Unsupervised Deep Clustering for Source Separation: Direct Learning from Mixtures using Spatial Information

Efthymios Tzinis[1], Shrikant Venkataramani[2], Paris Smaragdis[1,3]

[1]University of Illinois at Urbana-Champaign, Department of Computer Science
[2]University of Illinois at Urbana-Champaign, Department of Electrical & Computer Engineering
[3]Adobe Research

Motivation

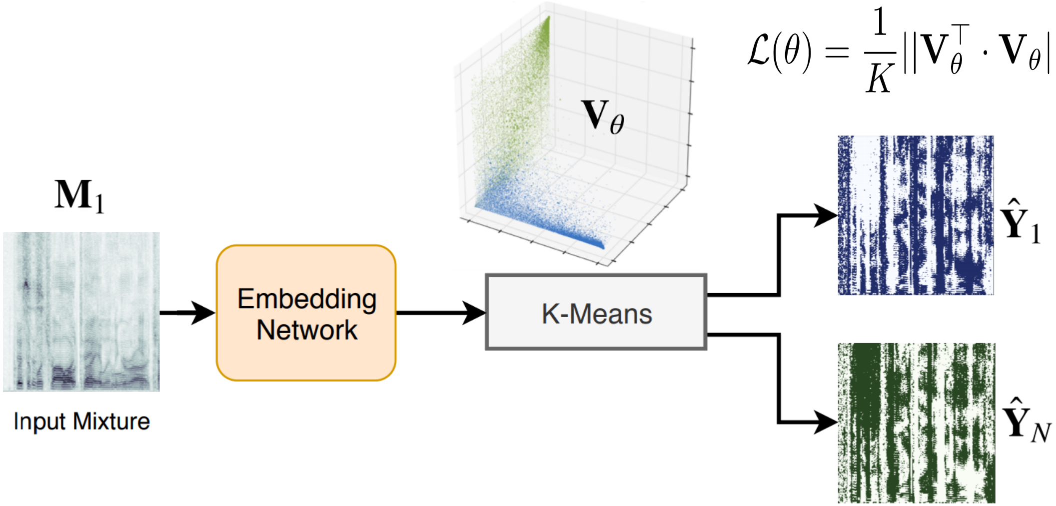
- Humans learn to distinguish sources without supervision
  - Can we extract features that help to develop a source separator?

- Unsupervised learning approach
  - Avoiding having to collect *paired* data (mixtures and clean sources)
    - Modern systems require input/output training pairs
  - Exploit spatial information in order to learn <u>single-channel</u> separation

Supervised Deep Clustering

- Embed the input STFT bins to a latent space
  - Hopefully these embeddings would become easily separable
- Requires a ground truth partitioning as training target
  - We need to know the dominating source for each STFT bin
- Can we do it unsupervised using spatial information?

$$\mathcal{L}(\theta) = \frac{1}{K}||\mathbf{V}_\theta^\top \cdot \mathbf{V}_\theta||_F + \frac{1}{N}||\mathbf{Y}^\top \cdot \mathbf{Y}||_F - \frac{2}{\sqrt{KN}}||\mathbf{V}_\theta^\top \cdot \mathbf{Y}||_F$$



Deriving targets from spatial features

- Dominant Source (DS) masks require paired inputs/outputs
  - Could be time consuming
  - Could result in biased data

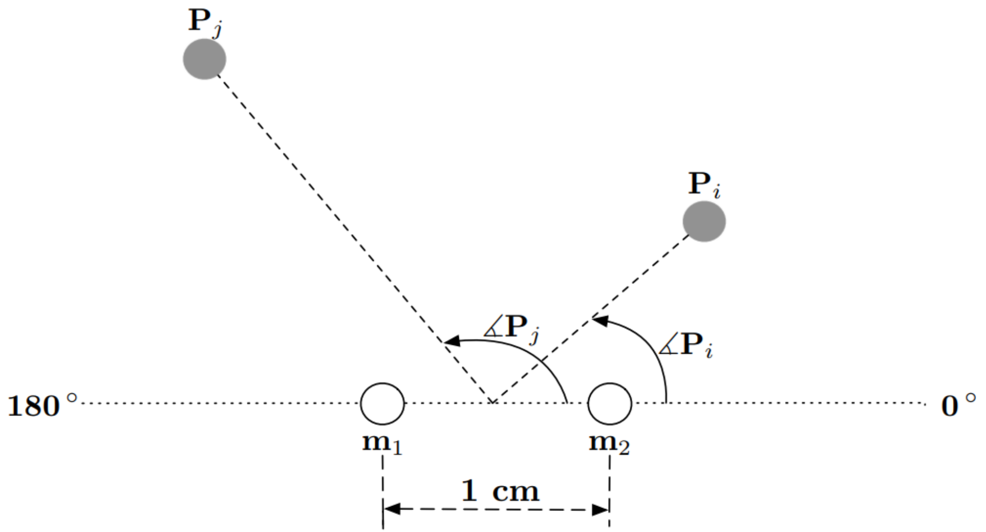
- We can instead exploit spatial information using 2 mics
  - We can construct masks without needing ground truth
  - Each TF bin gets a mask value derived from spatial features

Mixture model and Environment

- Assumptions:
  - Sources have distinct spatial locations $|\angle(\mathbf{P}_i) - \angle(\mathbf{P}_j)| > 10^o \ \forall\{i,j\}, i \neq j$
  - Anechoic environment
  - Two nearby microphones
    - Time difference less < 1 sample

$$\mathbf{m}_1(t) = a_1 \cdot \mathbf{s}_1(t) + \cdots + a_N \cdot \mathbf{s}_N(t)$$
$$\mathbf{m}_2(t) = a_1 \cdot \mathbf{s}_1(t+\delta\tau_1) + \cdots + a_N \cdot \mathbf{s}_N(t+\delta\tau_N)$$

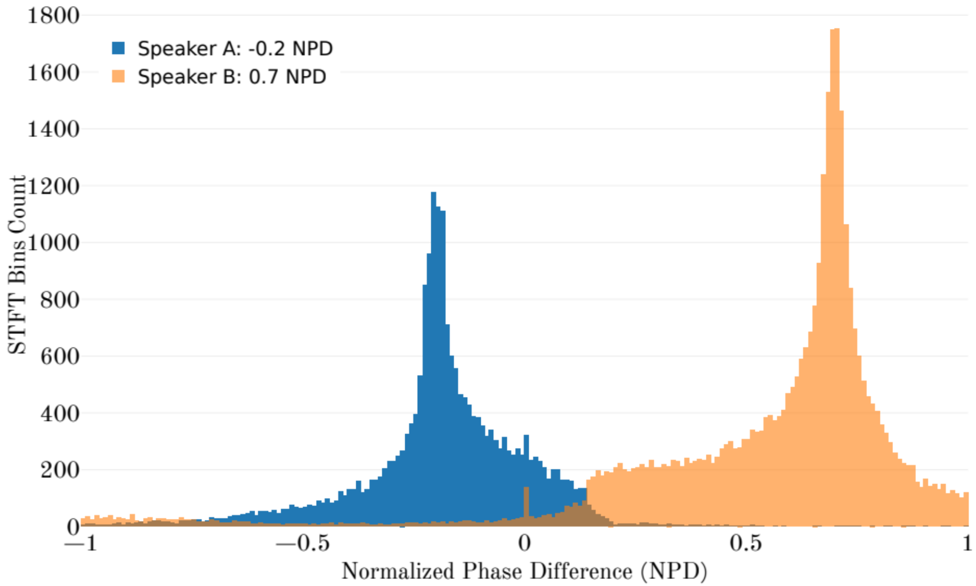
$$\begin{bmatrix} \mathbf{M}_1(\omega,m) \\ \mathbf{M}_2(\omega,m) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ e^{j\omega\delta\tau_1} & \cdots & e^{j\omega\delta\tau_N} \end{bmatrix} \cdot \begin{bmatrix} a_1 \cdot \mathbf{S}_1(\omega,m) \\ \vdots \\ a_N \cdot \mathbf{S}_N(\omega,m) \end{bmatrix}$$



Extracting spatial features

- Normalized Phase Difference (NPD)
  - Like DUET algorithm, we extract for each STFT bin a value corresponding to the cross-mic delay by using the STFT phase difference
  - Values are easily clustered if sources are spatially separable

$$\delta\phi(\omega, m) = \frac{1}{\omega} \angle \frac{\mathbf{M}_1(\omega, m)}{\mathbf{M}_2(\omega, m)}$$



Defining a separating partition

- Dominating Source (DS) partition
  - Derived from finding loudest source for each STFT bin

$$\mathbf{Y}_{DS}(\omega, m, i) = \begin{cases} 1 & i = \operatorname*{argmax}_{1 \leq j \leq N} \left(a_j \cdot |\mathbf{S}_j(\omega, m)|\right) \\ 0 & \text{otherwise} \end{cases}$$

- Raw Phase Difference (RPD) partition
  - Use the NPD values instead of the DS mask

$$\mathbf{Y}_{RPD} = \delta\phi(\omega, m) = \frac{1}{\omega} \angle \frac{\mathbf{M}_1(\omega, m)}{\mathbf{M}_2(\omega, m)}$$

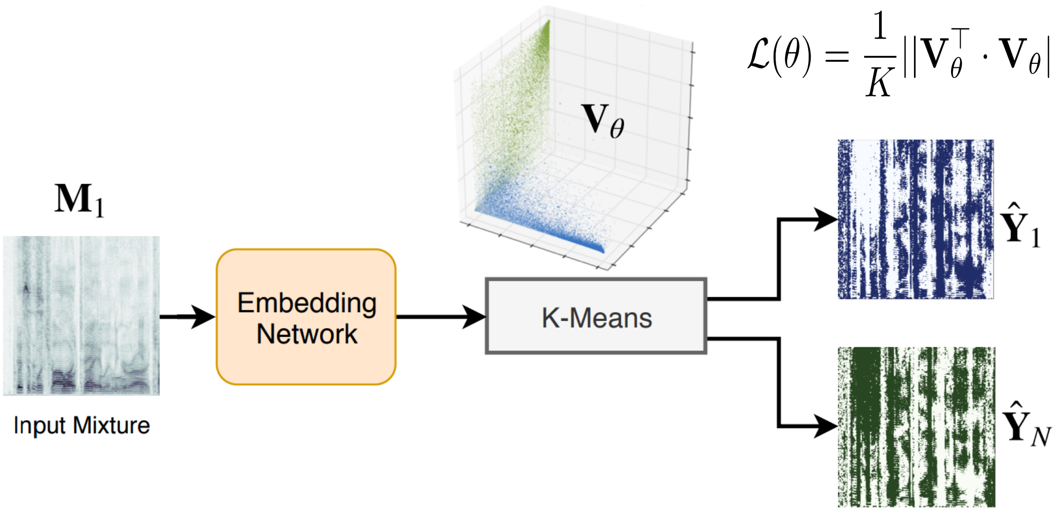
- Binary Phase Difference (BPD) partition
  - Cluster labels after a K-means clustering assignment on the NPD values $\mathcal{R}(\omega, m)$

$$\mathbf{Y}_{BPD}(\omega, m, i) = \begin{cases} 1 & i = \mathcal{R}(\omega, m) \\ 0 & \text{otherwise} \end{cases}$$

Unsupervised Deep Clustering

- Train the network using the unsupervised partition
  - Train using either the BPD or the RPD partition as targets
- Separating multiple speakers
  - Changing the number of clusters in K-means accordingly

$$\mathcal{L}(\theta) = \frac{1}{K}||\mathbf{V}_\theta^\top \cdot \mathbf{V}_\theta||_F + \frac{1}{N}||\mathbf{Y}^\top \cdot \mathbf{Y}||_F - \frac{2}{\sqrt{KN}}||\mathbf{V}_\theta^\top \cdot \mathbf{Y}||_F$$



Overall process

- Training procedure
  - Obtain multichannel spatial mixtures, extract NPD feature and train a deep clustering model using the NPD-based partionings as targets
- Separation procedure (Inference)
  - Receive a single-channel mixture, apply learned deep clustering network that clusters the STFT bins, and extract sources
- Important distinctions
  - We train on multichannel data, but deploy on monophonic inputs
  - We do not use separated sounds as targets at all

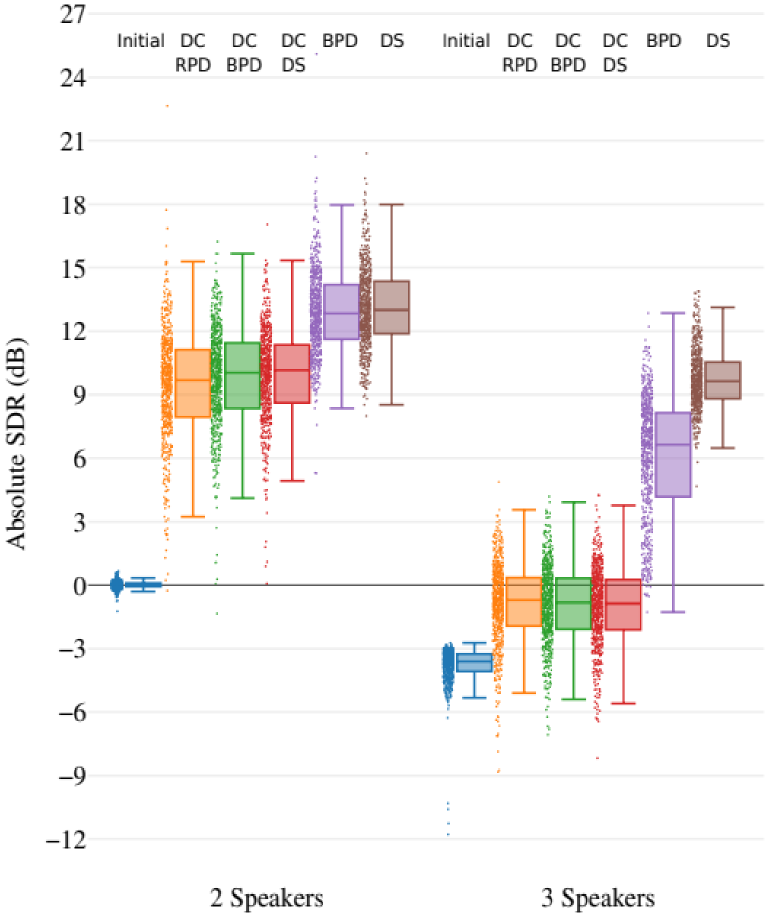
Experimental setup

- Generated mixture dataset using TIMIT
  - Female (f), Male (m) and Female-Male (fm) mixtures
  - Generated sets for 2 or 3 simultaneous speakers
  - Training data: 3 hours, Validation: 0.5 hour, Test: 1 hour

- Speaker independent experiments
  - Trained deep clustering model using Y_{DS}, Y_{RPD}, Y_{BPD} separately
  - Evaluated performance using Signal to Distortion Ratio (SDR)

Trained on 2-speaker Female-Male mixtures

- Ground truth comparison
  - BPD and DS partitions provide a similar upper bound in performance
- Separation comparison
  - Unsupervised methods based on BPD and RPD perform just as well as the baseline deep clustering model
    - But training was done on mixtures!



Mean SDR Improvement - 2 speakers

- Supervised and unsupervised partitions result to <u>similar performance</u>
  - BPD usually performs better than RPD
  - BPD is slightly worse than DS
    - Expected since the oracle mask is a little worse too

		f	*fm*	*m*	*all*
Separations	DC RPD	4.85	9.43	3.51	6.80
	DC BPD	7.17	9.99	4.97	8.03
	DC DS	7.57	10.15	5.16	8.26
Oracles	BPD	13.65	12.88	11.82	12.81
	DS	14.02	13.19	12.14	13.14

Mean SDR Improvement - 3 speakers

- DC was trained on 2 speakers and tested on 3 directly
- Similar behavior as before
  - We always obtain a net improvement of mixtures
  - Unsupervised and supervised partitions obtain similar performance

		f	*fm*	*m*	*all*
Separations	DC RPD	1.04	2.77	0.27	1.71
	DC BPD	1.75	2.75	1.39	2.16
	DC DS	1.66	2.67	1.44	2.11
Oracles	BPD	10.23	9.76	8.83	9.64
	DS	13.88	13.44	12.41	13.29

Conclusions

- We can learn to separate by using the phase difference
  - Separation can subsequently work on single-channel inputs
- Using unsupervised approach simplifies data collection
  - No need for paired data (mixtures and clean sources), can learn on the field
  - Performance hit is small, but could be reduced with more fine-tuning
- Easily extendable concept
  - Can be used with more sophisticated separation architectures
  - Can be applied using other types of partition features (spatial or not)
  - One can make use of more than two microphones

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Questions?

Efthymios Tzinis

etzinis2@illinois.edu
https://etzinis.com