A PROBABILISTIC SUBSPACE MODEL FOR MULTI-INSTRUMENT POLYPHONIC TRANSCRIPTION

Graham Grindlay
LabROSA, Dept. of Electrical Engineering
Columbia University
grindlay@ee.columbia.edu

Daniel P.W. Ellis
LabROSA, Dept. of Electrical Engineering
Columbia University
dpwe@ee.columbia.edu

ABSTRACT

In this paper we present a general probabilistic model suitable for transcribing single-channel audio recordings containing multiple polyphonic sources. Our system requires no prior knowledge of the instruments in the mixture, although it can benefit from this information if available. In contrast to many existing polyphonic transcription systems, our approach explicitly models the individual instruments and is thereby able to assign detected notes to their respective sources. We use a set of training instruments to learn a model space which is then used during transcription to constrain the properties of models fit to the target mixture. In addition, we encourage model sparsity using a simple approach related to tempering.

We evaluate our method on both recorded and synthesized two-instrument mixtures, obtaining average frame-level F-measures of up to 0.60 for synthesized audio and 0.53 for recorded audio. If knowledge of the instrument types in the mixture is available, we can increase these measures to 0.68 and 0.58, respectively, by initializing the model with parameters from similar instruments.

1. INTRODUCTION

Transcribing a piece of music from audio to symbolic form remains one of the most challenging problems in music information retrieval. Different variants of the problem can be defined according to the number of instruments present in the mixture and the degree of polyphony. Much research has been conducted on the case where the recording contains only a single (monophonic) instrument and reliable approaches to pitch estimation in this case have been developed [3]. However, when polyphony is introduced the problem becomes far more difficult as note harmonics often overlap and interfere with one another. Although there are a number of note properties that are relevant to polyphonic transcription, to date most research has focused on pitch, note onset time, and note offset time, while the problem of assigning notes to their source instruments has received substantially less attention. Determining the source of a note is not only important in its own right, but it is likely to improve overall transcription accuracy by helping to reduce cross-source interference. In order to distinguish between different instruments, we might wish to employ instrument-specific models. However, in general, we do not have access to the exact source models and so must estimate them directly from the mixture. This unsupervised learning problem is particularly difficult when only a single observation channel is available.

Non-negative Matrix Factorization (NMF) [8] has been shown to be a useful approach to single-channel music transcription [10]. The algorithm is typically applied to the magnitude spectrum of the target mixture, V, for which it yields a factorization $V \approx WH$ where W corresponds to a set of spectral basis vectors and H corresponds to the set of activation vectors over time. There are, however, several issues that arise when using NMF for unsupervised transcription. First, it is unclear how to determine the number of basis vectors required. If we use too few, a single basis vector may be forced to represent multiple notes, while if we use too many some basis vectors may have unclear interpretations. Even if we manage to choose the correct number of bases, we still face the problem of determining the mapping between bases and pitches as the basis order is typically arbitrary. Second, although this framework is capable of separating notes from distinct instruments as individual columns of W (and corresponding rows of H), there is no simple solution to the task of organizing these individual columns into coherent blocks corresponding to particular instruments.

Supervised transcription can be performed when W is known *a priori*. In this case, we know the ordering of the basis vectors and therefore how to partition H by source. However, we do not usually have access to this information and must therefore use some additional knowledge. One approach, which has been explored in several recent papers, is to impose constraints on the solution of W or its equivalent. Virtanen and Klapuri use a source-filter model to constrain the basis vectors to be formed from source spectra and filter activations [13]. Vincent et. al impose harmonicity constraints on the basis vectors by modeling them as combinations of narrow-band spectra [12]. In prior work, we proposed the *Subspace NMF* algorithm which learns a model parameter subspace from training examples and then constrains W to lie in this subspace [5].

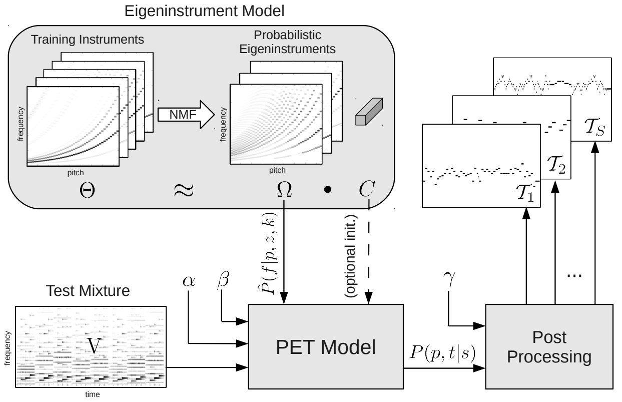

Figure 1. Illustration of the *Probabilistic Eigeninstrument Transcription* (PET) system. First, a set of training instruments are used to derive the eigeninstruments. These are then used by the PET model to learn the probability distribution $P(p,t|s)$, which is post-processed into source-specific binary transcriptions, $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_S$.

Recently, it has been shown [4, 9] that NMF is very closely related to *Probabilistic Latent Semantic Analysis* (PLSA) [6]. In this paper, we extend the *Subspace NMF* algorithm to a probabilistic setting in which we explicitly model the source probabilities, allow for multi-component note models, and use sparsity constraints to improve separation and transcription accuracy. The new approach requires no prior knowledge about the target mixture other than the number of instruments present. If, however, information about the instrument types is available, it can be used to seed the model and improve transcription accuracy.

Although we do not discuss the details here due to a lack of space, we note that our system effectively performs instrument-level source-separation as a part of the transcription process: once the model parameters have been solved for, individual sources can be reconstructed in a straightforward manner.

2. METHOD

Our system is based on the assumption that a suitably-normalized magnitude spectrogram, V, can be modeled as a joint distribution over time and frequency, $P(f,t)$. This quantity can be factored into a frame probability $P(t)$, which can be computed directly from the observed data, and a conditional distribution over frequency bins $P(f|t)$; spectrogram frames are treated as repeated draws from an underlying random process characterized by $P(f|t)$. We can model this distribution with a mixture of latent factors as follows:

$$P(f,t) = P(t)P(f|t) = P(t)\sum_z P(f|z)P(z|t) \qquad (1)$$

Note that when there is only a single latent variable z this is the same as the PLSA model and is effectively identical to NMF. The latent variable framework, however, makes it much more straightforward to introduce additional parameters and constraints.

Suppose now that we wish to model a mixture of S instrument sources, where each source has P possible pitches, and each pitch is represented by a set of Z components. We can extend the model described by (1) to accommodate these parameters as follows:

$$\tilde{P}(f|t) = \sum_{s,p,z} P(f|p,z,s)P(z|s,p,t)P(s|p,t)P(p|t) \qquad (2)$$

where we have used the notation $\tilde{P}(f|t)$ to denote the fact that our model reconstruction approximates the true distribution, $P(f|t)$. Notice that we have chosen to factor the distribution such that the source probability depends on pitch and time. Intuitively, this may seem odd as we might expect the generative process to first draw a source and then a pitch conditioned on that source. The reason for this factorization has to do with the type of sparsity constraints that we wish to impose on the model. This is discussed more fully in Section 2.2.2.

2.1 Instrument Models

$P(f|p,z,s)$ represents the instrument models that we are trying to fit to the data. However, as discussed in Section 1, we usually don't have access to the exact models that produced the mixture and a blind parameter search is highly under-constrained. The solution proposed in [5], which we extend here, is to model the instruments as mixtures of basis models or "eigeninstruments". This approach is similar in spirit to the eigenvoice technique used in speech recognition [7].

Suppose that we have a set of instruments models $\mathcal{M}$ for use in training. Each of these models $\mathcal{M}_i \in \mathcal{M}$ has FPZ parameters, which we concatenate into a super-vector, $\mathbf{m}_i$. These super-vectors are then stacked together into a matrix, Θ, and NMF with some rank K is used to find $\Theta \approx \Omega C$.[1] The set of coefficient vectors, C, is typically discarded at this point, although it can be used to initialize the full transcription system as well (see Section 3.4). The K basis vectors in Ω represent the eigeninstruments. Each of these vectors is reshaped to the F-by-P-by-Z model size to form the eigeninstrument distribution, $\hat{P}(f|p,z,k)$. Mixtures of this distribution can now be used to model new instruments as follows:

$$P(f|p,z,s) = \sum_k \hat{P}(f|p,z,k)P(k|s) \qquad (3)$$

where $P(k|s)$ represents an instrument-specific distribution over eigeninstruments. This model reduces the size of the parameter space for each source instrument in the mixture from FPZ, which is typically tens of thousands, to K which is typically between 10 and 100. Of course the quality of this parametrization depends on how well the eigeninstrument basis spans the true instrument parameter space, but assuming a sufficient variety of training instruments are used, we can expect good coverage.

[1] Some care has to be taken to ensure that the bases in Ω are properly normalized so that each section of F entries sums to 1, but so long as this requirement is met, any decomposition that yields non-negative basis vectors can be used.

2.2 Transcription Model

We are now ready to present the full transcription model proposed in this paper, which we refer to as *Probabilistic Eigeninstrument Transcription* (PET) and is illustrated in Figure 1. Combining the probabilistic model in (2) and the eigeninstrument model in (3), we arrive at the following:

$$\tilde{P}(f|t) = \sum_{s,p,z,k} \hat{P}(f|p,z,k)P(k|s)P(z|s,p,t)P(s|p,t)P(p|t) \tag{4}$$

Once we have solved for the model parameters, we calculate the joint distribution over pitch and time conditional on source:

$$P(p,t|s) = \frac{P(s|p,t)P(p|t)P(t)}{\sum_{p,t} P(s|p,t)P(p|t)P(t)} \tag{5}$$

This distribution represents the transcription of source s, but still needs to be post-processed to a binary pianoroll representation so that it can be compared with ground truth data. This is done using a simple threshold γ (see Section 3.3). We refer to the final pianoroll transcription of source s as $\mathcal{T}_s$.

2.2.1 Update Equations

We solve for the parameters in (4) using the Expectation-Maximization algorithm. This involves iterating between two update steps until convergence. In the first (expectation) step, we calculate the posterior distribution over the hidden variables s, p, z, and k, for each time-frequency point given the current estimates of the model parameters:

$$P(s,p,z,k|f,t) = \frac{\hat{P}(f|p,z,k)P(k|s)P(z|s,p,t)P(s|p,t)P(p|t)}{\tilde{P}(f|t)} \tag{6}$$

In the second (maximization) step, we use this posterior to maximize the expected log-likelihood of the model given the data:

$$\mathcal{L} = \sum_{f,t} V_{f,t} \log\Big(P(t)\tilde{P}(f|t)\Big) \tag{7}$$

where $V_{f,t}$ are values from our original spectrogram. This results in the following update equations:

$$P(k|s) = \frac{\sum_{f,t,z} P(s,p,z,k|f,t)V_{f,t}}{\sum_{f,k,t,z} P(s,p,z,k|f,t)V_{f,t}} \tag{8}$$

$$P(z|s,p,t) = \frac{\sum_{f,k} P(s,p,z,k|f,t)V_{f,t}}{\sum_{f,k,z} P(s,p,z,k|f,t)V_{f,t}} \tag{9}$$

$$P(s|p,t) = \frac{\sum_{f,k,z} P(s,p,z,k|f,t)V_{f,t}}{\sum_{f,k,s,z} P(s,p,z,k|f,t)V_{f,t}} \tag{10}$$

$$P(p|t) = \frac{\sum_{f,k,s,z} P(s,p,z,k|f,t)V_{f,t}}{\sum_{f,k,p,s,z} P(s,p,z,k|f,t)V_{f,t}} \tag{11}$$

2.2.2 Sparsity

The update equations given in Section 2.2.1 represent a maximum-likelihood solution to the model. However, in practice it can be advantageous to introduce additional constraints. The idea of parameter sparsity has proved to be useful for a number of audio-related tasks [1, 11]. For multi-instrument transcription, there are several ways in which it might make sense to constrain the model solution in this way. First, it is reasonable to expect that if pitch p is active at time t, then only a small fraction of the instrument sources are responsible for it. This belief can be encoded in the form of a sparsity prior on the distribution $P(s|p,t)$. Similarly, we generally expect that only a few pitches are active in each time frame, which implies a sparsity constraint on $P(p|t)$.

One way of encouraging sparsity in probabilistic models is through the use of the *entropic prior* [2]. This technique uses an exponentiated negative-entropy term as a prior on parameter distributions. Although it can yield good results, the solution to the maximization step is complicated, as it involves solving a system of transcendental equations. As an alternative, we have found that simply modifying the maximization steps in (10) and (11) as follows gives good results:

$$P(s|p,t) = \frac{\left[\sum_{f,k,z} P(s,p,z,k|f,t)V_{f,t}\right]^{\alpha}}{\sum_{s}\left[\sum_{f,k,z} P(s,p,z,k|f,t)V_{f,t}\right]^{\alpha}} \tag{12}$$

$$P(p|t) = \frac{\left[\sum_{f,k,s,z} P(s,p,z,k|f,t)V_{f,t}\right]^{\beta}}{\sum_{p}\left[\sum_{f,k,s,z} P(s,p,z,k|f,t)V_{f,t}\right]^{\beta}} \tag{13}$$

When α and β are less than 1, this is closely related to the *Tempered EM* algorithm used in PLSA [6]. However, it is clear that when α and β are greater than 1, the $P(s|p,t)$ and $P(p|t)$ distributions are "sharpened", thus decreasing their entropies and encouraging sparsity.

3. EVALUATION

3.1 Data

Two data sets were used in our experiments, one containing both synthesized and recorded audio and the other containing just synthesized audio. There are 15 tracks, 3256 notes, and 18843 frames in total. The specific properties of the data sets are summarized in Table 1. All tracks had two instrument sources, although the actual instruments varied. For the synthetic tracks, the MIDI versions were synthesized at an 8kHz sampling rate using *timidity* and the SGM V2.01 soundfont. A 1024-point STFT with 96ms window and 24ms hop was then taken and the magnitude spectrogram retained.

The first data set is based on a subset of the woodwind data supplied for the MIREX Multiple Fundamental Frequency Estimation and Tracking task. [2] The first 21 sec-

[2] http://www.music-ir.org/mirex/2009/index.php/Multiple_Fundamental_Frequency_Estimation_&_Tracking

	Type	# Tracks	# Notes	# Frames
Woodwind	S/R	6	1266	5424
Bach	S	3	724	7995

Table 1. Summary of the two data sets used. S and R denote synthesized and recorded, respectively.

onds from the bassoon, clarinet, oboe, and flute tracks were manually transcribed. These instrument tracks were then combined in all 6 possible pairings. It is important to note that this data is taken from the MIREX *development* set and that the primary test data is not publicly available. In addition, most authors of other transcription systems do not report results on the development data, making comparisons difficult.

The second data set is comprised of three pieces by J.S. Bach arranged as duets. The pieces are: *Herz und Mund und Tat und Leben* (BWV 147) for acoustic bass and piccolo, *Ich steh mit einem Fuß im Grabe* (BWV 156) for tuba and piano, and roughly the first half of *Wachet auf, ruft uns die Stimme* (BWV 140) for cello and flute. We chose instruments that were, for the most part, different from those used in the woodwind data set while also trying to keep the instrumentation as appropriate as possible.

3.2 Instrument Models

We used a set of 33 instruments of varying types to derive our instrument model. This included a roughly equal proportion of keyboard, plucked string, bowed, and wind instruments. The instrument models were generated with *timidity*, but in order to keep the tests with synthesized audio as fair as possible, a different soundfont (Papelmedia Final SF2 XXL) was used.[3] Each instrument model consisted of 58 pitches (C2-A6#), which were built as follows: notes of duration 1s were synthesized at an 8kHz sampling rate, using velocities 40, 80, and 100. A 1024-point STFT was taken of each, and the magnitude spectra were averaged across velocities to make the model more robust to differences in loudness. The models were then normalized so that the frequency components (spectrogram rows) summed to 1 for each pitch. Next, NMF with rank Z (the desired number of components per pitch) was run on this result with H initialized to a heavy main diagonal structure. This encouraged the ordering of the bases to be "left-to-right".

One potential issue with this approach has to do with the differences in the natural playing ranges of the instruments. For example, a violin generally cannot play below G3, although our model includes notes below this. Therefore, we masked out (i.e. set to 0) the parameters of the notes outside the playing range of each instrument used in training. Then, as described in Section 2.1, the instrument models were stacked into super vector form and NMF with a rank of $K = 30$ (chosen empirically) was run to find the instrument bases, Ω. These bases were then unstacked to form the eigeninstruments, $\hat{P}(f|p, z, k)$.

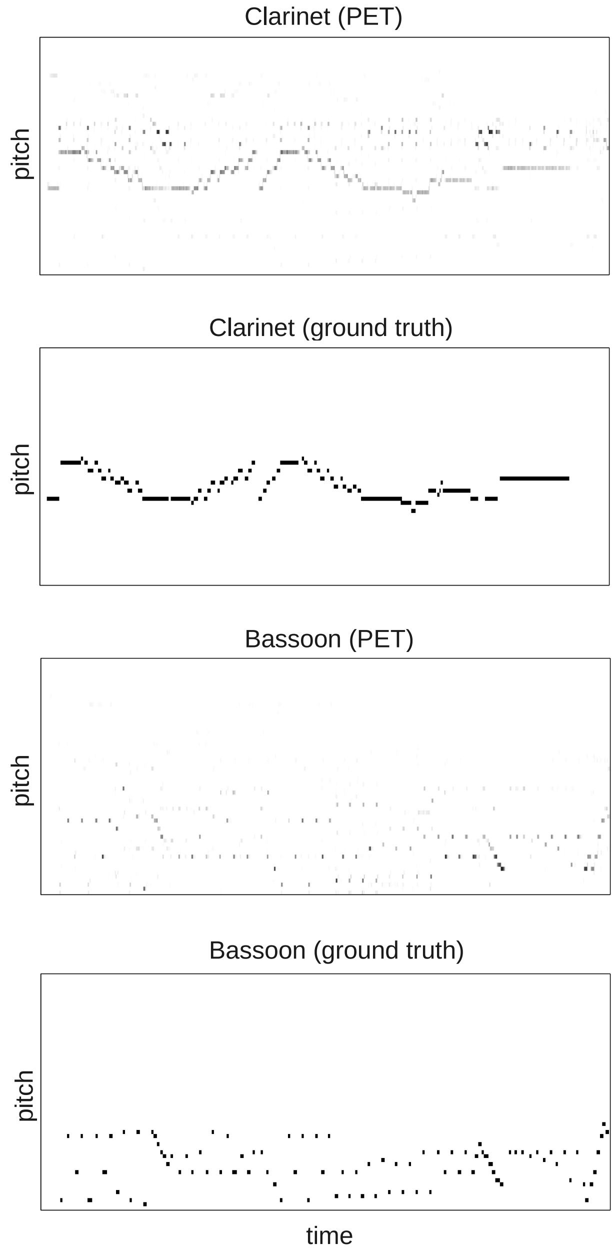

Figure 2. Example PET ($\beta = 2$) output distribution $P(p, t|s)$ and ground truth data for the bassoon-clarinet mixture from the recorded woodwind data set.

In preliminary experiments, we did not find a significant advantage to values of $Z > 1$ and so the full set of experiments presented below was carried out with only a single component per pitch.

3.3 Metrics

We evaluate our method using precision ($\mathcal{P}$), recall ($\mathcal{R}$), and F-measure ($\mathcal{F}$) on both the frame and note levels. Note that each reported metric is an average over sources. In addition, because the order of the sources in $P(p, t|s)$ is arbitrary, we compute sets of metrics for all possible permutations (two in our experiments since there are two sources) and report the set with the best frame-level F-measure.

[3] http://www.papelmedia.de/english/index.htm

When computing the note-level metrics, we consider a note onset to be correct if it falls within +/- 50ms of the ground truth onset. At present, we don't consider offsets for the note-level evaluation, although this information is reflected in the frame-level metrics.

The threshold γ used to convert $P(p,t|s)$ to a binary pianoroll was determined empirically for each algorithm variant and each data set. This was done by computing the threshold that maximized the average frame-level F-measure across tracks in the data set.

3.4 Experiments

We evaluated several variations of our algorithm so as to explore the effects of sparsity and to assess the performance of the eigeninstrument model. For each of the three data sets, we computed the frame and note metrics using the three variants of the PET model: PET without sparsity, PET with sparsity on the instruments given the pitches $P(s|p,t)$ ($\alpha = 2$), and PET with sparsity on the pitches at a given time $P(p|t)$ ($\beta = 2$). In these cases, all parameters were initialized randomly and the algorithm was run for 100 iterations.

Although we are primarily interested in blind transcription (i.e. no prior knowledge of the instruments present in the mixture), it is interesting to examine cases where more information is available as these can provide upper-bounds on performance. First, consider the case where we know the instrument types present in the mixture. For the synthetic data, we have access not only to the instrument types, but also to the oracle models for these instruments. In this case we hold $P(f|p,s,z)$ fixed and solve the basic model given in (2). The same can be done with the recorded data, except that we don't have oracle models for these recordings. Instead, we can just use the appropriate instrument models from the training set $\mathcal{M}$ as approximations. This case, which we refer to as "fixed" in the experimental results, represents a semi-supervised version of the PET system.

We might also consider using the instrument models $\mathcal{M}$ that we used in eigeninstrument training in order to initialize the PET model in the hope that the system will be able to further optimize their settings. We can do this by taking the appropriate eigeninstrument coefficient vectors $\mathbf{c}_s$ and using them to initialize $P(k|s)$. Intuitively, we are trying to start the PET model in the correct "neighborhood" of eigeninstrument space. These results are denoted "init".

Finally, as a baseline comparison, we consider generic NMF-based transcription (with generalized KL divergence as a cost function) where the instrument models (submatrices of W) have been initialized with a generic model defined as the average of the instrument models in the training set.

3.5 Results

The results of our approach are summarized in Tables 2–4. As a general observation, we can see that the sparsity factors have helped improve model performance in almost all cases, although different data sets benefit in different ways. For the synthetic woodwind data set, sparsity on sources, $P(s|p,t)$, increased the average F-measure on the frame-level, but at the note-level, sparsity on pitches, $P(p|t)$, had a larger impact. For the recorded woodwind data, sparsity on $P(p|t)$ benefited both frame and note-level F-measures the most. With the Bach data, we see that encouraging sparsity in $P(p|t)$ was much more important than it was for $P(s|p,t)$ on both the frame and note-level. In fact, imposing sparsity on $P(s|p,t)$ seems to have actually hurt frame-level performance relative to the non-sparse PET system. This may be explained by the fact that the instrument parts in the Bach pieces tend to be simultaneously active much of the time.

As we would expect, the baseline NMF system performs the worst in all test cases – not surprising given the limited information and lack of constraints. Also unsurprising is the fact that the oracle models are the top-performers on the synthetic data sets. However, notice that the randomly-initialized PET systems perform about

	Frame			**Note**		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
PET	0.56	0.64	0.56	0.42	0.73	0.51
$PET_{\alpha=2}$	0.60	0.61	0.60	0.46	0.73	0.56
$PET_{\beta=2}$	0.57	0.64	0.56	0.51	0.79	0.58
PET_{init}	0.71	0.68	0.68	0.64	0.86	0.71
PET_{oracle}	0.84	0.84	0.84	0.82	0.93	0.87
NMF	0.34	0.48	0.39	0.19	0.59	0.29

Table 2. Results for the synthetic woodwind data set. All values are averages across sources and tracks.

	Frame			**Note**		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
PET	0.52	0.52	0.50	0.41	0.73	0.50
$PET_{\alpha=2}$	0.49	0.57	0.51	0.41	0.78	0.51
$PET_{\beta=2}$	0.58	0.53	0.53	0.46	0.72	0.55
PET_{init}	0.60	0.60	0.58	0.48	0.82	0.58
PET_{fixed}	0.57	0.58	0.55	0.45	0.77	0.54
NMF	0.35	0.55	0.42	0.27	0.77	0.38

Table 3. Results for the recorded woodwind data set. All values are averages across sources and tracks.

	Frame			**Note**		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
PET	0.50	0.65	0.54	0.21	0.60	0.30
$PET_{\alpha=2}$	0.50	0.57	0.51	0.22	0.51	0.30
$PET_{\beta=2}$	0.55	0.66	0.59	0.24	0.65	0.34
PET_{init}	0.53	0.58	0.53	0.23	0.50	0.30
PET_{oracle}	0.91	0.85	0.87	0.53	0.83	0.64
NMF	0.36	0.50	0.42	0.09	0.46	0.14

Table 4. Results for the synthetic Bach data set. All values are averages across sources and tracks.

as well as the fixed model on recorded data. This implies that the algorithm was able to discover appropriate model parameters even in the blind case where it had no information about the instrument types in the mixture. It is also noteworthy that the best performing system for the recorded data set is the initialized PET variant. This suggests that, given good initializations, the algorithm was able to further adapt the instrument model parameters to improve the fit to the target mixture.

While the results on both woodwind data sets are relatively consistent across frame and note levels, the Bach data set exhibits a significant discrepancy between the two metrics, with substantially lower note-level scores. This is true even for the oracle model which achieves an average note-level F-measure of 0.64. There are two possible explanations for this. First, recall that our determination of both the optimal threshold γ as well as the order of the sources in $P(p, t|s)$ was based on the average frame-level F-measure. We opted to use frame-level metrics for this task as they are a stricter measure of transcription quality. However, given that the performance is relatively consistent for the woodwind data, it seems more likely that the discrepancy is due to instrumentation. In particular, the algorithms seem to have had difficulty with the soft onsets of the cello part in *Wachet auf, ruft uns die Stimme*.

4. CONCLUSIONS

We have presented a probabilistic model for the challenging problem of multi-instrument polyphonic transcription. Our method makes use of training instruments in order to learn a model parameter subspace that constrains the solutions of new models. Sparsity terms are also introduced to help further constrain the solution. We have shown that this approach can perform reasonably well in the blind transcription setting where no knowledge other than the number of instruments is assumed. In addition, knowledge of the types of instruments in the mixture (information which is relatively easy to obtain) was shown to improve performance significantly over the basic model. Although the experiments presented in this paper only consider two-instrument mixtures, the PET model is general and preliminary tests suggest that it can handle more complex mixtures as well.

There are several areas in which the current system could be improved. First, the thresholding technique that we have used is extremely simple and results could probably be improved significantly through the use of pitch dependent thresholding or more sophisticated classification. Second, and perhaps most importantly, although early experiments did not show a benefit to using multiple components for each pitch, it seems likely that the pitch models could be enriched substantially. Many instruments have complex time-varying structures within each note that would seem to be important for recognition. We are currently exploring ways to incorporate this type of information into our system.

5. ACKNOWLEDGMENTS

This work was supported by the NSF grant IIS-0713334. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

6. REFERENCES

[1] S.A. Abdallah and M.D. Plumbley. Polyphonic music transcription by non-negative sparse coding of power spectra. In *ISMIR*, 2004.

[2] M. Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11(5):1155–1182, 1999.

[3] A. de Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(1917), 2002.

[4] E. Gaussier and C. Goutte. Relation between PLSA and NMF and implications. In *SIGIR*, 2005.

[5] G. Grindlay and D.P.W. Ellis. Multi-voice polyphonic music transcription using eigeninstruments. In *WASPAA*, 2009.

[6] T. Hofmann. Probabilistic latent semantic analysis. In *Uncertainty in AI*, 1999.

[7] R. Kuhn, J. Junqua, P. Nguyen, and N. Niedzielski. Rapid speaker identification in eigenvoice space. *IEEE Transactions on Speech and Audio Processing*, 8(6):695–707, November 2000.

[8] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.

[9] M. Shashanka, B. Raj, and P. Smaragdis. Probabilistic latent variable models as non-negative factorizations. *Computational Intelligence and Neuroscience*, 2008, 2008.

[10] P. Smaragdis and J.C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *WASPAA*, 2003.

[11] P. Smaragdis, M. Shashanka, and B. Raj. A sparse non-parametric approach for single channel separation of known sounds. In *NIPS*, 2009.

[12] E. Vincent, N. Bertin, and R. Badeau. Harmonic and inharmonic non-negative matrix factorization for polyphonic pitch transcription. In *ICASSP*, 2008.

[13] T. Virtanen and A. Klapuri. Analysis of polyphonic audio using source-filter model and non-negative matrix factorization. In *NIPS*, 2006.