SEMI-SUPERVISED NMF WITH TIME-FREQUENCY ANNOTATIONS FOR SINGLE-CHANNEL SOURCE SEPARATION

Augustin Lefèvre
INRIA team SIERRA
augustin.lefevre@inria.fr

Francis Bach
INRIA team SIERRA
francis.bach@ens.fr

Cédric Févotte
LTCI/Telecom ParisTech
fevotte@telecom-paristech.fr

ABSTRACT

We formulate a novel extension of nonnegative matrix factorization (NMF) to take into account partial information on source-specific activity in the spectrogram. This information comes in the form of masking coefficients, such as those found in an ideal binary mask. We show that state-of-the-art results in source separation may be achieved with only a limited amount of correct annotation, and furthermore our algorithm is robust to incorrect annotations. Since in practice ideal annotations are not observed, we propose several supervision scenarios to estimate the ideal masking coefficients. First, manual annotations by a trained user on a dedicated graphical user interface are shown to provide satisfactory performance although they are prone to errors. Second, we investigate simple learning strategies to predict the Wiener coefficients based on local information around a given time-frequency bin of the spectrogram. Results on single-channel source separation show that time-frequency annotations allow to disambiguate the source separation problem, and learned annotations open the way for a completely unsupervised learning procedure for source separation with no human intervention.

1. INTRODUCTION

During the past decade, nonnegative matrix factorization (NMF) has become the core algorithm in single-channel source separation. A rich literature has been developed to adapt NMF to difficult scenarios in which sources are highly synchronized, and little or no development data is available.

In the past years, intensive research on Bayesian modelling and parameterized methods have been conducted to improve the identifiability of basis elements by restricting the complexity of the estimated model. More recently, another category of contributions consider incorporating information that is directly relevant to the data at hand, and specified by the user. In [2], time activation of the sources is used to specify direct constraints on the activation coefficients of the decomposition. Pitch estimates [5] were used for lead voice extraction. In [8], detailed score information is provided so that each individual note can be separated. While these contributions may use different NMF models, a common trait is that user information is used to specify the support of decomposition coefficients at the coding stage. A quite different line of work is proposed in [1, 3], where isolated signals are used as proxy for the source signals, so that information on both the basis functions and the activation coefficients can be used to constrain the factorization.

In this paper, we propose to annotate directly the time-frequency representation that is used to perform source separation. We assume that we are given recordings where a large fraction of time-frequency bins of the spectrogram may be assigned unambiguously to one dominant source. This hypothesis holds as long as there are not too many sources, and post-processing of the recording does not involve heavily non-linear effects. As illustrated in Figure 1, some patches in the spectrogram are cues for source-specific activity, which may be exploited as information on the optimal binary mask.

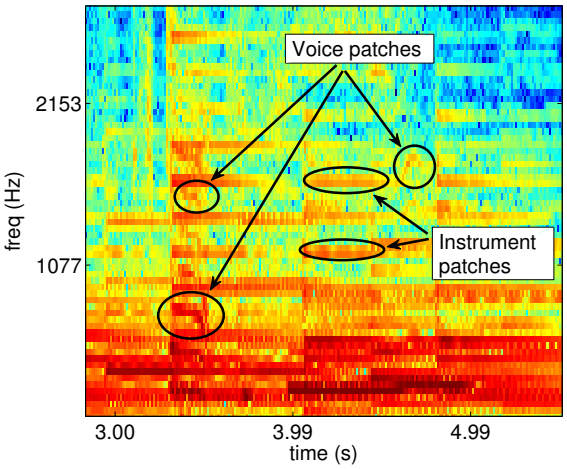


Figure 1: Cues from computational audio source analysis may be used as information on the optimal masking coefficients

In this article we make three contributions : we propose in Section 2 a novel modification of NMF (semi-supervised NMF) to take into account time-frequency annotations of the spectrogram, that is robust to errors in the annotations. In Section 2.2, we present a graphical user interface to retrieve such time-frequency annotations. In Section 3, we

propose supervised learning algorithms to automatize annotations, and explain how to combine them with semi-supervised NMF. Finally, we illustrate our contributions on publicly available source separation databases in Section 4.

2. SEMI-SUPERVISED NMF

2.1 Model and interpretation

In this section we propose a novel modification of NMF to incorporate annotations in the spectrogram. Let us first briefly summarize our NMF model and introduce mathematical notations, before proceeding to the main part of the contribution.

Given the short time Fourier transform of a signal $X \in \mathbb{C}^{F\times N}$ (in the following f indexes frequency and n time), we assume that $X = \sum_g S^{(g)}$, where $S^{(g)} \in \mathbb{C}^{F\times N}$ is the spectrogram of each source signal for $g \in \{1,\dots,G\}$. Define the power spectrograms of the sources $V^{(g)}_{fn} = |S^{(g)}|^2$. They are assumed to follow a linear model : $V^{(g)}_{fn} = \sum_{k=1}^{K_g} W^{(g)}_{fk} H^{(g)}_{kn}$, where $W^{(g)} \in \mathbb{R}_+^{F\times K_g}$, $H^{(g)} \in \mathbb{R}^{K_g\times N}$. Define $K = \sum_g K_g$, $W = (W^{(1)},\dots,W^{(G)}) \in \mathbb{R}_+^{F\times K}$ and $H^\top = ((H^{(1)})^\top,\dots,(H^{(G)})^\top) \in \mathbb{R}_+^{K\times N}$. Then, depending on the assumed distribution of $S^{(g)}$, estimation of W and H amounts to minimizing $d(V, WH)$ where d is a measure of fit between data and the underlying model. In this article we will use the Itakura-Saito divergence, but actually any β-divergence may be used.

Given estimates $\hat{V}^{(g)}_{fn}$ of the power spectrogram of each source, time domain estimates of the sources are then computed by Wiener filtering, where the Wiener coefficients of the source in the time-frequency domain are given by : $M^{(g)}_{fn} = \frac{\hat{V}^{(g)}_{fn}}{\hat{V}_{fn}}$.

The key idea in our contribution is the following : suppose we have at hand a set $\mathcal{L}$ of annotated time-frequency bins and a set of time-frequency masks $M^{(g)}_{fn}$ such that : $M^{(g)}_{fn} \in [0,1]$, and $\sum_g M^{(g)}_{fn} = 1$ if $(f,n) \in \mathcal{L}$, $\sum_g M^{(g)}_{fn} = 0$ otherwise.

For annotated time-frequency bins, we define target values for each source spectrogram : $\tilde{V}^{(g)}_{fn} = M^{(g)}_{fn} V_{fn}$.

The remaining, un-annotated entries of $\hat{V}$ are then computed so as to fit the observed spectrogram. This idea translates into the following optimization problem :

$$\min \sum_{(f,n)} d_{IS}(V_{fn}, \hat{V}_{fn}) + \lambda \sum_{\substack{(f,n)\in\mathcal{L}\\ g=1,\dots,G}} \mu_{fn} d_{IS}(\tilde{V}^{(g)}_{fn}, \hat{V}^{(g)}_{fn}), \quad (1)$$

where $d_{IS}(x,y) = \frac{x}{y} - \log\frac{x}{y} - 1$ is the Itakura-Saito divergence [1], and optimization is subject to the constraints that $W \geq 0$ (point-wise nonnegativity), $H \geq 0$, and $\sum_f W_{fk} = 1$ to avoid scaling ambiguity. We interpret the second term in Eq. (1) as a relaxed version of the constraints that $\hat{V}^{(g)}_{fn}$ be equal to their target value $M^{(g)}_{fn} V_{fn}$, for all annotated bins $(f,n) \in \mathcal{L}$.

[1] Given that some values are set to zero, we replace the IS divergence $d_{IS}(x,y)$ by $d_{IS}(\epsilon + x, \epsilon + y)$ (where $\epsilon = 10^{-7}$) in our optimization problem, in order to deal with ill-conditioning of the objective function.

We may tune the relative importance of annotation by varying parameter λ, from $\lambda = 0$ (standard NMF), to $\lambda \to +\infty$ (in which case $(WH)_{fn} = V^{(g)}_{fn}$ is enforced exactly if there are any feasible solutions). Thus, robustness to uncertainty in the annotations is introduced by replacing hard constraints by penalty terms in the NMF optimization problem. Note that since annotations dictate the assignment of components to sources, there is no need to group components by hand. We will discuss the role of μ_{fn} in the next section : in the case of user annotations, $\mu_{fn} = 1$. Let us discuss two cases :

(a) $\mathbf{M^{(g)}_{fn}} \in \{\mathbf{0},\mathbf{1}\}$: this is the case of user annotations, where time-frequency bins are labelled by hand. In this case, there can be only one active source at each time-frequency bin, since $\sum_g M^{(g)}_{fn} = 1$. This is a strong assumption, which is verified for a large fraction of the mixtures that are found in blind source separation.

(b) $\mathbf{M^{(g)}_{fn}} \in [\mathbf{0},\mathbf{1}]$: this general case is relevant to the learning procedures we introduce in the next section, since they output decision values in $[0,1]$.

Discussing the algorithm is beyond the scope of this paper : we used a multiplicative updates algorithm with appropriate modifications to deal with the additional terms in Eq. (1) [6].

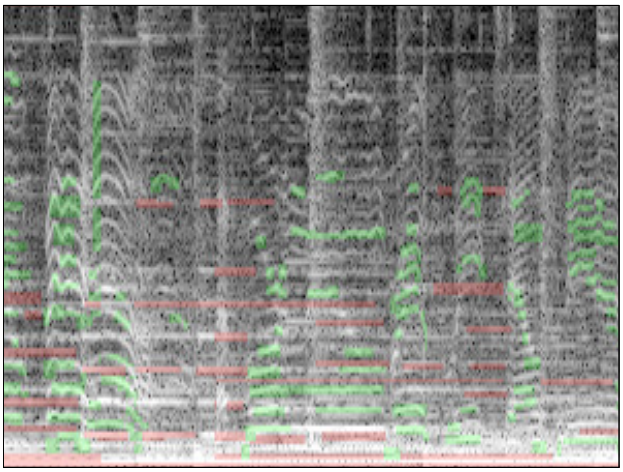


Figure 2: Example of user annotations in a ten seconds' audio track: green regions are assigned to voice, and red regions to accompaniment (**best seen in color**).

2.2 Relation with previous work

As in [2, 8, 5], annotations are used to constraint some sources to be inactive. In fact, time annotations are a special case of our model, where annotations are such that $M^{(g)}_{fn} = M^{(g)}_{f'n}$ for all (f, f') (i.e., zeroes come in columns). Our model deals with that case when there are two sources. The only difference between our model and [2] is that instead of enforcing $H_{kn} = 0$ as a hard constraint, we introduce a soft penalty to enforce $W_{fk}H_{kn} = 0$, with the added benefit that incorrect annotations are dealt with in a robust fashion. The case of more than two sources is dealt with a simple extension of Eq. (1), which we omit here for lack of space.

2.3 A graphical user interface for time-frequency annotation of spectrograms

In this section, we investigate manual annotation of the spectrogram. A GUI was designed in Matlab to annotate spectrograms (see Figure 2), with some extra sound functionalities to help the user. It takes sound files as input, applies some basic preprocessing (re-sampling at user-specified rate, down-mixing to mono), computes a time-frequency representation via user-specified parameters, and displays the spectrogram. Zooming and slide-rule navigation are enabled for better visualization. Annotation of sources is done with a simple rectangle drawing utility : one color for each source, as illustrated in Figure 2. Annotations are stored in an annotation mask of dimension $F \times N \times G$ (where (F, N) is the size of the spectrogram and G the number of sources). Several annotation masks may be loaded into memory and displayed alternatively, so the user can compare, for instance, manual annotations with the output of a blind source separation algorithm. Annotation masks may be exported to .mat format for further processing. Finally, we implemented playback functionalities to help the user annotate the spectrogram.

We designed the GUI to make the annotation process easier and faster : indeed, in our experience, while time annotations are easy and require only listening once or twice to the mix, time-frequency annotations are hard even for trained users : it takes up to one hour to annotate 20% of a twenty seconds track.

3. TOWARDS A SUPERVISED ALGORITHM FOR ANNOTATION

Research in computational audio scene analysis (CASA) has emphasized the role of frequency tracks in source identification : indeed by looking at a spectrogram, it is easy to assign a significant number of frequency tracks either to a voiced source or a musical source (see Figure 1). In previous works, such cues have been used to compute a similarity matrix that would then be used to perform clustering see [9, 4]. We propose here a supervised learning procedure to predict annotations automatically. At train stage, we have at hand separate sources so that we observe not only the mix, but also the Wiener coefficients $M_{fn}^{(g)}$ computed on the ground truth, while at test stage we only observe V. Thus, the goal is to predict $\mathbb{E}(M^{(g)}|V)$. In order to alleviate the computational burden [2], we make two restrictions on the learning procedure : each vector $(M_{fn}^{(1)}, \ldots, M_{fn}^{(G)})$ for a given time-frequency bin (f, n) is predicted independently of the others, and based only on the values of patches centered at that time-frequency bin.

We now introduce the features and algorithms used to train our predictor.

3.1 Features

The basic input to our learning algorithms consists in rectangular time-frequency blocks extracted from the input power spectrogram. The size of the rectangular blocks is fixed as a parameter of the algorithm. They are then normalized to have unit ℓ_1-norm so the features are scale invariant. We also considered taking the log of patches, adding coordinates of the patch as additional information, and taking a Gabor transform of the patches. The Gabor transform in particular was introduced so that correlations between pixels in each time-frequency blocks is taken into account. Finally, we also tried averaging the ground truth Wiener coefficients before learning, so that predicted regression surfaces are smoother in time-frequency space.

[2] indeed even for ten seconds' excerpts, there are more than 500×1000 time-frequency bins for standard STFT parameters

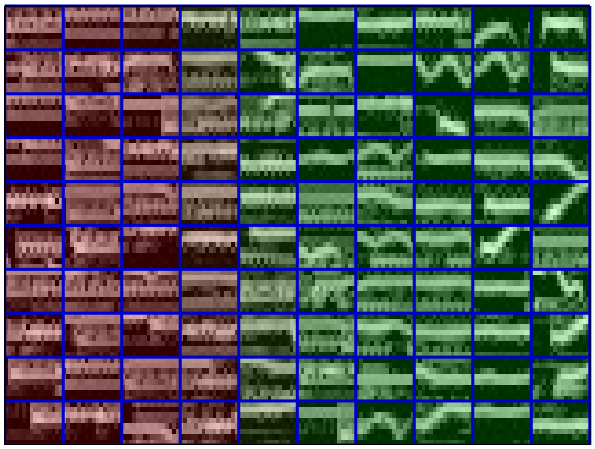


Figure 3: Samples of patches extracted from the SISEC database. Intensity reflects amplitude, patches which are labeled as accompaniment are in red, while patches which are labeled as voice are in green. Patches in brown have mixed Wiener coefficients (**best seen in color**).

3.2 Learning algorithms

Due to space limitation, we restrict ourselves to naming the algorithms we chose and highlighting the key parameters to tune. We refer the reader to standard textbooks on machine learning for more details (e.g., [7]).

K-nearest neighbors (knn): for each test point $x_i^{(test)}$, the C nearest points $x_j^{(train)}$, $j \in \{1, \ldots, C\}$ from the train set are used to predict $M_i^{(g)} = 1/C \sum_j M_j^{(g)}$.

Quantized knn (km): We learn C clusters from the train set using K-means; for each cluster, we compute average prediction coefficients $M_c^{(g)}$. For each test point, we predict $M_c^{(g)}$ from the nearest cluster c.

Random Forests (rf): We learn C regression trees of depth d from the train set and average over the C predictions for each test point.

We will refer to this supervised learning procedure as automatic annotations, no matter which algorithm is used.

3.3 Computation of μ_{fn} for automatic annotations

While the learning algorithms presented above predict Wiener coefficients, output values near 0.5 reflect uncertainty in the Wiener coefficients rather than prediction of mixed volumes. For this reason we introduce an additional tuning parameter μ_{fn} in Eq. (1), so that output values near 0.5

are less taken into account than values near $\{0, 1\}$. As a rule, we choose $\mu_{fn} = 1 - \frac{G}{G-1} \sum_g M_{fn}^{(g)}(1 - M_{fn}^{(g)})$, so that $0 \leq \mu_{fn} \leq 1$ and $\mu_{fn} = 0$ if all $M_{fn}^{(g)}$ are equal. Moreover, when annotations are in $\{0, 1\}$, we always have $\mu_{fn} = 1$.

4. EXPERIMENTAL RESULTS

4.1 Description of music databases

We used two publicly available databases in our experiments: the QUASI database [3] and the SISEC database for Professionally Produced Music Recordings [4]. All source tracks were down-sampled from 44100 Hz to 16000 Hz, and down-mixed to mono by taking the average of left and right channels. A voice track and accompaniment track are then created by aggregating the various source files, and then a final mix is created by summing the two tracks. Sine-bell windows of size 1024 with 512 overlap were used to compute short time Fourier transforms. The QUASI database contains longer tracks that are amenable to time annotations. The SISEC database contains short tracks where only time-frequency annotations can be used. Although detailed instrumental tracks are provided for most of the mixtures, we work only on single-channel signals. Since we are dealing with under-determined mixtures, we restrict ourselves to separating voice from accompaniment in each track, in order to alleviate the difficulty of the problem.

4.2 Ideal performance of semi-supervised NMF and robustness to wrong annotations

	SDR1	SDR2	SIR1	SIR2	SAR1	SAR2
0.1 %	-0.02	-0.60	5.15	5.16	3.62	2.33
1 %	0.70	0.24	4.59	6.25	4.39	2.85
10 %	6.71	6.68	13.57	16.53	7.95	7.40
100 %	10.40	10.41	19.88	20.88	11.00	10.88

Table 1: Mean results on the SISEC database, as the proportion of annotation increases.

Table 1 displays source separation results achieved by semi-supervised NMF on the SISEC database when fed with the actual Wiener coefficients computed from the ground truth sources. Source separation performance is measured by Source to Distortion Ratio (SDR), Source to Interference Ratio (SIR), and Source to Artefact Ratio (SAR). Higher values indicate better performance. As we can see, satisfactory results are obtained with as little as 10% of annotations. When 100% of annotations are given, NMF does nothing and the computed masks are simply the ideal Wiener coefficients computed from the sources.

We study the robustness of our NMF routine by replacing part of the ideal annotations by noise to simulate human errors. Table 2 displays average SDRs obtained when fixing the annotation rate to 10% and varying either the rate wrong annotations p or the optimization parameter λ. As expected, for fixed λ the average SDR drops as p increases. When p is fixed, there is an optimal value of λ that trades off the benefits and drawbacks of annotations. Fixing the target annotation rate to 10%, satisfactory results are obtained with up to 10% of wrong annotations (i.e.1% of the spectrogram).

λ	$p = 0$	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.5$
10^{-1}	0.11	-0.08	-1.76	-1.47	-1.47
10^{0}	5.59	4.10	3.50	2.29	1.20
10^{1}	7.59	6.53	5.32	3.43	0.59
10^{2}	7.07	5.66	4.54	3.15	0.77

Table 2: Mean SDR value as λ and the proportion of wrong annotations vary. The proportion of annotations is set to 0.1

4.3 Automatic annotation : comparison of algorithms and experimental results

method	**mean error (% improvement)**
4 8 loggabor km avg	0.141 $\pm$0.018 (14.9)
4 16 wcoords knn avg	0.140 $\pm$0.015 (15.9)
4 8 wcoords knn avg	0.138 $\pm$0.015 (16.8)
4 32 loggabor rf avg	0.137 $\pm$0.013 (17.4)
4 32 loggabor knn avg	0.137$\pm$0.010 (17.4)

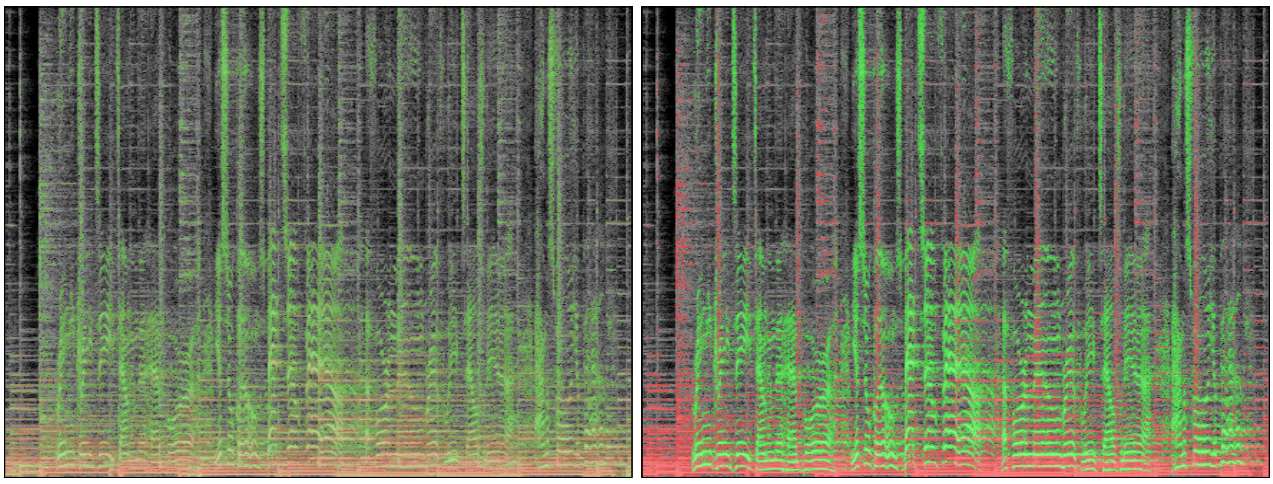
Table 3: Mean error on Wiener coefficient predictions on the SISEC database ($\%$ improvement over random prediction), for various learning strategies .

Learning algorithms were trained by dividing the SISEC database in two sets of tracks. For each set, we train detectors and test them on the other set. Thus we may compute annotations and run semi-supervised NMF for all tracks without the risk of overfitting. We emphasize the fact that each track is annotated with a detector that has never seen the spectrogram before : our method is purely supervised with no adaptation to test data. Parameters of the learning algorithms were selected at train stage by cross-validation. Time-frequency patches of size in $\{4, 8\}\times\{8, 16, 32\}$ were extracted. Out of each track we extract 5×10^3 patches at train time, and 10^5 patches a test time, so approximately 10% of the track is annotated at test time when semi-supervised NMF is called.

We display in Table 3 the results of the best 5 detectors, in terms of mean prediction error (first column) and in terms of relative improvement over a purely random predictor. Detectors are named after the following rule : {patch size} {feature} {learning method} {averaging or identical}. For instance, the tag **loggabor** corresponds to taking log then Gabor transform of patches, and **wcoords** adding frequency coordinates of the patches as side information. Note that we used exact Wiener coefficients to compute errors, so that all detectors can be compared even when averaging was used at train stage. The improvement over a random predictor is consistent across the features and the algorithms that were used. Figure 4 compares annotations provided by the best detectors from Table 3 with

[3] www.tsi.telecom-paristech.fr/aao/

[4] sisec.wiki.irisa.fr



(a) Automatic (b) Correct

Figure 4: Comparison of automatic annotations and correct annotations (at the same time-frequency bins). Gray-scale time-frequency bins are not annotated, red bins are annotated as accompaniment, green bins as voice(**best seen in color**).

ideal annotations at the same points were automatic annotations were made. Red time-frequency bins correspond to accompaniment, and green to voice. The most striking observation is that, while ideal annotations are in very bright colors (few Wiener coefficients are different from 0 or 1), automatic annotations, on the other hand, are generally biased towards 0.5. This is to be expected since predicting 0.5 incurs a risk of losing at most 0.25 (since we use a regression loss), while predicting 0 or 1 incurs a maximum loss of 1. The main asset of automatic annotations is that pitch tracks with varying frequency are successfully predicted as voice. Automatic annotations are biased towards predicting voice in the higher frequencies : however the learning algorithm in this example did not have the information of frequency. This might be because transients "look" a lot like patches of unvoiced speech. Finally, one may spot inconsistencies in the predictions in the sense that points belonging to the same pitch tracks are sometimes classified incoherently, which is not surprising since the learning algorithms we have proposed predict time-frequency bins independently.

To sum up, predictions of Wiener coefficients from local patches are not perfect but provide a good starting point for further modelling of the spectrogram. We expect that better performance could be obtained by using more advanced cues from CASA, such as pre-clustering the spectrogram into pitch tracks and transient tracks, before learning [5] .

4.4 Overall results

We now turn to results obtained by semi-supervised NMF combined with various annotation methods. On the SISEC database, manual time-frequency annotations were done with the GUI presented in Section 2.2. On the QUASI database, tracks were amenable to significant time annotations, so by comparing results on both databases we can compare the respective benefits of time-frequency annotations VS time annotations.

In both scenarios, we compare five methods :

auto : Automatic annotations and semi-supervised NMF. The best detector from Table 3 was chosen.

user : User annotations and semi-supervised NMF (time-frequency annotations for SISEC, manual annotations for QUASI). We tried $K \in \{5, 10, 20\}$ for the SISEC database and $\{10, 20, 50\}$ for the QUASI database, as well as $\lambda \in \{1, 10, 100\}$, and selected parameters yielding highest SDR for fair comparison with the baseline.

baseline : Run NMF and permute factors to obtain optimal SDR. We set $K = 8$ because it already takes a 10 times as long to evaluate SDRs for all permutation on a single track as it takes to run semi-supervised NMF.

self : set $s^{(g)} = \frac{1}{G}x$ as estimates for the sources, it serves to estimate the difficulty of the source separation problem for a given database.

oracle : results obtained with Wiener coefficients computed from the ground truth. In addition we display track by track annotation accuracy for user annotations, for comparison with Table 2. For each method, we ran NMF three times for 1000 iterations to avoid local minima, and kept the run with the lowest objective cost value.

Tables 5a and 5b display average evaluation metrics for each source (source 1 is always the accompaniment, and source 2 is always the voice), on two different databases :

	% annotated	% correct
track 1	0.23	0.91
track 2	0.10	0.89
track 3	0.29	0.91
track 4	0.17	0.81
track 5	0.22	0.95

Table 4: Evaluation of user annotations on the SISEC database.

[5] This is very similar to what is done in vision, where super-pixels help deal with consistency in prediction and alleviate the computational burden of predicting all pixel values.

	auto	user (t-f)	baseline	self	oracle
SDR1	0.97	6.21	6.16	3.09	14.79
SDR2	0.51	2.58	1.61	-3.18	11.53
SIR1	3.17	18.64	9.91	3.09	24.00
SIR2	4.57	11.35	5.09	-3.18	23.90
SAR1	6.74	6.91	9.26	279.17	15.41
SAR2	4.18	3.91	5.58	279.17	11.84
% ann.	8.69	19.81	0.00	0.00	100.00

(a) SISEC

	auto	user (t)	baseline	self	oracle
SDR1	6.76	7.59	6.29	6.21	16.88
SDR2	-4.33	-4.57	-1.71	-6.22	10.37
SIR1	6.97	15.05	13.81	6.21	25.62
SIR2	-3.75	4.09	1.88	-6.22	24.83
SAR1	21.91	9.00	7.71	268.45	17.66
SAR2	10.28	0.21	4.29	268.45	10.60
% ann.	6.91	100.00	0.00	0.00	100.00

(b) QUASI

Table 5: Results on the evaluated databases: (a) time-frequency annotations, (b) time annotations.

on the SISEC database, we experimented with time-frequency annotations since the tracks were too short for time annotations. Overall, results on the SISEC database are better than those on QUASI. Our interpretation is that since most of the time the accompaniment is active, the dictionaries tend to overfit the accompaniment and underfit the voice. Time-frequency annotations on SISEC yield SDRs that are a few points below that predicted by our benchmark from Table 2 : indeed human errors are not distributed randomly as was the case in our benchmark. Time-frequency annotations outperform the baseline by 1 point in SDR, which is significant because in semi-supervised NMF there is no manual grouping of the components. Time annotations loose to the baseline by -1 in SDR, but they are still significantly correlated with the true sources when compared with the baseline.

On the SISEC database, automatic annotations are also below the baseline, however they are also significantly correlated with the true sources, when compared with the "self" column. Signal to Interference Ratios are even comparable with those of the baseline on the SISEC database. Automatic annotations do not perform as well on the QUASI database since we trained detectors only on tracks from SISEC, so that more supervision would significantly improve those figures.

To conclude, we have shown that time-frequency annotations can improve significantly over NMF with ideally grouped components. On longer tracks, time only annotations yield reasonable results, but even when 100% of the track is annotated, the estimated sources contain strong interferences. Automatic annotations yield similar results, but leave considerable room for improvement, since with time-frequency annotations there will always be a point where enough annotations with limited errors will provide audible estimates of the sources.

5. CONCLUSION

We have proposed a novel formulation of semi-supervised NMF that successfully takes into account annotations to enhance the discriminative power of NMF. Semi-supervised NMF is defined so that when a certain amount of annotations is reached, source separation quality is near that of ideal binary masks. Manual annotations retrieved with our graphical user interface yield satisfactory results. We are investigating ways to define annotations independently of the particular time-frequency representation that is used.

Finally, semi-supervised NMF opens the way for interaction with methods from computational audio scene analysis. As such, the simple features and textbook pattern matching algorithms we have presented show promising results.

6. REFERENCES

[1] P. Smaragdis and G. Mysore. "Separation by "humming": User-guided sound extraction from monophonic mixtures.", *WASPAA*, 2012.

[2] B. Wang. "Musical Audio Stream Separation", *Msc Thesis*, 2009.

[3] D. Fitzgerald. "User Assisted Source Separation using Non-negative Matrix Factorisation", *Irish Signals and Systems Conference*, 2011.

[4] F. Bach and M.I. Jordan. "Blind one-microphone speech separation: a spectral learning approach." *NIPS*, 2004.

[5] J.-L. Durrieu and J.-P. Thiran. "Musical audio source separation based on user-selected f0 track." *(LVA/ICA)*, 2012.

[6] C. Févotte, N. Bertin, and J.-L. Durrieu. "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis." *Neural Computation*, 2009.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, 2nd edition edition, 2009.

[8] R. Hennequin, B. David, and R. Badeau. "Score informed audio source separation using a parametric model of non-negative spectrogram." *ICASSP*, 2011.

[9] M. Lagrange, L.G. Martins, J. Murdoch, and G. Tzanetakis. "Normalized cuts for predominant melodic source separation." *TASLP*, 2008.