The linear decomposition stage that assumes knowledge of the dictionary $\mathbf{D}$, could be made much more sophisticated as well, as in [2] for example. We did go for the simplest possible version however, because it is sufficient to determine feasibility.

[5]Details can be found in Appendix C.



(a) Approximate $\mathrm{mse}(\cdot)$ distribution (b) Approximate $\mathrm{ssim}(\cdot)$ distribution

Figure 3: Approximate distributions for two domain transfer quality measurements, mean squared error and structural similarity, over three different train / valid / test sets from the two splits called "All" and "No Overlap".

4 Results

The results are shown in Figures 1a, 1b and 2a. There are two train/test splits that we call "All" and "No Overlap". The test set is the same for both splits, and it contains music played by unknown piano models, meaning the timbral characteristics are always different between train and test. The difference between the splits called "All" and "No Overlap" is that for "All" there is considerable *musical overlap* between the train and test sets, meaning that the same musical piece occurs also in the train set, albeit played by a different piano. Of the 60 pieces in the test set, the musical content for 50 pieces occurs *at least once* in the train set. The musical overlap between test and validation set is 37 pieces. A somewhat helpful sketch outlining these properties is provided in Figure 2b. For more details on the musical overlap in the "All" split, please see Appendix D. We chose these two splits in order to be able to gauge the extent of the musical bias that is picked up by the domain transfer function, and will return to this question later.

Figure 1a shows the distribution over F-Measures for individual pieces, and Figure 1b focuses only on the medians of these distributions for better visual comparison. We can observe that a mismatch between spectral profiles in the dictionary $\mathbf{D}_T$ and spectral characteristics in the spectrogram $\mathbf{V}_S$ leads to lower transcription performance in terms of F-Measure. For the train set, we can also observe that the application of the domain transfer function to all spectrograms $\mathbf{V}_S$ in order to obtain all corresponding $\mathbf{V}_{T'}$ leads to improved performance, which is on par with or even *better*[6] than decomposition results on $\mathbf{V}_T$, where dictionary contents $\mathbf{D}_T$ and mixtures in the spectrogram match. Turning our attention to the test set, we notice the same gap between decomposition performance on $\mathbf{V}_S$ and $\mathbf{V}_T$. Regrettably, after applying the domain transfer function, the decomposition performance on $\mathbf{V}_{T'}$ does not increase nearly as much as for the train set. The main question is now, what is the cause for this performance gap?

5 Discussion and Interpretation of Results

Due to the "All" train/test split having considerable musical overlap between train and test sets, which is *removed* in "No Overlap", we can claim the following: the learned domain transfer function *still does* pick up some musical bias from the train set, *but* musical bias can not fully explain the large performance gap between decomposing $\mathbf{V}_T$ and $\mathbf{V}_{T'}$ on the test set. We attribute this gap to a failure of the domain transfer function to output spectrograms $\mathbf{V}_{T'}$ that are sufficiently similar to $\mathbf{V}_T$ on *previously unseen* data.

Our first claim is supported by the small difference in median F-measure from "All" / T' (0.489) to "No Overlap" / T' (0.478). Even though neither the task definition nor the minimized objective function for the training of the domain transfer function *necessitated* to exploit any knowledge about the musical content of the pieces, the setup also did *not explicitly* try to prevent the learning process from picking up any musical bias.

[6] We attribute this *increase* in performance from using $\mathbf{V}_{T'}$ over $\mathbf{V}_T$ to some additional temporal smoothing by the UNet that implements the domain transfer function.

Our second claim is (weakly) supported by the large performance gap between train and test for both splits "All" and "No Overlap". Much stronger, additional support for this line of reasoning can be seen in Figures 3a and 3b, where we computed two domain transfer quality measures for each individual piece in each set and each split. For the validation and test sets, the mean squared error $\mathrm{mse}(\cdot)$ (where lower means better) is considerably higher than for the train set. A similar gap can be observed for a structural similarity measure $\mathrm{ssim}(\cdot)$ [9] (where higher means better).

Hence, the main problem appears to be that the domain transfer function does not adequately generalize to spectrograms of unseen piano models. Possible next steps will focus on this aspect.

Acknowledgments

The LIT AI Lab is financed by the Federal State of Upper Austria.

References

[BBV09] Nancy Bertin, Roland Badeau, and Emmanuel Vincent. "Fast bayesian nmf algorithms enforcing harmonicity and temporal continuity in polyphonic music transcription". In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA '09, New Paltz, NY, USA, October 18-21, 2009*. IEEE, 2009, pp. 29–32. DOI: 10.1109/ASPAA.2009.5346531. URL: https://doi.org/10.1109/ASPAA.2009.5346531.

[Che+16] Tian Cheng, Matthias Mauch, Emmanouil Benetos, and Simon Dixon. "An Attack/Decay Model for Piano Transcription". In: *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*. Ed. by Michael I. Mandel, Johanna Devaney, Douglas Turnbull, and George Tzanetakis. 2016, pp. 584–590. URL: https://wp.nyu.edu/ismir2016/wp-content/uploads/sites/2294/2016/07/085%5C_Paper.pdf.

[EBD10] Valentin Emiya, Roland Badeau, and Bertrand David. "Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle". In: *IEEE Trans. Audio, Speech & Language Processing* 18.6 (2010), pp. 1643–1654.

[Hoy04] Patrik O. Hoyer. "Non-negative Matrix Factorization with Sparseness Constraints". In: *J. Mach. Learn. Res.* 5 (2004), pp. 1457–1469. URL: http://jmlr.org/papers/volume5/hoyer04a/hoyer04a.pdf.

[Hua+19] Sicong Huang, Qiyang Li, Cem Anil, Xuchan Bao, Sageev Oore, and Roger B. Grosse. "TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) Pipeline for Musical Timbre Transfer". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=S1lvm305YQ.

[Iso+17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-to-Image Translation with Conditional Adversarial Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 5967–5976. DOI: 10.1109/CVPR.2017.632. URL: https://doi.org/10.1109/CVPR.2017.632.

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6980.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*. 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28. URL: http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[Wan+04] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Trans. Image Process.* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861. URL: https://doi.org/10.1109/TIP.2003.819861.