Deep clustering: Discriminative embeddings for segmentation and separation

John R. Hershey
MERL
Cambridge, MA
hershey@merl.com

Zhuo Chen
Columbia University
New York, NY
zc2204@columbia.edu

Jonathan Le Roux
MERL
Cambridge, MA
leroux@merl.com

Shinji Watanabe
MERL
Cambridge, MA
watanabe@merl.com

Abstract

We address the problem of acoustic source separation in a deep learning framework we call “deep clustering”. Rather than directly estimating signals or masking functions, we train a deep network to produce spectrogram embeddings that are discriminative for partition labels given in training data. Previous deep network approaches provide great advantages in terms of learning power and speed, but previously it has been unclear how to use them to separate signals in a class-independent way. In contrast, spectral clustering approaches are flexible with respect to the classes and number of items to be segmented, but it has been unclear how to leverage the learning power and speed of deep networks. To obtain the best of both worlds, we use an objective function that to train embeddings that yield a low-rank approximation to an ideal pairwise affinity matrix, in a class-independent way. This avoids the high cost of spectral factorization and instead produces compact clusters that are amenable to simple clustering methods. The segmentations are therefore implicitly encoded in the embeddings, and can be ”decoded” by clustering. Preliminary experiments show that the proposed method can separate speech: when trained on spectrogram features containing mixtures of two speakers, and tested on mixtures of a held-out set of speakers, it can infer masking functions that improve signal quality by around 6dB. We show that the model can generalize to three-speaker mixtures despite training only on two-speaker mixtures. The framework can be used without class labels, and therefore has the potential to be trained on a diverse set of sound types, and to generalize to novel sources. We hope that future work will lead to segmentation of arbitrary sounds, with extensions to microphone array methods as well as image segmentation and other domains.

1 Introduction

In real world perception, we are often confronted with the problem of selectively attending to objects whose features are intermingled with one another in the incoming sensory signal. In computer vision, the problem of scene analysis is to partition an image or video into regions attributed to the visible objects present in the scene. In audio there is a corresponding problem known as auditory scene analysis [1,2], which seeks to identify the components of audio signals corresponding to individual sound sources in a mixture signal. Both of these problems can be approached as *segmentation* problems, where we formulate a set of “*elements*” in the signal via an indexed set of features, each of which carries (typically multi-dimensional) information about part of the signal. For images, these elements are typically defined spatially in terms of pixels, whereas for audio signals they may be defined in terms of time-frequency coordinates. The segmentation problem is then solved by segmenting elements into groups or partitions, for example by assigning a group label to each element. Note that although *clustering* methods can be applied to segmentation problems, the segmentation problem is technically different in that clustering is classically formulated as a domain-independent problem based on simple objective functions defined on pairwise point relations, whereas partitioning may depend on complex processing of the whole input, and the task objective may be arbitrarily defined via training examples with given segment labels.

Segmentation problems can be broadly categorized into *class-based* segmentation problems where the goal is to learn from training class labels to label known object classes, versus more general *partition-based* segmentation problems where the task is to learn from labels of partitions, without requiring object class labels, to segment the input. Solving the partition-based problem has the advantage that unknown objects could then be segmented. In this paper, we propose a new partition-based approach which learns *embeddings* for each input elements, such that the correct labeling can be determined by simple clustering methods. We focus on the single-channel audio domain, although our methods are applicable to other domains such as images and multi-channel audio. The motivation for segmenting in this domain, as we shall describe later, is that using the segmentation as a mask, we can extract parts of the target signals that are not corrupted by other signals.

Since class-based approaches are relatively straightforward, and have been tremendously successful at their task, we first briefly mention this general approach. In class based vision models, such as [3–5], a hierarchical classification scheme is trained to estimate the class label of each pixel or super-pixel region. In the audio domain, single-channel speech separation methods, for example, segment the time-frequency elements of the spectrogram into regions dominated by a target speaker, either based on classifiers [6–8], or generative models [9–11]. In recent years, the success of deep neural networks for classification problems has naturally inspired their use in class-based segmentation problems [4, 12], where they have proven very successful.

However class-based approaches have some important limitations. First, of course, the assumed task of labeling known classes fundamentally does not address the general problem in real world signals that there may be a large number of possible classes, and many objects may not have a well-defined class. It is also not clear how to directly apply current class-based approaches to the more general problem. Class-based deep network models for separating sources require explicitly representing output classes and object instances in the output nodes, which leads to complexities in the general case. Although generative model-based methods can in theory be flexible with respect to the number of model types and instances after training, inference typically cannot scale computationally to the potentially larger problem posed by more general segmentation tasks.

In contrast, humans seem to solve the partition-based problem, since they can apparently segment well even with novel objects and sounds. This observation is the basis of Gestalt theories of perception, which attempt to explain perceptual grouping in terms of features such as proximity and similarity [13]. The partition-based segmentation task is closely related, and follows from a tradition of work in image segmentation and audio separation. Application of the perceptual grouping theory to audio segmentation is generally known as computational auditory scene analysis (CASA) [14, 15].

Spectral clustering is an active area of machine learning research with application to both image and audio segmentation. It uses local affinity measures between features of elements of the signal, and optimizes various objective functions using spectral decomposition of the normalized affinity matrix [16]. In contrast to conventional *central clustering* algorithms such as k-means, spectral clustering has the advantage that it does not require points to be tightly clustered around a central prototype, and can find clusters of arbitrary topology, provided that they form a connected sub-graph. Because of the local form of the pairwise kernel functions used, in difficult spectral clustering problems the affinity matrix has a sparse block-diagonal structure that is not directly amenable to central clustering, which works well when the block diagonal affinity structure is dense. The powerful but computationally expensive eigenspace transformation step of spectral clustering addresses this, in effect, by ”fattening” the block structure, so that connected components become dense blocks, prior to central clustering [17].

Although affinity-based methods were originally unsupervised inference methods, multiple-kernel learning methods such as [17, 18] were later introduced to train weights used to combine separate affinity measures. This allows us to consider using them for partition-based segmentation tasks in which partition labels are available, but without requiring specific class labels. In [17], this was applied to speech separation by including a variety of complex features developed to implement various auditory scene analysis grouping principles, such as similarity of onset/offset, pitch, spectral envelope, and so on, as affinities between time-frequency regions of the spectrogram. The input features included a dual pitch-tracking model in order to improve upon the relative simplicity of kernel-based features, at the expense of generality.

Rather than using specially designed features and relying on the strength of the spectral clustering framework to find clusters, we propose to use deep learning to derive embedding features that make the segmentation problem amenable to simple and computationally efficient clustering algorithms such as k-means, using the partition-based training approach. Learned feature transformations known as *embeddings* have recently been gaining significant interest in many fields. Unsupervised embeddings obtained by auto-associative deep networks, used with relatively simple clustering algorithms, have recently been shown to outperform spectral clustering methods [19, 20] in some cases. Embeddings trained using pairwise metric learning, such as word2vec [21] using neighborhood-based partition labels, have also been shown to have interesting invariance properties. We present below an objective function that minimizes the distances between embeddings of elements within a partition, while maximizing the distances between embeddings for elements in different partitions. This appears to be an appropriate criterion for central clustering methods. The proposed embedding approach has the attractive property that all partitions and their permutations can be represented implicitly using the fixed-dimensional output of the network.

The experiments described below show that the proposed method can separate speech using a speaker-independent model with an open set of speakers at test time. As in [17], we derive partition labels by mixing signals together and observing their spectral dominance patterns. After training on a database of mixtures of speakers trained in this way, we show that without any modification the model shows a promising ability to separate three-speaker mixtures despite training only on two-speaker mixtures. Although results are preliminary, the hope is that this work leads to methods that can achieve class-independent segmentation of arbitrary sounds, with additional application to image segmentation and other domains.

2 Learning deep embeddings for clustering

We define as x a raw input signal, such as an image or a time-domain waveform, and as $X_n = g_n(x), n \in \{1, \dots, N\}$, a feature vector indexed by an element n. In the case of images, n typically may be a superpixel index and X_n some vector-valued features of that superpixel; in the case of audio signals, n may be a time-frequency index (t, f), where t indexes frames of the signal and f frequencies, and $X_n = X_{t,f}$ the value of the complex spectrogram at the corresponding time-frequency bin. We assume that there exists a reasonable partition of the elements n into regions, which we would like to find, for example to further process the features X_n separately for each region. In the case of audio source separation, for example, these regions could be defined as the sets of time-frequency bins in which each source dominates, and estimating such a partition would enable us to build time-frequency masks to be applied to X_n, leading to time-frequency representations that can be inverted to obtain isolated sources.

To estimate the partition, we seek a K-dimensional embedding $V = f_\theta(x) \in \mathbb{R}^{N \times K}$, parameterized by θ, such that performing some simple clustering in the embedding space will likely lead to a partition of $\{1, \dots, N\}$ that is close to the target one. In this work, $V = f_\theta(x)$ is based on a deep neural network that is a global function of the entire input signal x (we allow for a feature extraction step to create the network input; in general, the input features may be completely different from X_n). Thus our transformation can take into account global properties of the input, and the embedding can be considered a permutation- and cardinality-independent encoding of the network's estimate of the signal partition. Here we consider a unit-norm embedding, so that $|v_n|^2 = \sum_k v_{n,k}^2 = 1, \forall n$, where $v_n = \{v_{n,k}\}$ and $v_{n,k}$ is the value of the k-th dimension of the embedding for element n. We omit the dependency of V on θ to simplify notations.

The partition-based training requires a reference label indicator $Y = \{y_{n,c}\}$, mapping each element n to each of c arbitrary partition classes, so that $y_{n,c} = 1$ if element n is in partition c. For a training objective, we seek embeddings that enable accurate clustering according to the partition labels. To do this, we need a convenient expression that is invariant to the number and permutations of the partition labels from one training example to the next. One such objective for minimization is

$$C(\theta) = |VV^T - YY^T|_W^2 = \sum_{i,j:y_i=y_j} \frac{(\langle v_i, v_j\rangle - 1)^2}{d_i} + \sum_{i,j:y_i\neq y_j} \frac{(\langle v_i, v_j\rangle - 0)^2}{\sqrt{d_i d_j}}, \tag{1}$$

$$= \sum_{i,j:y_i=y_j} \frac{|v_i - v_j|^2}{d_i} + \sum_{i,j} \frac{\left(|v_i - v_j|^2 - 2\right)^2}{4\sqrt{d_i d_j}} - N, \tag{2}$$

where $|A|_W^2 = \sum_{i,j} w_{i,j} a_{i,j}^2$ is a weighted Frobenius norm, with $W = d^{-\frac{1}{2}} d^{-\frac{T}{2}}$, where $d_i = YY^T 1$ is an $(N \times 1)$ vector of partition sizes: that is, $d_i = |\{j : y_i = y_j\}|$. In the above we use the fact that $|v_n|^2 = 1, \forall n$. Intuitively, this objective pushes the inner product $\langle v_i, v_j\rangle$ to 1 when i and j are in the same partition, and to 0 when they are in different partitions. Alternately, we see from (2) that it pulls the squared distance $|v_i - v_j|^2$ to 0 for elements within the same partition, while preventing the embeddings from trivially collapsing into the same point. Note that the first term is the objective function minimized by k-means, as a function of cluster assignments, and in this context the second term is a constant. So the objective reasonably tries to lower the k-means score for the labeled cluster assignments at training time.

This formulation can be related to spectral clustering as follows. We can define an ideal affinity matrix $A^* = YY^T$, that is block diagonal up to permutation and use an inner-product kernel, so that $A = VV^T$ is our affinity matrix. Our objective becomes $C = |A - A^*|_\mathrm{F}^2$, which measures the deviation of the model's affinity matrix from the ideal affinity. Note that although this function ostensibly sums over all pairs of data points i, j, the low-rank nature of the objective leads to an efficient implementation, defining $D = \mathrm{diag}(YY^T 1)$:

$$C = |VV^T - YY^T|_\mathrm{W}^2 = |V^T D^{-\frac{1}{2}} V|_\mathrm{F}^2 - 2|V^T D^{-\frac{1}{2}} Y|_\mathrm{F}^2 + |Y^T D^{-\frac{1}{2}} Y|_\mathrm{F}^2, \tag{3}$$

which avoids explicitly constructing the $N \times N$ affinity matrix. In practice, N is orders of magnitude greater than K, leading to a significant speedup. To optimize a deep network, we typically need to use first-order methods. Fortunately derivatives of our objective function with respect to V are also efficiently obtained due to the low-rank structure:

$$\frac{\partial C}{\partial V^T} = 4D^{-\frac{1}{2}} VV^T D^{-\frac{1}{2}} V - 4D^{-\frac{1}{2}} YY^T D^{-\frac{1}{2}} V. \tag{4}$$

This low-rank formulation also relates to spectral clustering in that the latter typically requires the Nyström low-rank approximation to the affinity matrix, [22] for efficiency, so that the singular value decomposition (SVD) of an $N \times K$ matrix can be substituted for the much more expensive eigenvalue decomposition of the $K \times K$ normalized affinity matrix. Rather than following spectral clustering in making a low-rank approximation of a full-rank model, our method can be thought of as directly optimizing a low-rank affinity matrix so that processing is more efficient and parameters are tuned to the low-rank structure.

At test time, we compute the embeddings V on the test signal, and cluster the rows $v_i \in \mathbb{R}^K$, for example using k-means. We also alternately perform a spectral-clustering style dimensionality reduction before clustering, starting with a singular value decomposition (SVD), $\tilde{V} = USR^T$, of normalized $\tilde{V} = D^{-\frac{1}{2}} V$, where $D = VV^T 1_N$, sorted by decreasing eigenvalue, and clustering the normalized rows of the matrix of m principal left singular vectors, with the i'th row given by $\tilde{u}_{i,r} = u_{i,r} / \sqrt{\sum_{r'=1}^m u_{i,r'}} \; : r \in [1, m]$, similar to [23].

3 Speech separation experiments

3.1 Experimental setup

We evaluate the proposed model on a speech separation task: the goal is to separate each speech signal from a mixture of multiple speakers. While separating speech from non-stationary noise is in general considered to be a difficult problem, separating speech from other speech signals is particularly challenging because all sources belong to the same class, and share similar characteristics. Mixtures involving speech from same gender speakers are the most difficult since the pitch of the voice is in the same range. We here consider mixtures of two speakers and three speakers (the latter always containing at least two speakers of the same gender). However, our method is not limited in the number of sources it can handle or the vocabulary and discourse style of the speakers. To investigate the effectiveness of our proposed model, we built a new dataset of speech mixtures based on the Wall Street Journal (WSJ0) corpus, leading to a more challenging task than in existing datasets. Existing datasets are too limited for evaluation of our model because, for example, the speech separation challenge [24] only contains a mixture of two speakers, with a limited vocabulary and insufficient training data. The SISEC challenge (e.g., [25]) is limited in size and designed for the evaluation of multi-channel separation, which can be easier than single-channel separation in general.

A training set consisting of 30 hours of two-speaker mixtures was generated by randomly selecting utterances by different speakers from the WSJ0 training set `si_tr_s`, and by mixing them at various signal-to-noise ratios (SNR) between 0 dB and 5 dB. We also designed the two training subsets from the above whole training set (*whole*), one considered the balance of the mixture of the genders (*balanced*, 22.5 hours), and the other only used the mixture of female speakers (*female*, 7.5 hours). 10 hours of cross validation set were generated similarly from the WSJ0 training set, which is used to optimize some tuning parameters, and to evaluate the source separation performance of the closed speaker experiments (**closed speaker set**). 5 hours of evaluation data was generated similarly using utterances from sixteen speakers from the WSJ0 development set `si_dt_05` and evaluation set `si_et_05`, which are based on the different speakers from our training and closed speaker sets (**open speaker set**). Note that many existing speech separation methods (e.g., [5, 26]) cannot handle the open speaker problem without special adaptation procedures, and generally require knowledge of the speakers in the evaluation. For the evaluation data, we also created 100 utterances of three-speaker mixtures for each closed and open speaker set as an advanced setup. All data were downsampled to 8 kHz before processing to reduce computational and memory costs.

The input features X were the log short-time Fourier spectral magnitudes of the mixture speech, computed with a 32 ms window length, 8 ms window shift, and the square root of the hann window. To ensure the local coherency, the mixture speech was segmented with the length of 100 frames, roughly the length of one word in speech, and processed separately to output embedding V based on the proposed model. The ideal binary mask was used to build the target Y when training our network. The ideal binary mask gives ownership of a time-frequency bin to the source whose magnitude is maximum among all sources in that bin. The mask values were assigned with 1 for active and 0 otherwise (binary), making YY^T as the ideal affinity matrix for the mixture.

To avoid problems due to the silence regions during separation, a binary weight for each time-frequency bin was used during the training process, only retaining those bins such that each source's magnitude at that bin is greater than some ratio (arbitrarily set to -40 dB) of the source's maximum magnitude. Intuitively, this binary weight guides the neural network to ignore bins that are not important to all sources.

3.2 Training procedure

Networks in the proposed model were trained given the above input X and the ideal affinity matrix YY^T. The network structure used in our experiments has two bi-directional long short-term memory (BLSTM) layers, followed with one feedforward layer. Each BLSTM layer has 600 hidden cells and the feedforward layer corresponds with the embedding dimension (i.e., K). Stochastic gradient descent with momentum 0.9 and fixed learning rate 10^-5 was used for training. In each updating step, a Gaussian noise with zero mean and 0.6 variance was added to the weight. We prepared several networks used in the speech separation experiments using different embedding dimensions from 5 to 60. In addition, two different activation functions (logistic and tanh) were explored to form the embedding V with different ranges of $v_{n,k}$. For each embedding dimension, the weights for the corresponding network were initialized randomly from the scratch according to a normal distribution with zero mean and 0.1 variance with the tanh activation and *whole* training set. In the experiments of a different activation (logistic) and different training subsets (*balanced* and *female*), the network was initialized with the one with the tanh activation and *whole* training set. The implementation was

Table 1: SDR improvements (in dB) for different clustering methods.

method	closed speaker set	open speaker set
oracle NMF	5.06	-
DC oracle k-means	6.54	6.45
DC oracle spectral	6.35	6.26
DC global k-means	5.87	5.81

Table 2: SDR improvements (in dB) for different embedding dimensions K and activation functions

	closed speaker set		open speaker set	
model	DC oracle	DC global	DC oracle	DC global
$K = 5$	-0.77	-0.96	-0.74	-1.07
$K = 10$	5.15	4.52	5.29	4.64
$K = 20$	6.25	5.56	6.38	5.69
$K = 40$	6.54	5.87	6.45	5.81
$K = 60$	6.00	5.19	6.08	5.28
$K = 40$ logistic	6.59	5.86	6.61	5.95

based on CURRENNT, a publicly available training software for DNN and (B)LSTM networks with GPU support (`https://sourceforge.net/p/currennt`).

3.3 Speech separation procedure

In the test stage, the speech separation was performed by constructing a time-domain speech signal based on time-frequency masks for each speaker. The time-frequency masks for each source speaker were obtained by clustering the row vectors of embedding V, where V was outputted from the proposed model for each segment (100 frames), similarly to the training stage. The number of clusters corresponds to the number of speakers in the mixture. We evaluated various types of clustering methods: k-means on the whole utterance by concatenating the embeddings V for all segments; k-means clustering within each segment; spectral clustering within each segment. For the within-segment clusterings, one needs to solve a permutation problem, as clusters are not guaranteed to be consistent across segments. For those cases, we report oracle permutation results (i.e., permutations that minimize the L^2 distance between the masked mixture and each source's complex spectrogram) as an upper bound on performance.

One interesting property of the proposed model is that it can potentially generalize to the case of three-speaker mixtures without changing the training procedure in Section 3.2. To verify this, speech separation experiments on three-speaker mixtures were conducted using the network trained with two speaker mixtures, simply changing the above clustering step from 2 to 3 clusters. Of course, training the network including mixtures involving more than two speakers should improve performance further, but we shall see that the method does surprisingly well even without retraining.

As a standard speech separation method, supervised sparse non-negative matrix factorization (SNMF) was used as a baseline [26]. While SNMF may stand a chance separating speakers in male-female mixtures when using a concatenation of bases trained separately on speech by other speakers of each gender, it would not make sense to use it in the case of same-gender mixtures. To give SNMF the best possible advantage, we use an oracle: at test time we give it the basis functions trained on the actual speaker in the mixture. For each speaker, 256 bases were learned on the clean training utterances of that speaker. Magnitude spectra with 8 consecutive frames of left context were used as input features. At test time, the basis functions for the two speakers in the test mixture were concatenated, and their corresponding activations computed on the mixture. The estimated models for each speaker were then used to build a Wiener-filter like mask applied to the mixture, and the corresponding signals reconstructed by inverse STFT.

For all the experiment, performance was evaluated in terms of averaged signal-to-distortion ratio (SDR) using the `bss_eval` toolbox [27]. The initial SDR averaged over the mixtures was 0.16 dB for two speaker mixtures and -2.95 dB for three speaker mixtures.

Table 3: SDR improvement (in dB) for each type of mixture. Scores averaged over male-male (m+m), female-female (f+f), female-male (f+m), or all mixtures.

	training gender	**closed speaker set**				**open speaker set**			
method	**distribution**	m+m	f+f	f+m	all	m+m	f+f	f+m	all
oracle NMF	speaker dependent	3.25	3.31	6.53	4.90	-	-	-	-
DC oracle permute	*whole*	3.79	4.29	9.04	6.54	4.49	3.21	8.69	6.45
	balanced	3.89	4.35	8.74	6.42	4.61	3.49	8.27	6.41
	female	-	5.03	-	-	-	4.04	-	-
DC global k-means	*whole*	2.54	2.85	9.07	5.87	3.51	1.42	8.57	5.80
	balanced	2.78	2.87	8.63	5.72	3.89	1.74	8.27	5.83
	female	-	3.88	-	-	-	2.56	-	-

Table 4: SDR improvement (in dB) for three speaker mixture

method	**closed speaker set**	**open speaker set**
oracle NMF	4.42	-
DC oracle	3.50	2.81
DC global	2.74	2.22

4 Results and discussion

As shown in Table 1, both the oracle and non-oracle clustering methods for proposed system significantly outperform the oracle NMF baseline, even though the oracle NMF is a strong model with the important advantage of knowing the speaker identity and has speaker-dependent models. For the proposed system the open speaker performance is similar to the closed speaker results, indicating that the system can generalize well to unknown speakers, without any explicit adaptation methods. For different clustering methods, the oracle k-means outperforms the oracle ”spectral clustering” by 0.19 dB showing that the embedding represents centralized clusters. To be fair, what we call spectral clustering here is using our outer product kernel instead of a local kernel function such as a Gaussian, as commonly used in spectral clustering. However a Gaussian kernel could not be used here due to computational complexity. Also note that the oracle clustering method in our experiment resolves the permutation of two (or three in Table 4) speakers in each segment. In the dataset, each utterance usually contains 6$\sim$8 segments so the permutation search space is relatively small for each utterance. Hence this problem may have an easy solution to be explored in future work. For the non-oracle experiments, the whole utterance clustering also performs relatively well compared to baseline. Given the fact that the system was only trained with individual segments, the effectiveness of the whole utterance clustering suggests that the network learns features that are globally important, such us pitch, timbre etc.

In Table 2, the $K = 5$ system completely fails, either because optimization of the current network architecture fails, or the embedding fundamentally requires more dimensions. The performance of $K = 20$, $K = 40$, $K = 60$ are similar, showing that the system can operate in a wide range of parameter values. We arbitrarily used tanh networks in most of the experiments because the tanh network has larger embedding space than logistic network. However, in Table 2, we show that in retrospect the logistic network performs slightly better than the tanh one.

In Table 3, since the female and male mixture is an intrinsically easier segmentation problem, the performance of mixture between female and male is significantly better than the same gender mixtures for all situations. As mentioned in Section 3, the random selection of speaker would also be a factor for the large gap. With more balanced training data, the system has better performance for the same gender separation with a sacrifice of its performance for different gender mixture. If we only focus on female mixtures, the performance is still better.

Figure 2 shows an example of embeddings for two different mixtures (female-female and male-female), in which a few embedding dimensions are plotted for each time-frequency bin in order to show how they are sensitive to different aspects of each signal.

In Table 4, the proposed system can also separate the mixture of three speakers, even though it is only trained on two-speaker mixtures. As discussed in previous sections, unlike many separation algorithms, deep clustering can naturally scale up to more sources, and thus make it suitable for many real world tasks when the number of sources is not available or fixed. Figure 1 shows one

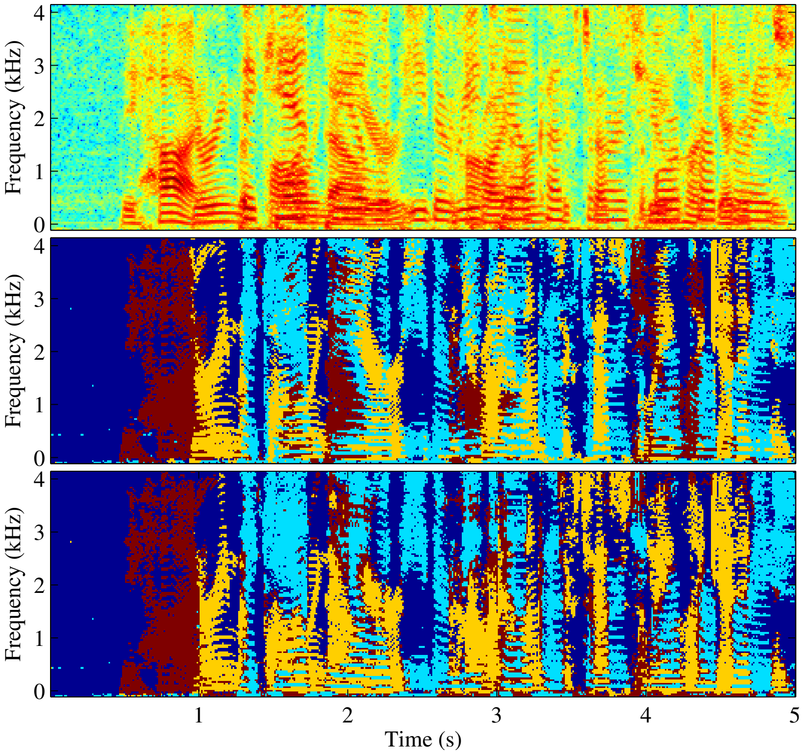

Figure 1: An example of three-speaker separation. Top: log spectrogram of the input mixture. Middle: ideal binary mask for three speakers. The dark blue shows the silence part of the mixture. Bottom: output mask from the proposed system trained on two-speaker mixtures.

example of the separation for three speaker mixture in the open speaker set case. Note that we also did experiments with mixtures of three fixed speakers for the training and testing data, and the SDR improvement of the proposed system is 6.15.

Deep clustering has been evaluated in a variety of conditions and parameter regimes, on a challenging speech separation problem. Since these are just preliminary results, we expect that further refinement of the model will lead to significant improvement. For example, by combining the clustering step into the embedding BLSTM network using the deep unfolding technique [28], the separation could be jointly trained with embedding and lead to potential better result. Also in this work, the BLSTM network has a relatively uniform structure. Alternative architectures with different time and frequency dependencies, such as deep convolutional neural networks [3], or hierarchical recursive embedding networks [4], could also be helpful in terms of learning and regularization. Finally, scaling up training on databases of more disparate audio types, as well as applications to other domains such as image segmentation, would be prime candidates for future work.

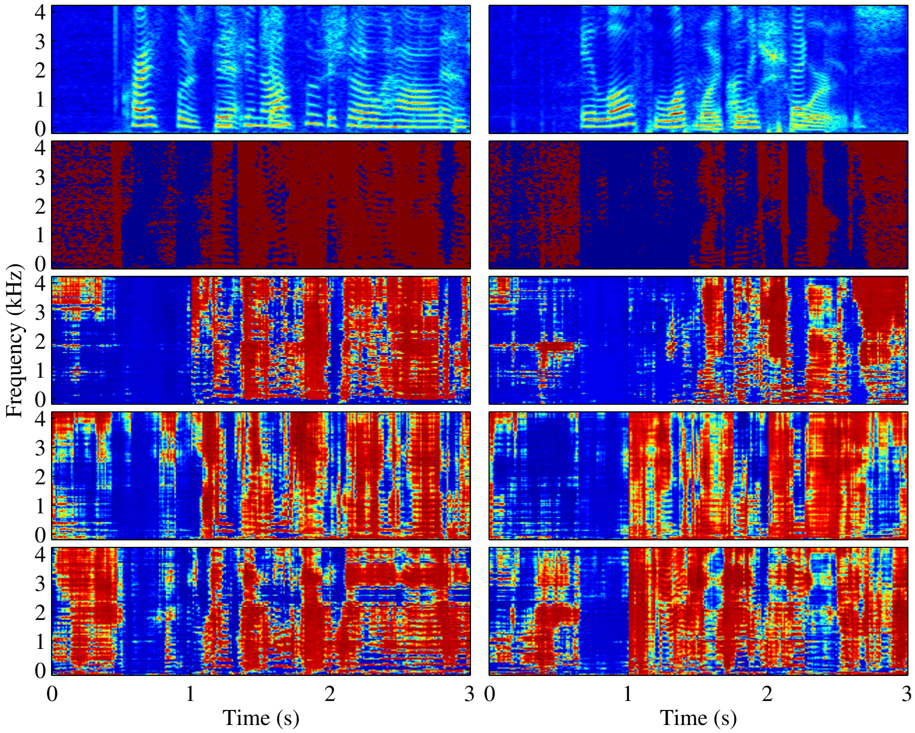

Figure 2: Examples of embeddings for two mixtures: f+f (left) and f+m (right). 1st row: spectrogram; 2nd row: ideal binary mask; 3rd-5th row: embeddings.

References

[1] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound*, MIT press, 1994.

[2] C. J. Darwin and R. P. Carlyon, “Auditory grouping,” in *Hearing*, B. Moore, Ed. Elsevier, 1995.

[3] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” *IEEE Trans. PAMI*, vol. 35, no. 8, pp. 1915–1929, 2013.

[4] A. Sharma, O. Tuzel, and M.-Y. Liu, “Recursive context propagation network for semantic scene labeling,” in *Proc. NIPS*, 2014, pp. 2447–2455.

[5] P. Smaragdis, “Convolutive speech bases and their application to supervised speech separation,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 1, pp. 1–12, 2007.

[6] R. J. Weiss and D. P. Ellis, “Estimating single-channel source separation masks: Relevance vector machine classifiers vs. pitch-based masking,” in *Proc. SAPA*, 2006, pp. 31–36.

[7] G. Kim, Y. Lu, Y. Hu, and P. C. Loizou, “An algorithm that improves speech intelligibility in noise for normal-hearing listeners,” *J. Acoust. Soc. Am.*, vol. 126, no. 3, pp. 1486–1494, 2009.

[8] Y. Wang, K. Han, and D. Wang, “Exploring monaural features for classification-based speech segregation,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 2, pp. 270–279, 2013.

[9] S. T. Roweis, “Factorial models and refiltering for speech separation and denoising,” in *Proc. Eurospeech*, 2003, pp. 1009–1012.

[10] M. Schmidt and R. Olsson, “Single-channel speech separation using sparse non-negative matrix factorization,” in *Proc. Interspeech*, 2006, pp. 3111–3119.

[11] J. R. Hershey, S. J. Rennie, P. A. Olsen, and T. T. Kristjansson, “Super-human multi-talker speech recognition: A graphical modeling approach,” *Comput. Speech Lang.*, vol. 24, no. 1, pp. 45–66, 2010.

[12] Y. Wang and D. Wang, “Towards scaling up classification-based speech separation,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 7, pp. 1381–1390, 2013.

[13] M. Wertheimer, “Laws of organization in perceptual forms,” in *A Source book of Gestalt psychology*, W. A. Ellis, Ed., pp. 71–88. Routledge and Kegan Paul, 1938.

[14] M. P. Cooke, *Modelling auditory processing and organisation*, Ph.D. thesis, Univ. of Sheffield, 1991.

[15] D. P. W. Ellis, *Prediction-Driven Computational Auditory Scene Analysis*, Ph.D. thesis, MIT, 1996.

[16] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. PAMI*, vol. 22, no. 8, pp. 888–905, 2000.

[17] F. R. Bach and M. I. Jordan, “Learning spectral clustering, with application to speech separation,” *JMLR*, vol. 7, pp. 1963–2001, 2006.

[18] C. Fowlkes, D. Martin, and J. Malik, “Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches,” in *Proc. CVPR*, 2003, vol. 2, pp. 54–61.

[19] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, “Learning deep representations for graph clustering,” in *Proc. AAAI*, 2014.

[20] P. Huang, Y. Huang, W. Wang, and L. Wang, “Deep embedding network for clustering,” in *Proc. ICPR*, 2014, pp. 1532–1537.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. NIPS*, 2013, pp. 3111–3119.

[22] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, “Spectral grouping using the nyström method,” *IEEE Trans. PAMI*, vol. 26, no. 2, pp. 214–225, 2004.

[23] A. Y. Ng, M. I. Jordan, Y. Weiss, et al., “On spectral clustering: Analysis and an algorithm,” in *Proc. NIPS*, 2002, pp. 849–856.

[24] M. Cooke, J. R. Hershey, and S. J. Rennie, “Monaural speech separation and recognition challenge,” *Computer Speech & Language*, vol. 24, no. 1, pp. 1–15, 2010.

[25] E. Vincent, S. Araki, F. Theis, G. Nolte, P. Bofill, H. Sawada, A. Ozerov, V. Gowreesunker, D. Lutter, and N. Q. Duong, “The signal separation evaluation campaign (2007–2010): Achievements and remaining challenges,” *Signal Processing*, vol. 92, no. 8, pp. 1928–1936, 2012.

[26] J. Le Roux, F. J. Weninger, and J. R. Hershey, “Sparse NMF – half-baked or well done?,” Tech. Rep. TR2015-023, MERL, Cambridge, MA, USA, Mar. 2015.

[27] E. Vincent, R. Gribonval, and C. Févotte, “Performance measurement in blind audio source separation,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 4, pp. 1462–1469, 2006.

[28] J. R. Hershey, J. Le Roux, and F. Weninger, “Deep unfolding: Model-based inspiration of novel deep architectures,” Sep. 2014, arXiv:1409.2574.