FOLLOWING MUSICAL SOURCES BY EXAMPLE

Paris Smaragdis[†,‡], *Gautham J. Mysore*[‡]

[†]Depts. of Computer Science and Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, Urbana, IL, USA
[‡]Advanced Technology Labs
Adobe Systems Inc., San Fransisco, CA, USA

ABSTRACT

In this paper we present a system that is capable of tracking the pitch and volume of a musical source by making use of training data. We show how we can use pitch-tagged training example sounds to construct a model of a target source, and then use that model to track such a source in unseen mixtures. We do so using a regularized decomposition approach that is designed to strive for semantic continuity in its estimates.

Index Terms— Polyphonic transcription, non-negative models

1. INTRODUCTION

For humans, understanding musical sources and being able to detect and transcribe them when observed inside a mixture is a learned process. Through repetitive ear training exercises we learn to associate sounds with specific instruments and notes, and eventually we develop the ability to understand music using such terms. The computerized counterpart of this approach is, however, not as developed. There are multiple approaches that attack the problem of dissecting a musical mixture into sources and notes, but they are most often based on user-defined principles or treated as parameter estimation problems that rely on exact definitions of source energy and pitch.

In this paper we explore a different approach, we examine the basis of a system that is capable of being instructed how a target source sounds like and what its notes are, and then use that information in order to discover that source when it is present in complex musical mixtures. We show that at an abstract level this approach can be casted as a decomposition with an added constraint for semantic continuity and examine its performance on real-world music data.

2. PROPOSED APPROACH

The basic tool we will make use of in this paper of is a probabilistic non-negative decomposition of spectrograms. Our approach will be similar to the one in [1], but we will consider the case where not all sources are known. In doing so we will introduce a new constraint that ensures *semantic continuity*. In the following subsections we will describe the basic approach and then show how it can be enhanced for semantic continuity.

2.1. Probabilistic decompositions of sources

The formulation that we will use will decompose normalized magnitude spectra into a set of overcomplete dictionary elements and their corresponding weights. At its basic level, this approach similar to many already existing NMF/PLCA models. These approaches can be interpreted as either non-negative factorizations or as latent variable probabilistic models. We will use the latter interpretation since that would allow us to perform intuitive inference when using priors and constraints.

In order to setup the notation that we will use we will begin by describing the basic model. For a sound $s(t)$ we will obtain its time-frequency transform:

$$S_t(f) = \mathcal{F}[s(t, \dots, t+N-1)]. \tag{1}$$

For the purposes of this paper, the transform $\mathcal{F}(\cdot)$ will be a Fourier transform with the appropriate use of a tapering window in order to minimize spectral leakage. The use of alternative transforms (e.g., constant-Q or warped Fourier transforms) is also possible, but lie outside for the scope of this paper.

Like all operations that focus on timbral and pitch characteristics we desire to have invariance from phase and scale changes. In order to do so we will only retain the magnitude of the time-frequency transform and also normalize all its time frames so that they sum to a constant value (1 in our case):

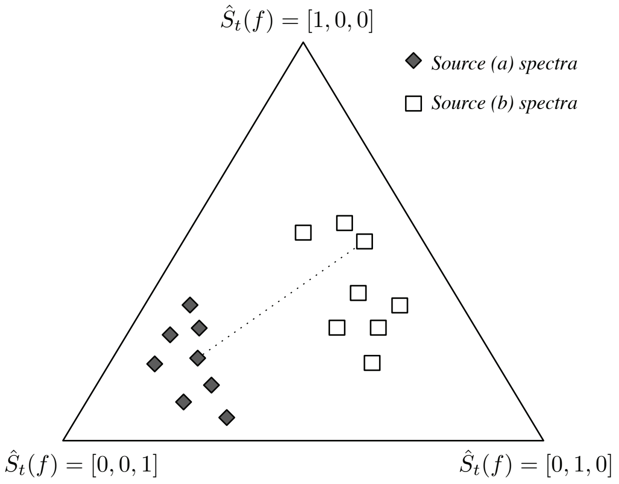
$$\hat{S}_t(f) = \frac{||S_t(f)||}{\sum_f ||S_t(f)||} \tag{2}$$

By analyzing a sound using this process we are left with a set of normalized magnitude spectra that describe all its observable spectral configurations. It is convenient to represent this space of spectra inside a simplex, a space that can contain all possible normalized spectra. For most sounds, their constituent normalized spectra will occupy a subspace of that simplex, an area that defines their timbral characteristics. A simple example with normalized spectra of only three frequencies is shown in figure 1. A very convenient feature of this representation is that whenever two normalized spectra mix, the resulting normalized spectrum will lie on the line that connects the original spectra. In order to aid the subsequent inference task it is also helpful to think of the normalized spectra as being probability distributions of energy across frequencies. Using that interpretation we would have the probability of frequency f at time frame t to be $P_t(f) \equiv \hat{S}_t(f)$.

We are now ready to define a probabilistic model that can analyze mixtures based on a prior learning step where source examples are used. The basic form of the model we will use is:

$$\begin{aligned} P_t(f) \approx P_t(a) \sum_z P^{(a)}(f|z) P_t^{(a)}(z) \\ + P_t(b) \sum_z P^{(b)}(f|z) P_t^{(b)}(z) \end{aligned} \tag{3}$$

978-1-4673-0046-9/12/$26.00 ©2012 IEEE



$\hat{S}_t(f) = [1, 0, 0]$
◆ Source (a) spectra
□ Source (b) spectra
$\hat{S}_t(f) = [0, 0, 1]$
$\hat{S}_t(f) = [0, 1, 0]$

Fig. 1. *The normalized spectra $\{3\}$-simplex with samples from two sources. In general, dissimilar sources will occupy different parts of that space. The line defined by connecting any two spectra (the dotted line in this space) contains all the possible normalized spectra that a mixture of these two can generate.*

The spectral probabilities $P_t(f)$ are the measurements we make by observing a mixture of two sound classes, and represent the probability of observing energy at time time t and frequency f. This is then approximated as a weighted sum of a set of dictionary elements $P^{(a)}(f|z)$ and $P^{(b)}(f|z)$. These dictionary elements can be learned from training examples for the two sound classes (a) and (b). The two sets of weights, $P_t^{(a)}(z)$ and $P_t^{(b)}(z)$, combined with the source priors, $P_t(a)$ and $P_t(b)$, regulate how these dictionary elements are to be combined to approximate the observed input. All of the above probabilities are discrete and contain a finite number of elements. The latent variable z serves as an index for addressing the dictionary elements. All parameters of this model can be learned using the Expectation-Maximization algorithm as shown in [2].

Variations of this approach have been used for many source separation and denoising problems. The one that we will make use of in this paper is the approach in [3]. In that work instead of learning a set of dictionary elements it was shown that one can simply assign the dictionary elements to be the training data and use a sparsity prior to perform what is now an overcomplete decomposition. This effectively means that in the above model we do not need to learn dictionaries, but instead we would set $P^{(a)}(f|z) \equiv \hat{S}_z^{(a)}(f)$ and $P^{(b)}(f|z) \equiv \hat{S}_z^{(b)}(f)$, where $\hat{S}^{(a)}$ and $\hat{S}^{(b)}$ are the normalized spectra we obtain from the training data for sources (a) and (b). That means that for each observed mixture point $P_t(f)$ in the normalized spectra simplex we would find one dictionary element from each of the two sources such that the observation lies on the line that connects these two elements. Note that this model is also able to resolve mixtures with more than two sources. That can be achieved in one of two ways. One is that we can model each source with its own dictionary and extend the model in equation (3) to have more than two terms. However for most cases we can define the two sources to be the target source (e.g. a singer) and a source model that encompasses all the other sources we are not interested in (e.g. all the accompanying instruments). The latter form, which we will make use of in this paper, is more practical since it involves a smaller number of dictionaries and a simplified model structure.

2.2. Unknown sources extension

In this paper we are interested in obtaining source information by having observed training examples of only the target source. This problem cannot be solved with the formulation in equation (3) since it needs to have training examples for all the sources in the observed mixture. A simple way to resolve this problem is to use something akin to the semi-supervised model in [4].

We will use the same model as the one we had in equation (3), only this time we will assume that the only dictionary elements that we know of will be the ones for the target source $P^{(a)}(f|z)$, whereas all the other sources will be unknown. This means that we would need to estimate not only the weights for both the target and the non-target sources, but also the dictionary elements of the non-target sources. These will be modeled as a single source using the dictionary elements $P^{(b)}(f|z)$. The only known parameters of the model are $P^{(a)}(f|z)$ which we set to be equal to the normalized spectra of the training data $\hat{S}_t^{(a)}(f)$. Just as before we can use the Expectation-Maximization algorithm to estimate $P^{(b)}(f|z)$, $P_t^{(a)}(z)$ and $P_t^{(b)}(z)$, $P_t(a)$ and $P_t(b)$ simultaneously. This will be an iterative procedure where the resulting estimation equations are:

$$P_t(z, s|f) = \frac{P_t(s)P_t^{(s)}(z)P^{(s)}(f|z)}{\sum_{s'} P_t(s') \sum_{z'} P_t^{(s')}(z')P^{(s')}(f|z')} \tag{4}$$

$$P^{(b)*}(f|z) = \sum_t P_t(z, b|f)P_t(f) \tag{5}$$

$$P_t^{(a)*}(z) = \sum_f P_t(z, a|f)P_t(f) \tag{6}$$

$$P_t^{(b)*}(z) = \sum_f P_t(z, b|f)P_t(f) \tag{7}$$

$$P_t(a) = \frac{\sum_z P_t^{(a)*}(z)}{\sum_z P_t^{(a)*}(z) + \sum_z P_t^{(b)*}(z)} \tag{8}$$

$$P_t(b) = \frac{\sum_z P_t^{(b)*}(z)}{\sum_z P_t^{(a)*}(z) + \sum_z P_t^{(b)*}(z)} \tag{9}$$

Where the * operator denotes an unnormalized parameter estimate and s is used as a source index. To obtain the current estimates of the parameters we make sure to normalize them all so that they properly sum to 1 in every iteration. Equation 4 corresponds to the E-step, whereas the others make up the M-step. The geometry of this process is shown in figure 2. Given the training data for the target source, for every observed mixture input spectrum we will infer a region that the plausible dictionary elements of the competing sources will lie. This subspace will be defined by the two lines with the greatest possible angle between them, that connect two of the dictionary elements with the observed mixture point. This is because of the geometric constraint that the mixture of two points in this space lie on the line defined by these points. The union of all of these areas as inferred from multiple mixture points will define the space where the dictionary elements for the competing sources lie.

2.3. Estimating source and pitch probabilities

We will now focus on how we could use this kind of decomposition to infer the presence of a source as well as its pitch. A convenient feature of the current model is that it is using training data directly as dictionary elements. Since these dictionary elements are then used to explain the mixture, we can use prior tagging information from the training data to infer semantic information about the mixture.

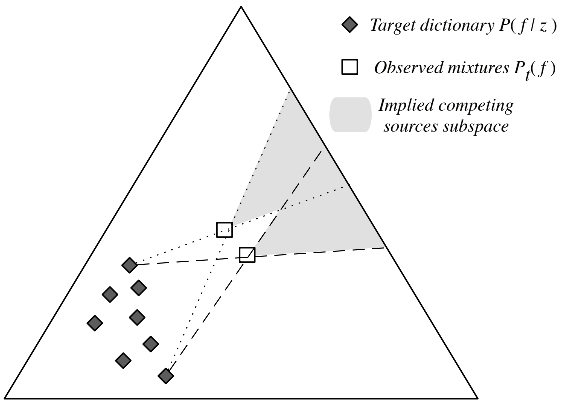

Target dictionary P(f | z)
Observed mixtures P_t(f)
Implied competing sources subspace

Fig. 2. *Inferring a source's subspace given a target source and two mixture points. Using the most disparate lines that connect the target dictionary elements with each observed point we define an area where the competing sources lie.*

For the purposes of this paper we care about two specific pieces of information, the energy of a source and its pitch. We can easily infer the energy of a source by using that source's prior (in the target's case, $P_t(a)$). In order to get an estimate of the pitch of that source we will have to use some a priori semantic tagging. Recall that in order to construct the target dictionary $P^{(a)}(f|z)$ we use the normalized spectra from representative recordings of that source. These recordings, not being mixtures, can be automatically pitch-tagged so that for each dictionary element can have an associated pitch value to it. After analysis of a mixture we will obtain the set of priors $P_t(a)$ and weights $P_t^{(a)}(z)$, which we can then combine to form an estimate of pitch across time. We do so by forming the distribution:

$$P_t^{(a)}(q) = \sum_{\{z: p^{(a)}(z)=q\}} P_t^{(a)}(z) \qquad (10)$$

where $p^{(a)}(z)$ is the estimated pitch value associated with the dictionary element $P^{(a)}(f|z)$, and $P_t^{(a)}(q)$ denotes the probability that the target source has the pitch q at time t. The summation term essentially computes the sum of all the weights that are associated with each pitch value to derive a distribution for pitch.

2.4. Semantic continuity

In theory, the aforementioned approach would suffice for performing the analysis we set to do, however in practice one observes that the estimates of $P_t^{(a)}(z)$ are considerably noisy and that an insightful estimate of $P_t^{(a)}(q)$ is hard to obtain. In the past these issues have been addressed with the use of a sparsity prior applied on $P_t^{(a)}(z)$, but in our experience this process does not work as well in this scenario. We instead formulate a *semantic continuity* constraint, which as we will demonstrate later, produces sparse results with temporal smoothness constraints using a single constraint.

In the context of this paper we define semantic continuity as having a minimal change of $P_t^{(a)}(q)$ between successive time indices. This means that we expect to see sustained pitch values, which is expected in musical signals, but that we also don't expect to have large jumps in the tracked melodies (also largely true for most music). We define that constraint in the form of a transition matrix regulating the likelihood that after seeing activity in dictionary elements associated with a specific pitch that we will see in the next time step a jump to dictionary elements that are associated with any other pitch. As described we like to penalize large pitch jumps, hence we define the following transition matrix:

$$P(z_{t+1} = i | z_t = j) \propto e^{-||p^{(a)}(z=i) - p^{(a)}(z=j)||/\sigma} \qquad (11)$$

where $P(z_{t+1} = i|z_t = j)$ denotes the probability that $P_{t+1}^{(a)}(z = i)$ will be activated if $P_t^{(a)}(z = j)$ is active. For simplicity we omit the normalizing factor that ensures that $P(z_{t+1} = i|z_t = j)$ sums to 1. The two pitch values $p(z = i)$ and $p(z = j)$ are the pitch tags associated with the two dictionary elements $P^{(a)}(f|z = i)$ and $P^{(a)}(f|z = j)$. The form of this matrix imposes an increased likelihood that in future estimates we will see more activity from dictionary elements that are associated to a close pitch to the current ones. The constant σ regulates how important the pitch distance is in constructing that matrix.

Having formed the transition matrix we now want to incorporate it in the learning process. We will, as shown above, estimate the weights $P_t^{(a)}(z)$ in each iteration, but in addition to that we will manipulate these estimates to impose the transition matrix structure. To do so we simply perform a forward-backward pass over the intermediate estimates and then normalize them.

For every estimated weights distribution $P_t^{(a)}(z)$ we have an expectation that it should be proportional to $\sum_{z_t} P(z_{t+1}|z_t)P_t^{(a)}(z_t)$. This will be of course different from the estimate we get in the M-step, therefore we use an extra processing step to impose the expected structure on the current estimate. To do so we define the forward and backward terms that represent our expected estimates given a forward and a backward pass through $P_t^{(a)}(z)$:

$$F_{t+1}(z) = \sum_{z_t} P(z_{t+1}|z_t)P_t^{(a)}(z) \qquad (12)$$

$$B_t(z) = \sum_{z_{t+1}} P(z_{t+1}|z_t)P_{t+1}^{(a)}(z) \qquad (13)$$

And then estimate the final value of $P_t^{(a)}(z)$ as:

$$P_t^{(a)}(z) = \frac{P_t^{(a)*}(z)(C + F_t(z) + B_t(z))}{\sum_z P_t^{(a)*}(z)(C + F_t(z) + B_t(z))} \qquad (14)$$

where $P_t^{(a)*}(z)$ is the estimate of $P_t^{(a)}(z)$ using the rule in equation (6), and C is a parameter that controls the influence of the transition matrix. As C tends to infinity, the effect of the forward and backward terms becomes negligible, whereas as C tends to 0 we tend to modulate the estimated $P_t^{(a)}(z)$ by the predictions of these two terms, thereby imposing the expected structure.

In practice we would also like to impose transition likelihoods for the non-target sources, especially as they relate to the target. To that end we use the above formulation to define a transition matrix that applies to all the dictionary elements. We can think of that matrix as being comprised out of four parts. One part will be a section that looks like the matrix in equation (11), this is the part that will regulate the transitions between the dictionary elements of the target. Likewise there will be a region that regulates the transition between the dictionary elements of the non-target sources. Since we do not have any requirements about these components we set all of these transition likelihoods to be equiprobable. The remaining two sections will regulate the transition between the target elements and the non-target elements and vice-versa. We set the transitions from non-target elements to target elements to be zero, since we don't want the structure of the target weights to be perturbed by the estimates of the non-target sources. On the other hand we set the transitions from the target elements to the non-target elements to be a constant non-zero value so that we encourage more of the use of the non-target components in order to obtain a sparser representation for the target. Applying these constraints involves a trivial generalization of the above process.

3. EXPERIMENTS

To demonstrate the use of this approach we construct the following experiment. The mixture that we wish to analyze was the song "Message in a Bottle" by the Police. The target source was the lead vocal line by Sting. In order to train our system to focus on the target source we used various recordings of Sting singing without any accompaniment. All audio recordings used a sample rate of 22,050 Hz. We then pitch tracked the training data and constructed the target source dictionary $P^{(a)}(f|z)$. The frequency transform we used is the DFT with a window of 1024pt and a hop size of 256pt. In order to have a more focused dictionary we discarded the dictionary elements that were not pitched, or corresponded to parts with low energy. This resulted in a set of 1228 dictionary elements for Sting's voice. We used four times as many components to describe all the competing sources, and employed the proposed analysis method to estimate all the required parameters. We run the experiment twice, once with $C = \infty$ (therefore ignoring semantic continuity) and once with $C = 0.0015$ and $\sigma = 10$. The transition probability from target to non-target components was set to 0.5.

The results of a small segment of this analysis are shown in figure 3. In all of these plots we display the pitch probability multiplied by the target prior, i.e. $P_t(a)P_t^{(a)}(q)$, and thus obtain a sense of when the target was active and what the most likely pitch was. The top plot shows the ground truth of a roughly 6 second singing segment. The middle plot shows the obtained estimate when using the plain model of equation (3). In addition to our estimate of $P_t(a)P_t^{(a)}(q)$ we also plot the expected pitch using the formula:

$$\hat{p}_t = \sum_z P_t^{(a)}(z)p^{(a)}(z) \tag{15}$$

For regions where $P_t^{(a)}(z)$ was under the 50th percentile of its values we assumed that the source is inactive and that there is no pitch. As is clearly evident the resulting output is not easily interpretable, the expected pitch values are wrong, and it contains significant energy at points where there is no singing taking place. The bottom plot shows the results when we use the semantic continuity constraint. It is easy to see that the resulting estimates are very close to the ground truth, and that they result in robust pitch estimates. The use of the proposed constraint succeeded in offloading irrelevant energy to the non-target components, and also acted as a sparsity regularizer.

4. CONCLUSIONS

In this paper we presented a system that can learn to follow the energy and pitch of a target source in a mixture after being shown examples of that source to use as a reference. In order to achieve that goal we used a probabilistic decomposition that made use of a semantic continuity constraint. The use of that constraint organized the resulting estimates of the energy and pitch of the target source in such a way so that one can easily infer these parameters. We demonstrated the results of this method by applying it on a complex real-world mixture.

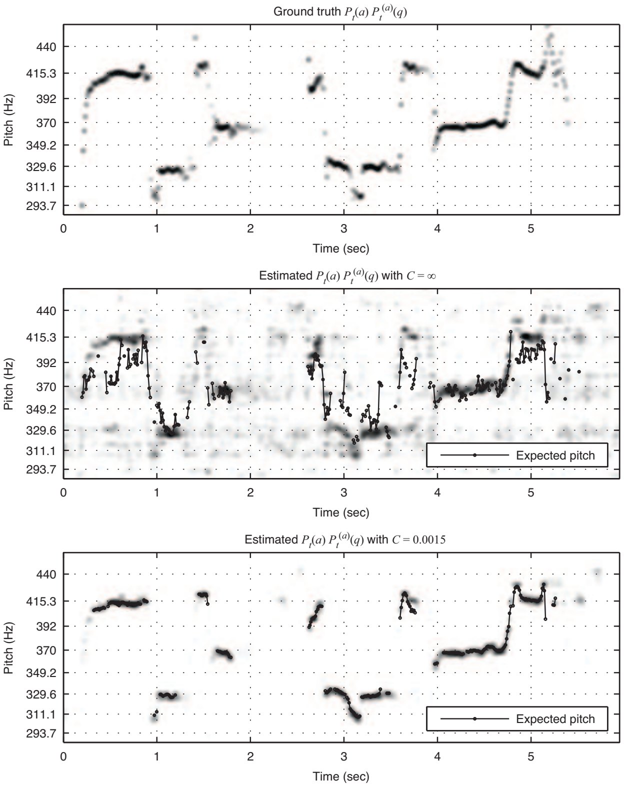


Fig. 3. *The pitch/enegy distributions for a segment of the song "Message in a Bottle". The top plot shows the true distribution of the singer's voice. The middle plot shows its estimates if we do not use the semantic continuity constraint, and the bottom plot shows the results when using that constraint. The lines in the two bottom plots show the expected pitch for each time point as estimated from these distributions. For presentation purposes the shown distributions have been slightly blurred so that point probabilities are become visible.*

5. REFERENCES

[1] Smaragdis, P. 2011 Polyphonic Pitch Tracking by Example, in proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY 2011.

[2] Shashanka, M.V.S. 2007. Latent Variable Framework for Modeling and Separating Single Channel Acoustic Sources. Department of Cognitive and Neural Systems, Boston University, August 2007.

[3] Smaragdis, P., M. Shashanka, and B. Raj. 2009. A sparse non-parametric approach for single channel separation of known sounds. In Neural Information Processing Systems. Vancouver, BC, Canada. December 2009.

[4] Smaragdis, P. Raj, B. and Shashanka, M.V. 2007. Supervised and Semi-Supervised Separation of Sounds from Single-Channel Mixtures. In proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation. London, UK. September 2007.