Sound Recognition in Mixtures

Juhan Nam[1*], Gautham J. Mysore[2], and Paris Smaragdis[2,3]

[1] Center for Computer Research in Music and Acoustics, Stanford University,
[2] Advanced Technology Labs, Adobe Systems Inc.,
[3] University of Illinois at Urbana-Champaign

Abstract. In this paper, we describe a method for recognizing sound sources in a mixture. While many audio-based content analysis methods focus on detecting or classifying target sounds in a discriminative manner, we approach this as a regression problem, in which we estimate the relative proportions of sound sources in the given mixture. Using certain source separation ideas, we directly estimate these proportions from the mixture without actually separating the sources. We also introduce a method for learning a transition matrix to temporally constrain the problem. We demonstrate the proposed method on a mixture of five classes of sounds and show that it is quite effective in correctly estimating the relative proportions of the sounds in the mixture.

1 Introduction

Nowadays, a huge volume of multimedia content is available and is rapidly increasing over broadband networks. While the content is usually managed or searched using manually annotated text or collaborative information from users, there has been increasing efforts to automatically analyze the content and find relevant information. In particular, some researchers have tried to analyze the content by recognizing sounds in the video because information in the audio domain is crucial for certain tasks, such as sports highlight detection and event detection in surveillance systems [1]. Moreover, audio data has a relatively low bandwidth.

The majority of audio-based content analysis methods focus on detecting a target source or classifying sound classes in a discriminative manner [2, 3]. Although they are successful in some detection or classification tasks, such discriminative approaches have a limitation in that most real-world sounds are mixtures of multiple sources. It is therefore useful to be able to simultaneously model multiple sources for various applications such as searching for certain scenes in a film soundtrack. For example, if we want to search for a scene with a specific actor in which a car is passing by and background music is present, it would be useful to model each of these sources.

In this paper, we propose a generative approach, which models a mixture sound as multiple single sources and estimates the relative proportion of each

* This work was performed while interning at Adobe Systems Inc.

source. Our method is based on probabilistic latent component analysis (PLCA) [4], which is a variant of non-negative matrix factorization (NMF). PLCA has been widely used as a way of modeling sounds in the spectral domain because of the interpretable decomposition and extensible capability as a probabilistic model. We first formalize our problem using a PLCA-based approach and then we propose an improved model which takes temporal characteristics of each source into account. Lastly, we evaluate our method with a dataset and discuss the results.

2 Proposed Method

The basic methodology that we follow is that of supervised source separation using PLCA [5]. For each source, we estimate a dictionary of basis elements from isolated training data of that source. Then, given a mixture, we estimate a set of mixture weights. Using these weights, it is possible to separate the sources (typical PLCA-based supervised source separation). However, without actually separating the sources, we estimate the relative proportion of each source in the mixture. Since we bypass the actual separation process, we can do certain things to improve sound recognition performance even when it does not improve source separation performance. Specifically, we choose the dictionary sizes based on sound recognition performance. Also, we impose a temporal continuity constraint that helps this performance but could introduce fairly heavy artifacts if we were to actually separate the sources. Note that we refer to a source as a general class of sounds, such as speech, music and other environmental sounds.

2.1 Basic Model

PLCA is an additive latent variable model that is used to decompose audio spectrograms [4]. An asymmetric version of PLCA models each time frame of a spectrogram as a linear combination of dictionary elements as follows:

$$X(f,t) \approx \gamma \sum_{z} P(f|z) P_t(z) \tag{1}$$

where $X(f,t)$ is the audio spectrogram, z is a latent variable, each $P(f|z)$ is a dictionary element, $P_t(z)$ is a distribution of weights at time frame t, and γ is a constant scaling factor. All distributions are discrete. Given $X(f,t)$, we can estimate the parameters of $P(f|z)$ and $P_t(z)$ using the EM algorithm.

We model single sound sources and their mixtures using PLCA. We first compute the spectrogram $X_s(f,t)$ given isolated training data of source s. We then use Eq. 1 to estimate a set of dictionary elements and weights that correspond to that source. In the basic model, we assume that a single source is characterized by the dictionary elements. Therefore, we retain the dictionary elements while discarding the weights. Using the dictionary elements from each single source, we build a larger dictionary to represent a mixture spectrogram. This is formed by simply concatenating the dictionaries of the individual sources. Thus, if we have a spectrogram $X_M(f,t)$ that is a mixture of two sources, we model it as follows[4]:

$$X_M(f,t) \approx \gamma \sum_{z \in \{\mathbf{z_{s_1}}, \mathbf{z_{s_2}}\}} P(f|z) P_t(z) \tag{2}$$

where $\mathbf{z_{s_1}}$ and $\mathbf{z_{s_2}}$ represent the dictionary elements that belong to source 1 and source 2 respectively. Since the dictionary elements of both sources are already known, we keep them fixed and simply estimate the weights $P_t(z)$ at each time frame using the EM algorithm. The weights tell us the relative proportion of each dictionary element in the mixture. It is therefore intuitive that the sum of the weights that correspond to a given source, will give us the proportion of that source present in the mixture. Accordingly, we compute the relative proportions of the sources at each time frame by simply summing the corresponding weights as follows:

$$r_t(s_1) = \sum_{z \in \mathbf{z_{s_1}}} P_t(z) \tag{3}$$

$$r_t(s_2) = \sum_{z \in \mathbf{z_{s_2}}} P_t(z) \tag{4}$$

2.2 Modeling Temporal Dependencies

When we learn a model for a single source from isolated training data of that source, we obtain a dictionary of basis elements and a set of weights. In the previous subsection, we discarded the weights as they simply tell us how to fit the dictionary to that specific instance of training data. This is usually the practice when performing NMF or PLCA based supervised source separation [5].

Although the weights are specific to the training data, they do contain certain information that is more generally applicable. One such piece of information is temporal dependencies amongst dictionary elements. For example, if a dictionary element is quite active in one time frame, it is usually likely to be quite active in the following time frame as well. However, there are usually more such dependencies present such as things like a high presence of dictionary element m in time frame t usually followed by a high presence of dictionary element n in time frame $t+1$. Using the weights of adjacent time frames, we can infer this information. For time frames t and $t+1$ of source s, we can compute this dependency as follows:

$$\phi_s(z_t, z_{t+1}) = P(z_t) P(z_{t+1}), \ \forall z \in \mathbf{z_s}.$$

This gives us the affinity of every dictionary element to every other dictionary element in two adjacent time frames. If we average this value over all time frames

[4] It is straightforward to extend this to more sources.

and normalize, we obtain a set of conditional probability distributions that serve as a transition matrix as follows:

$$P_s(z_{t+1}|z_t) = \frac{\sum_{t=1}^{T-1} \phi_s(z_t, z_{t+1})}{\sum_{z_{t+1}} \sum_{t=1}^{T-1} \phi_s(z_t, z_{t+1})}.$$

When we learn dictionaries from isolated training data, we compute such a transition matrix for each source. As a result, our model for each source consists of a dictionary and a transition matrix.

Given a mixture, our method of estimating weights should be accordingly changed to make use of the transition matrix. First, we should have a joint transition matrix $P(z_{t+1}|z_t)$ that corresponds to the concatenated dictionaries. Since we assume that the activity of the dictionary elements in one dictionary are independent of those in other dictionaries, we construct the joint transition matrix by diagonalizing individual transition matrices. For example, if we have two sound sources and two corresponding transition matrices $T1$ and $T2$, the joint transition matrix is formed as $T = [T_1\ 0; 0\ T_2]$(using Matlab notation).

Once we obtain the concatenated dictionary and transition matrix, we move on to the actual sound recognition stage. Given the mixture, we first estimate the weights $P_t(z)$ as described in the previous subsection. We call this our initial weights estimate $P_t^{(i)}(z)$. Using these estimates, we obtain a new estimate of the weights that is more consistent with the dependencies that are implied by the joint transition matrix[5]. We do this by first computing re-weighting terms in the forward and backward directions to impose the joint transition matrix in both directions:

$$F_{t+1}(z) = \sum_{z_t} P(z_{t+1}|z_t) P_t^{(i)}(z).$$

$$B_t(z) = \sum_{z_{t+1}} P(z_{t+1}|z_t) P_{t+1}^{(i)}(z).$$

Using the above terms, we perform the re-weighting and normalize as follows to get our final estimate of the weights:

$$P_t(z) = \frac{P_t^{(i)}(z)\,(C + F_t(z) + B_t(z))}{\sum_z P_t^{(i)}(z)\,(C + F_t(z) + B_t(z))},$$

where C is a parameter that controls the influence of the joint transition matrix. As C tends to infinity, the effect of the forward and backward re-weighting terms becomes negligible, whereas as C tends to 0 we tend to modulate the estimated $P_t^{(i)}(z)$ by the predictions of these two terms, thereby imposing the expected structure. This re-weighting is performed after the M step in every EM iteration. Finally, we obtain the relative proportions of single sources at each time frame by simply summing the corresponding weights as in Eq. 3 and 4.

[5] This is analogous to smoothing an estimated time series with a moving average filter if we believe that the time series is slowly varying.

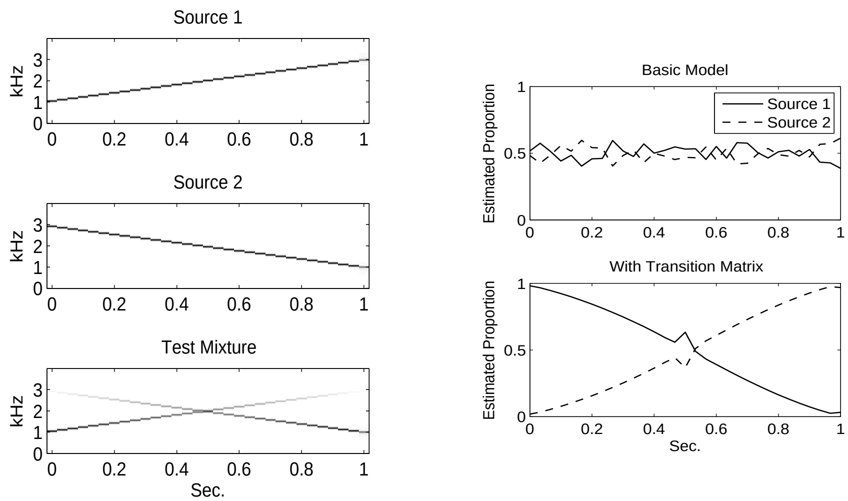


Fig. 1: A toy example: training sources are given as chirps that have frequencies changing in opposite directions and the test mixture is created by linearly cross-fading the two chirps. The basic model fails to discriminate the two sources whereas the model using the transition matrix successfully estimates the cross-fading curves, although there is a little glitch in the intersection.

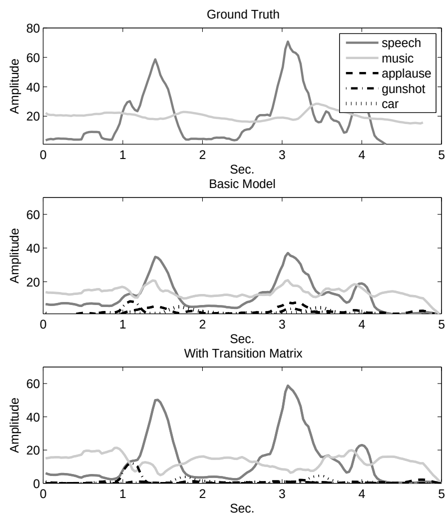
Fig. 1 illustrates the effect of re-weighing by the transition matrix. In the example, two source signals are given as chirps that have frequencies changing in opposite directions and thus they produce the same dictionary but different transition matrices. The test signal is created by cross-fading the two chirps. The basic model estimates approximately the same proportions of the two sources because both dictionaries explain the mixture equally well at every time frame. On the other hand, the re-weighting using the transition matrix successfully estimates the cross-fading curves by filtering out weights inconsistent with temporal dependencies of each source.

3 Experimental Results

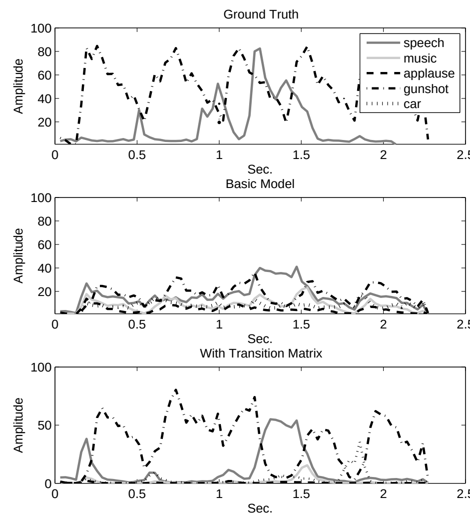
We evaluated the proposed method on five classes of sound sources–speech, music, applause, gun shot and car. We collected ten clips of sound files for each class. Speech and music files were extracted from movies, each about 25 seconds long. Other sound files were obtained from a sound effects library.[6] They have different lengths from less than one to five seconds. We resampled all sound files to 8kHz, and used a 64ms Hann window with 32ms overlap to compute the spectrograms.

In the training phase, we obtained a dictionary of elements and a transition matrix for each sound source. Since we are not separating the sources and

[6] www.sound-ideas.com



(a) speech and music



(b) speech and gun shot

Fig. 2: Estimated relative proportions for mixtures of two sources. For the purpose of visualization, instead of the relative proportions, we shows the amplitude envelopes obtained by multiplying the relative proportions to the sum of the magnitudes in that time frame ($\sum_f X(f,t)$) (an approximation to the mixture envelope). The top plots are the ground truth computed from individual sources. The middle and bottom plots show the results using the basic model and the improved model with the transition matrix.

therefore do not need a high-quality reconstruction, we chose a small number of dictionary elements (less than 15) for each sound class. One of difficulties that we encountered was choosing different combinations of dictionary sizes for single sound sources because if we consider all possible combinations of dictionary sizes, the number of possibilities exponentially grows.[7] Therefore, we constrain the number of possible combinations using some heuristics. For example, dictionary sizes of speech and music should be greater than those of other environmental sounds because speech and music generally have more variations in the training data. Under this idea, we chose ten sets of dictionary sizes. The maximum numbers of dictionary sizes were 12, 15, 5, 5, 8 for speech, music, applause, gunshot and car sounds, respectively, and the minimum numbers were 1 for all classes.

Fig. 2 shows an example in which the test sound is given as a mixture of two sources. For the mixture of speech and music sounds, both models recognize the two sources fairly well. However, in the basic model, separation between

[7] For example, if we consider three different dictionary sizes for each source, this number of possible combinations will be 3^5

Test sources	speech	music	applause	gun shot	car
basic model	0.37	0.45	0.20	0.76	0.41
with transition matrix	0.26	0.32	0.03	0.42	0.39

Table 1: Estimation errors for single test sources

speech and music is somewhat diluted and loud utterances of speech are partly explained by other sources, which are absent from the test sound. On the other hand, the model with the transition matrix shows better separation between speech and music and suppresses other sources more effectively. For the mixture of speech and gunshot sounds, the two models show more apparent difference in performance. The basic model completely fails to estimate the relative proportions as the gunshot sound is represented by many other sources, whereas the model with the transition matrix restores the original envelopes fairly well.

We performed a more formal evaluation using ten-fold cross-validation; we split the dataset into nine training files and one test file for each source at each validation stage. We estimated the relative proportions for single sources and mixtures of two and three sources from the test files. For the mixture sounds, we also adjusted the relative gains of the sources and separately performed the estimation.[8] To quantify the accuracy of the estimation, we computed the following metric:

$$\text{Estimation error} = \frac{1}{N}\sum_{s}\sum_{t}|r_t(s) - g_t(s)|, \tag{5}$$

where $r_t(s)$ is the estimated proportion from Eq. 3 and 4, $g_t(s)$ is the ground truth proportion and N is the number of time frames in the test file. We obtained the ground truth proportion from the ratio of envelopes between singles sources and the mixture at each time frame. The envelope was computed by summing the magnitudes in that time frame ($\sum_f X(f,t)$). We measured this metric only for active sources, that is, those in the test sound. Note that the ground truth proportion is 1 for single test sounds since no other sound is present in that case. Throughout the the cross-validation, we repeatedly performed testing with different sets of dictionary sizes and four re-weighting strengths ($C = 0.3$, 0.5, 0.7 and 1.0). Then, we computed the average of the estimation errors from the cross validations to find the best parameter sets.

Table 1 shows the results for the single test source. In the basic model, the significant proportion of the test sound is explained by dictionaries of other sources, particularly for gun shot sounds. However, the model with the transition matrix show significant improvement for most sounds. Table 2 and 3 shows the results for the mixtures of two and three sources. Although the improvements are slightly less than those in the single source case, the model with transition

[8] For the mixtures of two sources, the relative gains of the two sources were adjusted to be -12dB, -6dB, 0dB, 6dB and 12dB. For the mixtures of three sources, they were adjusted to be -6dB, 0dB, 6dB for each pair.

Test sources	speech/music	speech/gun shot	speech/applause	music/car
basic model	0.17 / 0.27	0.19 / 0.48	0.13 / 0.16	0.26 / 0.25
with transition matrix	0.15 / 0.21	0.15 / 0.34	0.13 / 0.12	0.21 / 0.26

Table 2: Estimation errors for mixtures of two sources

Test sources	speech/music/gun shot	speech/music/car
basic model	0.17 / 0.21 / 0.25	0.16 / 0.20 / 0.20
with transition matrix	0.15 / 0.18 / 0.25	0.15 / 0.17 / 0.21

Table 3: Estimation errors for mixtures of three sources

matrix generally outperform the basic model. Note that as we have more sources in the test sound, the estimation errors for individual sources become smaller because the relative proportions of single sources are also smaller.

4 Conclusions

In this paper we presented a method of estimating the relative proportions of single sources in sound mixtures. We first proposed a method of performing this estimation using standard PLCA. We then proposed a method to improve this estimation by accounting for temporal dependencies among dictionary elements. Our experiments on five classes of sound sources showed promising results, particularly with the model that considers temporal dependencies. Future work includes testing on a larger database and exploring more evaluation metrics.

References

1. Radhakrishnan, R., Xiong, Z., Otsuka, I. : A Content-Adaptive Analysis and Representation Framework for Audio Event Discovery from Unscripted Multimedia. EURASIP Journal on Applied Signal Processing, 1-24 (2006)
2. Li, Y., Dorai, C.: Instructional Video Content Analysis Using Audio Information. IEEE TASLP, VOL. 14, NO. 6, (2006)
3. Tran, H. D., Li, H.: Sound Event Recognition With Probabilistic Distance SVMs. IEEE TASLP, VOL. 19, NO. 6, (2011)
4. Smaragdis, P., Raj, B., Shashanka, M.: A probabilistic latent variable model for acoustic modeling. In Advances in models for acoustic processing, NIPS. (2006)
5. Smaragdis, P., Raj, B., Shashanka, M.: 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)