APPROXIMATE NEAREST-SUBSPACE REPRESENTATIONS FOR SOUND MIXTURES

Paris Smaragdis

University of Illinois at Urbana-Champaign, USA
Adobe Systems Inc.

ABSTRACT

In this paper we present a novel approach to describe sound mixtures which is based on a geometric viewpoint. In this approach we extend the idea of a nearest-neighbor representation to address the case of superimposed sources. We show that in order to account for mixing effects we need to perform a search for nearest-subspaces, as opposed to nearest-neighbors. In order to reduce the excessive computational complexity of this search we present an efficient algorithm to solve this problem which amounts to a sparse coding approach. We demonstrate the efficacy of this algorithm by using it to separate mixtures of speech.

Index Terms— Sound mixtures, separation, audio, speech

1. INTRODUCTION

Models of sounds have traditionally been parametric. Representations such as source-filter models, sinusoidal analyses, and statistical representations like Gaussian mixtures and hidden Markov models have been used successfully for many audio applications for years. Lately however, there has been a conceptual switch in the world of data processing that emphasizes simpler models coupled with larger training data sets. Such approaches forgo complex modeling and instead rely on large training data in order to boost their performance. In this paper we will explore this idea in the form of a simple non-parametric model, especially as it applies to mixtures of sounds. We will show that instead of using compact source models that try to describe sounds in a mixture, we can instead selectively use parts of training data to put them together and explain the mixing process. This approach allows us to distance ourselves from complex learning processes and their complications (overfitting, etc), and provides a simple description that can be semantically very powerful.

2. GEOMETRY OF SOUND MIXTURES

2.1. The space of normalized spectra

As a starting representation of audio we will consider the magnitude spectrogram form. For a given sound $x(t)$ its corresponding magnitude spectrogram will be notated as $X_t(f)$ and will represent the amount of signal energy at time t and frequency f. For the purposes of this paper we will not consider the temporal dynamics and instead examine only one spectral frame at a time. This allows us to instead focus on the *spectral composition* of each frame, as opposed to the absolute energy that $X_t(f)$ represents. The spectral composition of each frame will be represented by a probability vector that describes how energy is distributed across frequencies. This can be easily computed as:

$$\hat{X}_t(f) = \frac{X_t(f)}{\sum_{f'} X_t(f')}. \quad (1)$$

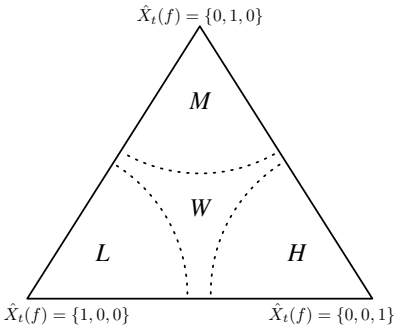

Fig. 1. *The simplex of three-frequency spectra. The vertices correspond to only one frequency being active, whereas the center point W, represents the point where all frequencies are equally active. Region L represents low-frequency spectra, region M mid-frequency spectra and region H high-frequency spectra. Other points in-between are able to express any possible spectral composition for a three-frequency sound.*

By doing so we effectively remove any information about the instantaneous energy of an analyzed sound and instead focus on its timbral qualities at any point in time. By representing sound this way we implicitly define a constrained space in which the proposed feature vectors live in. Because of the property that all elements of this representation sum to unity, all extracted vectors $\hat{X}_t(f)$ will be constrained to lie on a simplex. The dimensionality of this space will be equal to the overall number of frequencies minus one. A simplified case is illustrated in figure 1, where a 3-frequency space is shown.

2.2. Mixture geometry

In order to consider mixtures we will make the assumption that when we have sounds that mix, their magnitude spectra superimpose linearly. Although this is not an exact consequence, it is an assumption that has been used frequently by the source separation community and is generally accepted as being approximately true. Under this assumption when we have the mixture of two sounds $z(t) = x(t)+y(t)$, we will expect that its magnitude spectrogram will equal the sum of the magnitude spectrograms of the two sounds individually, i.e.:

$$Z_t(f) = X_t(f) + Y_t(f). \quad (2)$$

Where $X_t(f)$, $Y_t(f)$ and $Z_t(f)$ are magnitude spectral frames of $x(t)$, $y(t)$ and $z(t)$ respectively, that represent energy at time t and frequency f. Just as before we only want to consider the spectral composition of each source and represent each spectral vector as a frequency distribution. We obtain that by normalizing the signal

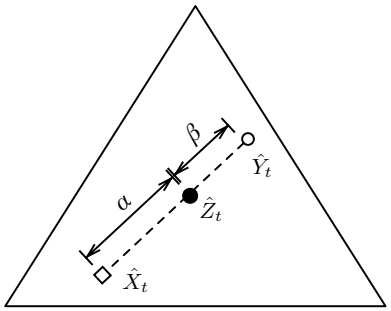

Fig. 2. *The result of mixing two magnitude spectra compositions through $\hat{Z}_t(f) = \alpha\hat{X}_t(f) + \beta\hat{Y}_t(f)$. The two source points are denoted by the diamond and the circle in the spectrum composition simplex. Their mixture, denoted by a bold dot, will be constrained to lie on the subspace (dotted line in this case), that connects the two sources. The position of the mixture point along that subspace will be determined by the energy ratio of the two sources as coded in the parameters α and β.*

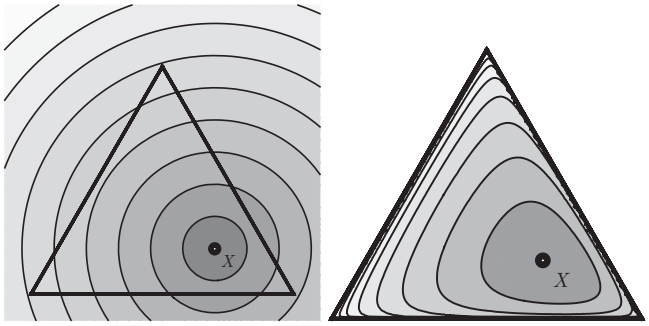

Fig. 3. *Comparison of two similarity functions in the spectral composition simplex. Lighter colors denote less similarity as compared to the reference point X. On the left we show the similarity pattern that arises when we assume Gaussian distributed data and on the left when we assume Dirichlet distributed data. Note how the Dirichlet assumption respects the geometry of the space and produces an appropriate similarity measure.*

spectra by:

$$\begin{aligned}\hat{Z}_t(f) &= \frac{Z_t(f)}{\sum_f Z_t(f)} = \frac{X_t(f) + Y_t(f)}{\sum_{f'} X_t(f') + Y_t(f')} \\ &= \alpha\hat{X}_t(f) + \beta\hat{Y}_t(f),\end{aligned}$$

where the hat operator denotes normalized magnitude spectra. If we visualize the result of this operation in the spectral composition simplex that we defined in the previous section it will look as shown in figure 2.

3. NON-PARAMETRIC MODELS FOR SOUNDS

3.1. Single-source model

Given this framework we will now construct a simple non-parametric model that can be used to approximate an input sound represented by a series of composition spectra $\hat{X}_t$. In order to do so we will assume that we have a set of training data $\hat{\mathcal{X}}$ that represents the kind of sound that we wish to model. Having that, we can approximate each input frame $\hat{X}_t$ by its nearest-neighbor $\hat{\mathcal{X}}_\tau$, such that:

$$\tau = \underset{\tau}{\operatorname{argmax}}\, d(\hat{X}_t|\hat{\mathcal{X}}_\tau), \tag{3}$$

for a given similarity function $d()$. Finding which training frame $\hat{\mathcal{X}}_\tau$ is the closest to the current frame $\hat{X}_t$ involves a straightforward nearest-neighbor search, which scales linearly in complexity with the size of the training data. Once the approximation for all vectors $\hat{X}_t$ is found, we scale these vectors by the original gain of the corresponding spectrum at that time and use the original signal phase to invert the magnitude spectrogram approximation to the time domain. This particular way of modeling sound spectra constitutes a simple non-parametric model very similar to vector quantization approaches, only this time we employ the entire training data as opposed to using a compact codebook.

3.2. Measuring similarity

One complication that arises in this representation comes from the definition of the similarity function $d()$. One might be tempted to use a form based on the Euclidean distance, but as we show in this section this is an inappropriate measure. The use of Euclidean distance inside the spectral composition simplex implies that we are making a Gaussian distribution assumption for the spectral composition frames. This means that the similarity function itself is expressed as a Gaussian likelihood:

$$d(X|Y) \propto \mathcal{N}(X; \mu = Y, \Sigma = \mathbf{I}), \tag{4}$$

where $\mathcal{N}(X; \mu, \Sigma)$ denotes the Gaussian likelihood of an input X with a mean of μ and a covariance of Σ. This similarity measure and its relation to the spectral composition simplex is shown in the left subplot of figure 3. Note its inappropriateness since it allows for points outside of the simplex and it does not take into account the special shape of this space.

A more appropriate distribution in this space is the Dirichlet distribution [2], which is explicitly defined on a simplex and is used to describe compositional data like the ones we have. Under this model the similarity function becomes:

$$d(X|Y) \propto \mathcal{D}(Y; X + 1), \tag{5}$$

where $\mathcal{D}(Y; X)$ denotes the Dirichlet distribution with an input Y and hyper-parameters X. The mode of this distribution as defined in the above equation will be at X and it will measure the likelihood of an input X being Y. If we examine the log likelihood of this model we can see that it resolves to the following form:

$$d(X|Y) \propto \sum x_i \log(y_i), \tag{6}$$

that is the formula for cross-entropy, which from information theory we know to be an appropriate measure to compare two probability vectors. As shown in the right subplot of figure 3, this similarity is bounded by the spectral composition simplex and is appropriately shaped to account for its shape. For the rest of this paper we will be considering this to be the default similarity measure to be used.

3.3. The sound mixtures model

The non-parametric model shown in the previous section is a straightforward process and can model single sounds very well. We will now examine the same idea as it applies on mixtures of

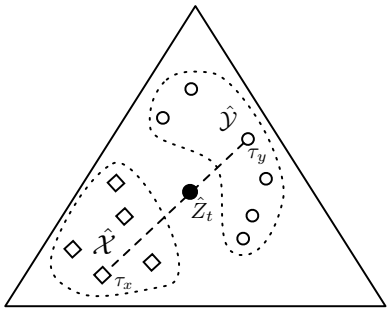

Fig. 4. *An illustration of the nearest-subspace approach for modeling sound mixtures. The two non-parametric models of the mixed sounds are represented by the clusters $\hat{\mathcal{X}}$ (diamonds) and $\hat{\mathcal{Y}}$ (circles). Given an input frame $\hat{Z}_t$ (dark dot) we want to identify the indices τ_x and τ_y that identify two training points, one from each class, that define a subspace that passes the closest through it.*

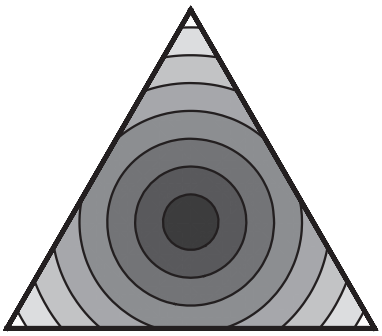

Fig. 5. *The ℓ_2-norm regularizing term as applied to compositional data. This term is maximized (lighter color) by points that lie on the vertices. Using this as a regularizing term to be maximized results in a sparse weights estimate.*

sounds. For simplicity we start with a mixture consisting of two sources: $z(t) = x(t) + y(t)$. Just like before we will have a training data set, this time consisting of spectral composition vectors of the constituent sources and not of the mixture directly. The training data will therefore consist of two parts, $\hat{\mathcal{X}}$ and $\hat{\mathcal{Y}}$ which will represent the classes of sounds for sources $x(t)$ and $y(t)$ respectively. As should be evident, the nearest-neighbor approach will not apply in this case. We will not necessarily be able to approximate each mixture input $\hat{Z}_t$ using one element from either $\hat{\mathcal{X}}$ or $\hat{\mathcal{Y}}$. Instead we would have to model each $\hat{Z}_t$ as a weighted summation of one element from $\hat{\mathcal{X}}$ and one from $\hat{\mathcal{Y}}$, i.e.:

$$\begin{aligned} \hat{Z}_t &\approx \alpha\hat{\mathcal{X}}_{\tau_\S} + \beta\hat{\mathcal{Y}}_{\tau_\dagger} \\ \{\alpha, \beta, \tau_x, \tau_y\} &= \underset{\alpha,\beta,\tau_x,\tau_y}{\operatorname{argmax}}\, d(\hat{Z}_t|\alpha\hat{\mathcal{X}}_{\tau_x} + \beta\hat{\mathcal{Y}}_{\tau_y}), \end{aligned}$$

where the similarity function $d()$ is cross-entropy as discussed in the previous section. After solving this problem we will obtain a nearest-neighbor approximation of each source independently and an indication of their relative amplitude. This problem amounts to a *nearest-subspace* search. Unlike the single-source case where we search for a nearest-point, we now search for two points, one from each source dictionary, that form a subspace that passes the closest to our input. This is illustrated in figure 4.

3.4. A direct search approach

Solving the nearest-subspace problem as defined in the previous section is a computationally intensive operation with a prohibitive complexity. Consider a small-sized problem using 60 seconds of training data for each source and spectra of about 2,000 dimensions, which results in about 7,500 training vectors for each source. To find the dictionary indices for a single input frame we would need to perform $7,500^2$ searches. Each search itself would involve multiple evaluations of cross-entropy and a non-linear optimization to find the parameters α and β. For a 10 second mixture that would amount to more than 7 billion optimization problems. Medium-sized training sets of the order of 10 minutes will require the solution of many trillions of separate optimizations. The rapidly increasing complexity with respect to the amount of training data results in an unacceptable number of computations for our task even in toy simulations.

An alternative approach would be to map this problem to a nearest-neighbor search as shown in [1], however for the dimensionality and scale of our training data that would involve performing a nearest-neighbor search over a set of solutions which would be infeasible to store in memory (more than a petabyte for the small scale problem of 60 seconds of training data per source).

3.5. A regularization approach

In order to address this problem in an efficient manner we recast this search as a sparse coding problem. We attempt to maximize the cross-entropy between each input frame and the reconstruction model:

$$d\left(\hat{\mathcal{Z}}_t|\alpha\sum w_i\hat{\mathcal{X}}_i + \beta\sum v_i\hat{\mathcal{Y}}_i\right), \tag{7}$$

where w and v are weights vectors that contain only one non-zero element each. Because our spectral composition data is normalized to sum to one, the following constraints hold:

$$\begin{aligned} w_i, v_i, \alpha, \beta &\geq 0 \\ \sum w_i = \sum v_i = \alpha + \beta &= 1. \end{aligned}$$

Because of the above constraints, w and v will also themselves lie on a simplex. We will call that the weights simplex, where each vertex will represent a point from the training data set. Our goal now is to obtain estimates of w and v whose position on their respective weights simplex lies on a vertex. Such a solution would mean that only one value in these weight terms will be non-zero which is our desired outcome. In order to impose that constraint we need to perform a regularized optimization that pushes towards such a solution. A regularizing term that can enforce that behavior is that of maximizing the ℓ_2-norm of w and v. The effect of that term is shown in the weights simplex in figure 5. We see that the regularization term is maximized towards the vertices and that it would bias the final solution to be sparser. Note that usually ℓ_2 maximization will not result in a sparse solution, this is a special case due to the constraints that we use on the weight vectors. In order to compact the problem definition we can consolidate all the weights and the α, β parameters in a single weight vector h that acts on all the training data $\hat{\mathcal{U}} = [\hat{\mathcal{X}}, \hat{\mathcal{Y}}]$ simultaneously:

$$d\left(\hat{\mathcal{Z}}_t|\sum h_i\hat{\mathcal{U}}_i\right), \tag{8}$$

where now we want only two of the elements of h to be non-zero, one in each segment that corresponds to a different source, and $\hat{\mathcal{U}}_i$

implies the selection of the ith column of $\hat{\mathcal{U}}$. Maximizing the expression in the above equation under the given constraints and the sparsity regularization term results in the iterative estimator:

$$\begin{aligned} r(f) &= \hat{Z}_t(f) / \sum h_i \hat{\mathcal{U}}(f)_i \\ t_i &= h_i \sum r(f) \hat{\mathcal{U}}(f)_i \\ t_i^* &= t_i + \mu \left(t_i^2 / \sum t_j^2 \right) \\ h_i^{new} &= t_i^* / \sum t_j^*, \end{aligned}$$

where $\mu \geq 0$ determines the strength of the regularization term. Upon convergence we keep only the largest element of h that corresponds to training data from each source, and by examining the relation between these two values we can easily compute α and β. This approach is computationally very efficient and can resolve 10 second mixtures with 60 seconds of training data per source in less than 30 seconds on a typical workstation system.

4. EXPERIMENTS ON MIXTURES

We used a set of speech recordings to evaluate the ability of this model to represent mixtures. The recordings were from the speech separation challenge [3] and involved short spoken sentences from various speakers. In our experiments we used two speakers at a time. We used all but one sentence from each speaker to construct the training data sets $\hat{\mathcal{X}}$ and $\hat{\mathcal{Y}}$ and we mixed the two remaining sentences as a 0dB mixture and obtained $\hat{\mathcal{Z}}$. We used a DFT size of 2048, a frame overlap of 256 samples and a Hann window in order to obtain the speaker dependent spectral frames. We additionally warped the frequency axis of the spectra so as to obtain a constant-Q-like transform and give more importance to the low frequencies. These spectral frames were subsequently normalized to obtain the compositional spectra $\hat{\mathcal{X}}$ $\hat{\mathcal{Y}}$ and $\hat{\mathcal{Z}}$.

In order to evaluate the ability to model mixtures we attempted to separate the mixed signals using the training data information. This was done by identifying the two training data points, one from each source, that define the nearest subspace to each mixture point, and then reconstructing each source independently using only them. In order to obtain each source estimate we set the instantaneous composition spectrum of each source to each of the two optimal points $\alpha\hat{\mathcal{X}}_{\tau_x}$ and $\beta\hat{\mathcal{Y}}_{\tau_y}$ for that time and then scale them using the instantaneous gain of the mixture at that point. This provides the two spectral estimates for the two sources which we can then convert back to the time domain by using the phase of the mixture spectrogram.

An example case is shown in figure 6. For multiple such experiments the results according to the evaluation metrics in [4] were around 20dB for Signal to Interference Ratio and around 5dB for the Signal to Distortion or Artifacts Ratios. The strong SIR value shows that the sources are well separated from each other (something we can subjectively confirm with listening tests), and the weak SDR and SAR are a result of the fact that we do not attempt an optimal reconstruction of the source, but instead we find an approximation.

5. CONCLUSIONS

In this paper we present a new way to decompose mixtures of sounds by using verbatim parts from training data of the constituent source classes and treating mixture analysis as a nearest-subspace problem. This way of explaining mixtures, given proper training data, is primarily affected by one factor: how much the training data from different sources overlaps in the frequency composition simplex. It is in principle invariant to the number of sources since any mixture problem can be seen as a binary segmentation between a target and an interference (the only complication of having many sources being the increased probability of the target and the interference overlapping in the frequency composition simplex). Other factors such as reverberation and propagation effects are also not an issue as long as they don't color the sources enough to significantly change their spectral composition (not an observed problem in general).

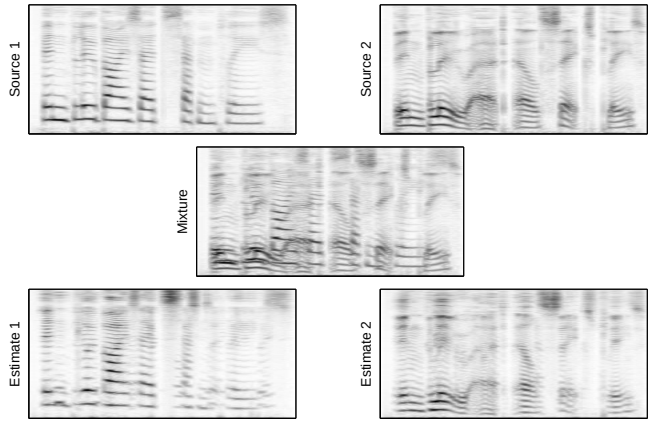

Fig. 6. *A result from a single source separation experiment. The top plots show two sentences spoken by two different speakers. The middle plot shows their observed mixture, and the bottom plots the estimates sources given training data from each speaker.*

Although source separation was the only context we presented this approach in, we have to stress that the scope of this model is wider-ranging. A great advantage of this representation is that the source estimates are being explained using verbatim parts of the training data. That allows us to use existing semantic information from the training data (e.g. phonetic labels, or note transcriptions), to map that information on a mixture and thus perform recognition and indexing operations on mixtures without having to separate any sources. This is a departure from the usual source-separation-to-analysis model and one which we feel will become increasingly predominant in the future.

6. ACKNOWLEDGEMENT

The author wishes to thank Prof. Bhiksha Raj of Carnegie Mellon University for his helpful insights during the development of this approach.

7. REFERENCES

[1] Basri, R., T. Hassner and L. Zelnik-Manor. 2007. Approximate Nearest Subspace Search with Applications to Pattern Recognition, CVPR'07.

[2] Minka, T. 2000. Estimating a Dirichlet distribution. Technical report, MIT Media Laboratory.

[3] `http://www.dcs.shef.ac.uk/~martin/SpeechSeparationChallenge.htm`

[4] Févotte, C., R. Gribonval and E. Vincent. 2005. BSS EVAL Toolbox User Guide, IRISA Technical Report 1706, Rennes, France, April 2005.