ANN Supported Source Separation

Ilya Sklyar
ilya.sklyar@rwth-aachen.de

Selected Topics in Human Language Technology and Pattern Recognition
27.06.2017

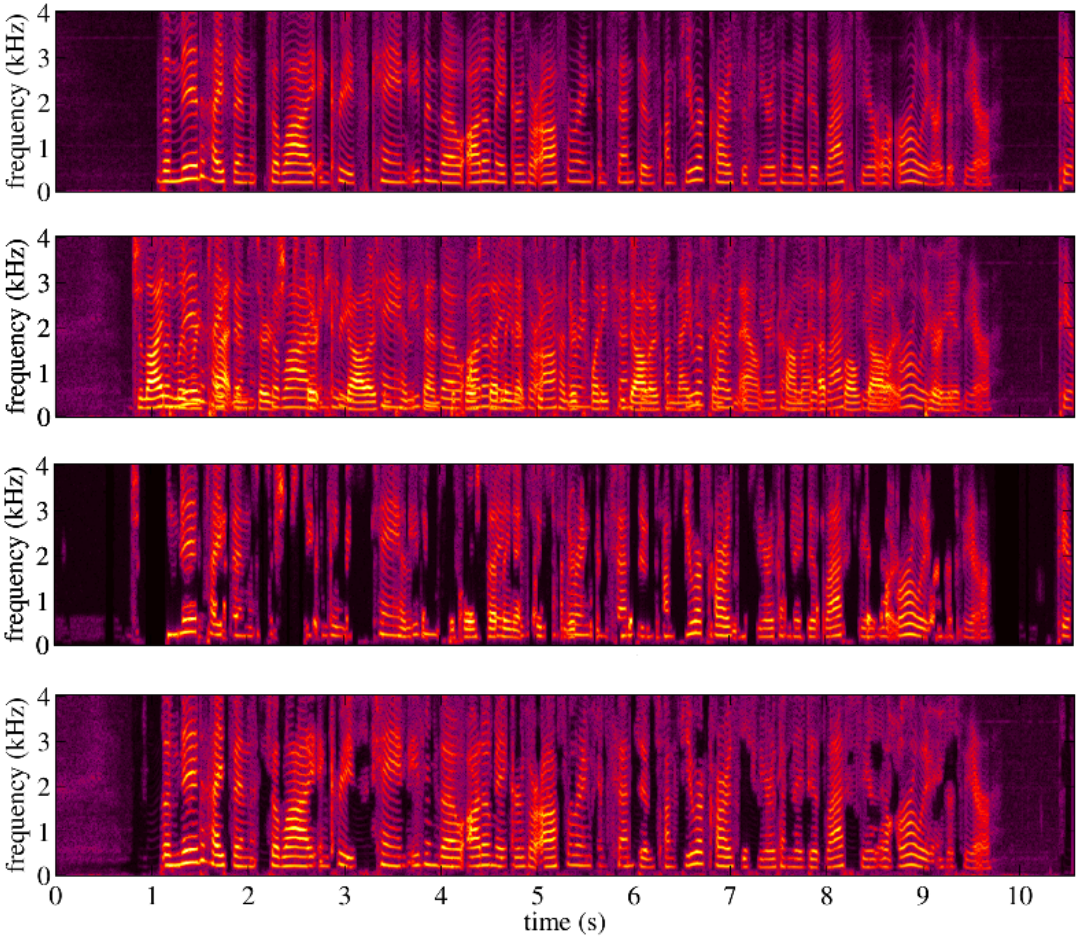
Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany

Outline

Introduction

- Cocktail party problem [Cherry 57]
  - Overlapping mixture of sounds
  - Arbitrary number of sources
  - Properties of sources are not known in advance
  - Sources can have very similar nature (i.e. two female speakers)
  - Single channel: no spatial information
- Wide range of applications
  - Virtual assistants
  - Hearing aids
  - Meeting transcriptions
  - Automatic captioning for audio/video recordings
  - General audio separation

Motivation: separating mixture of two female speakers



Clean

Mixture

Conventional

ANN-supported

[Dem]

Literature

[Hershey & Chen^{+} 16]: Deep clustering: Discriminative embeddings for segmentation and separation. *ICASSP 2016*.

- Generation of the embeddings via ANN that learns similarity stricture of time-frequency bins in the mixture
- Clustering of the embeddings to obtain binary masks that distinguish sources

[Isik & Roux^{+} 16]: Single-channel multi-speaker separation using deep clustering.

- Enhancement network for signal reconstruction
- End-to-end training

[Chen & Luo^{+} 16]: Deep attractor network for single-microphone speaker separation.

- Attractor points estimation in the embedding space
- More efficient end-to-end training

Literature

[Yu & Kolbæk^{+} 16]: Permutation Invariant Training of Deep Models for Speaker-Independent Multi-talker Speech Separation.

- Mixture is represented as a set of sources
- Label assignment is performed simultaneously with error evaluation

[Kolbæk & Yu^{+} 17]: Multi-talker Speech Separation and Tracing with Permutation Invariant Training of Deep Recurrent Neural Networks

- All frames of the same speaker are aligned to the same output layer
- LSTM network is used to learn long-term dependencies in the mixture

[Yu & Chang^{+} 17]: Recognizing Multi-talker Speech with Permutation Invariant Training.

- Separation is performed implicitly in ASR framework
- Direct recognition of multiple streams of speech

Mathematical Formulation of the Problem

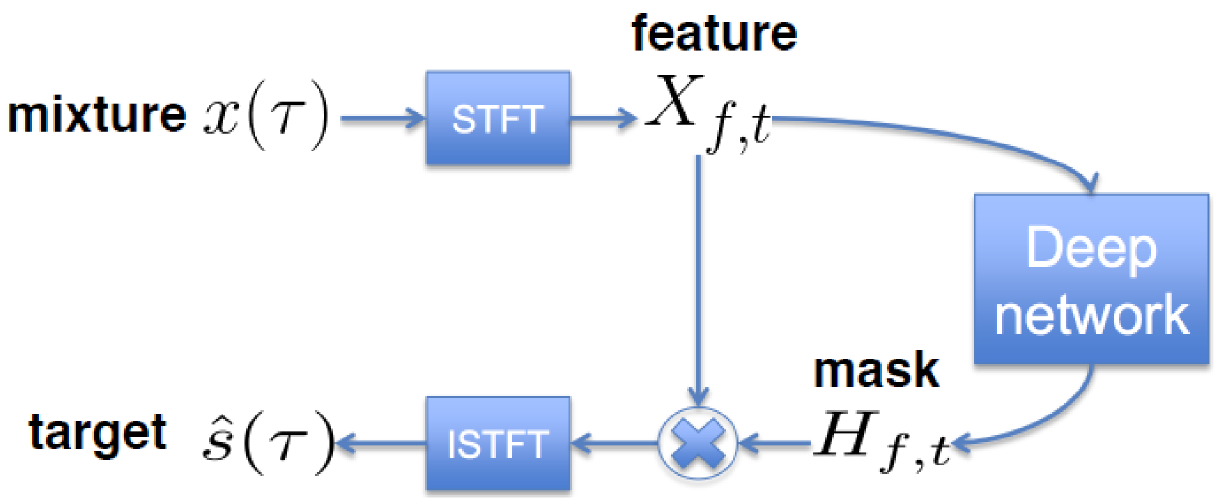
- **Mixed signal** $x(t) = \sum_{c=1}^{C} s_c(t)$
- **The goal of source separation is to estimate individual source signals** $s_c(t)$
- **Separation is performed on the time-frequency representations of signals obtained with short-time Fourier transformation (STFT)**
- **Given mixture** $X_i = \sum_{c=1}^{C} S_{c,i}$ **the task is to recover STFT spectral magnitudes of the source signals** $S_{c,i}$ **for all** c **and** i
  - $i = (t, f), i \in 1, .., N$ **is time-frequency bin**
  - $t \in 1, ..., T$ **is time frame**
  - $f \in 1, ..., F$ **is frequency bin**
  - $N = T \times F$
- **Problem: infinite number of possible combinations of** $S_{c,i}$ **that yield same** X_i
- **Solution: learn regularities between pairs of** $S_{c,i}$ **and** X_i **from training data**

Conventional methods

- **Computational Auditory Scene Analysis (CASA) [Hu & Wang 13]:**
  - **Spectral segmentation based on perceptual grouping cues**
  - **Advantage: no overfitting**
  - **Disadvantage: requires very careful tuning**
- **Spectral clustering [Bach & Jordan 06]:**
  - **Multiple kernel learning for approximating similarity matrices**
  - **Eigenvalue Decomposition (EVD) and k-means clustering**
  - **High complexity**
- **Non-negative matrix factorization (NMF) [Le Roux & Weninger^{+} 15]:**
  - **Dimensionality reduction technique that learns useful properties of sound**
  - **Requires modelling for each type of sound**
  - **Fails in speaker-independent conditions**

Motivation for Deep Learning

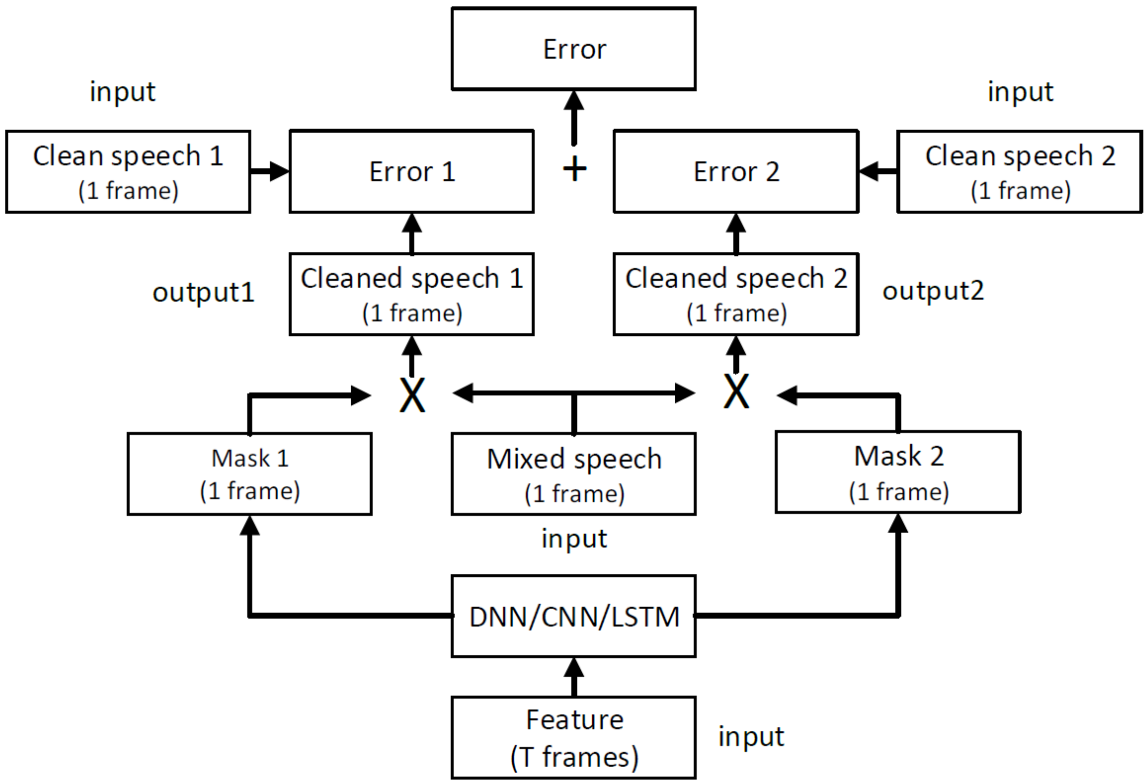
- Conventional methods suffer from strong applicability limitations and lack of generalization
- Artificial Neural Networks (ANNs) achieved significant results in other speech processing tasks such as recognition and enhancement
- State-of-the-art Speech vs. Noise separation is done in regression framework:



[Chen 15]

Multiclass regression attempt to perform two-talker speech separation

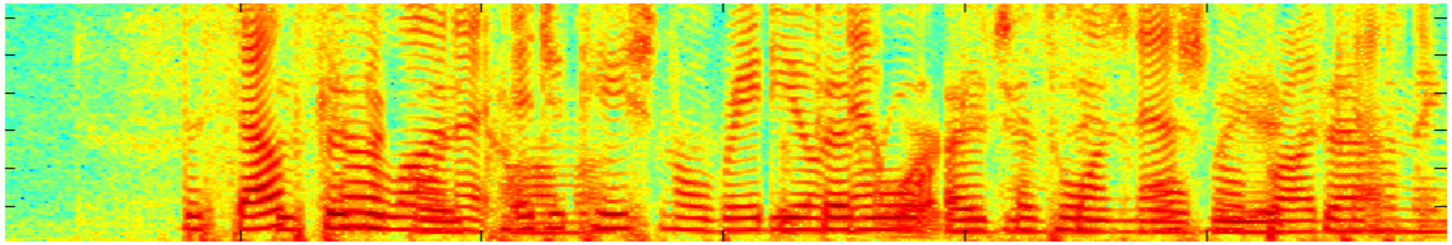
- **Given T frames of mixed speech ANN model $h_\theta(X)$ infers one frame t of the mask $H_c(t)$ for each of the sources c via MSE training criterion**
- **Reconstruction formula: $\tilde{S}_c(t) = H_c(t) \circ X(t)$**
  - **$\circ$ is element-wise multiplication**



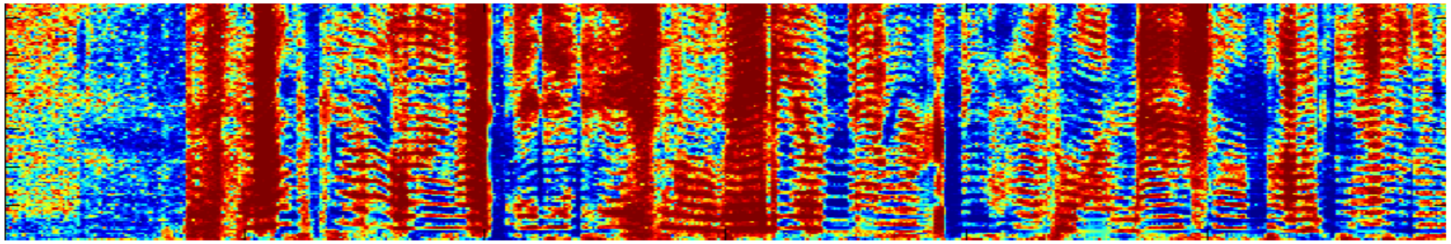
[Kolbæk & Yu+ 17]

Result

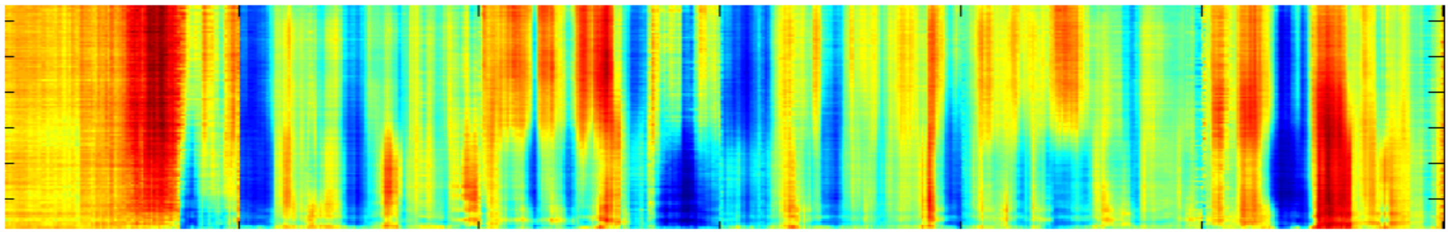
- Multiclass regression model with ANN fails to separate two-speaker mixture



Input mixture



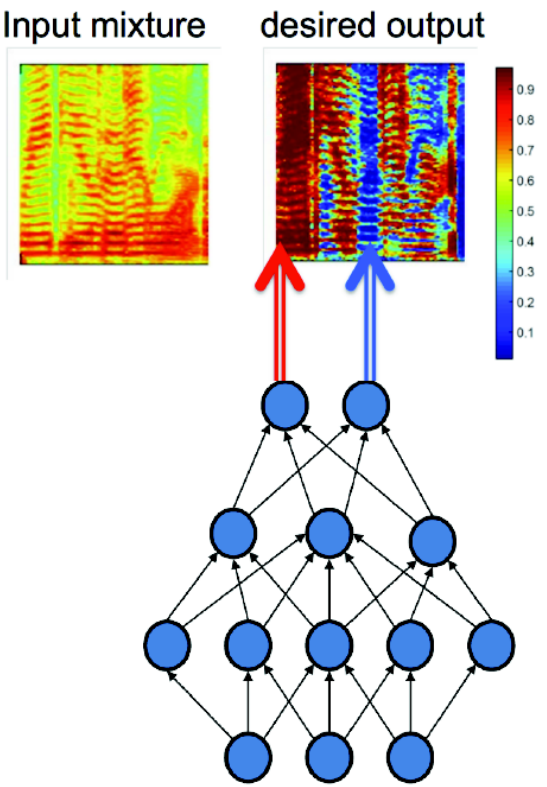
Oracle output



Regression output

[Chen 15]

Encountered Problems

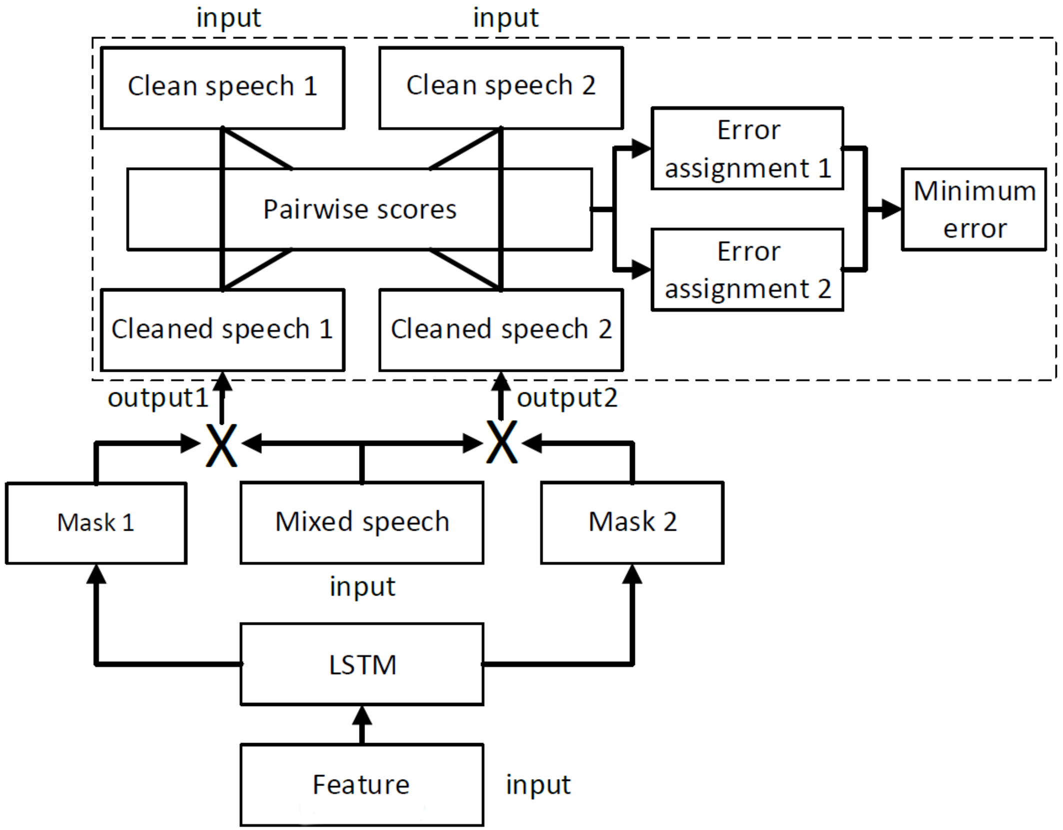


[Chen 15]

- **Permutation problem**
  - Order of sources is irrelevant: mixture A+B is described correctly by both permutations (A,B) and (B,A)
  - Which target output to use for each source?
  - Random assignment produces conflicting gradients in the training phase
- **Output dimension mismatch problem**
  - Cocktail party processor must separate speech signals belonging to arbitrary many sources
  - Fixed output dimension is not flexible to adapt to the arbitrary number of sources

Permutation Invariant Training [Yu & Kolbæk^{+} 16]

- **Solves permutation problem:**
  - **Represents reference streams in a set instead of an ordered list**
  - **Performs label assignment simultaneously with error evaluation**



adopted from [Yu & Kolbæk^{+} 16]

PIT Recipe

- **Compute C^2 pairwise mean squared errors (MSE) between each target source S_l and each reconstructed source $\tilde{S}_r$:**

$$J_{r,l} = \frac{1}{T \cdot F} \left\| \tilde{S}_r - S_l \right\|_F^2$$

  - $r, l \in 1, ..., C$

- **Construct a set of $C!$ possible assignments between target and estimated sources and estimate total error of each assignment $a \in \{1, ..., C!\}$:**

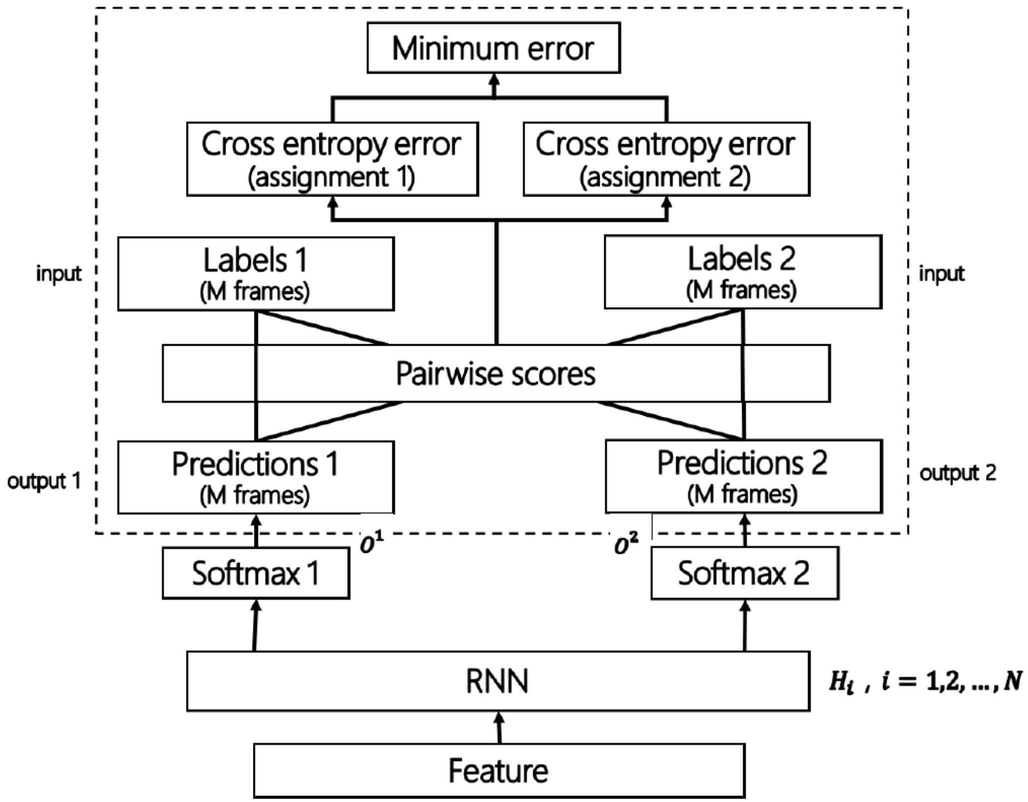
$$J_a = \frac{1}{C} \sum_{(r,t) \in a} J_{r,t}$$

- **Chose an optimal assignment to optimize network parameters**

$$a_{opt} = \arg\min_a J_a$$

PIT with ASR [Yu & Chang+ 17]

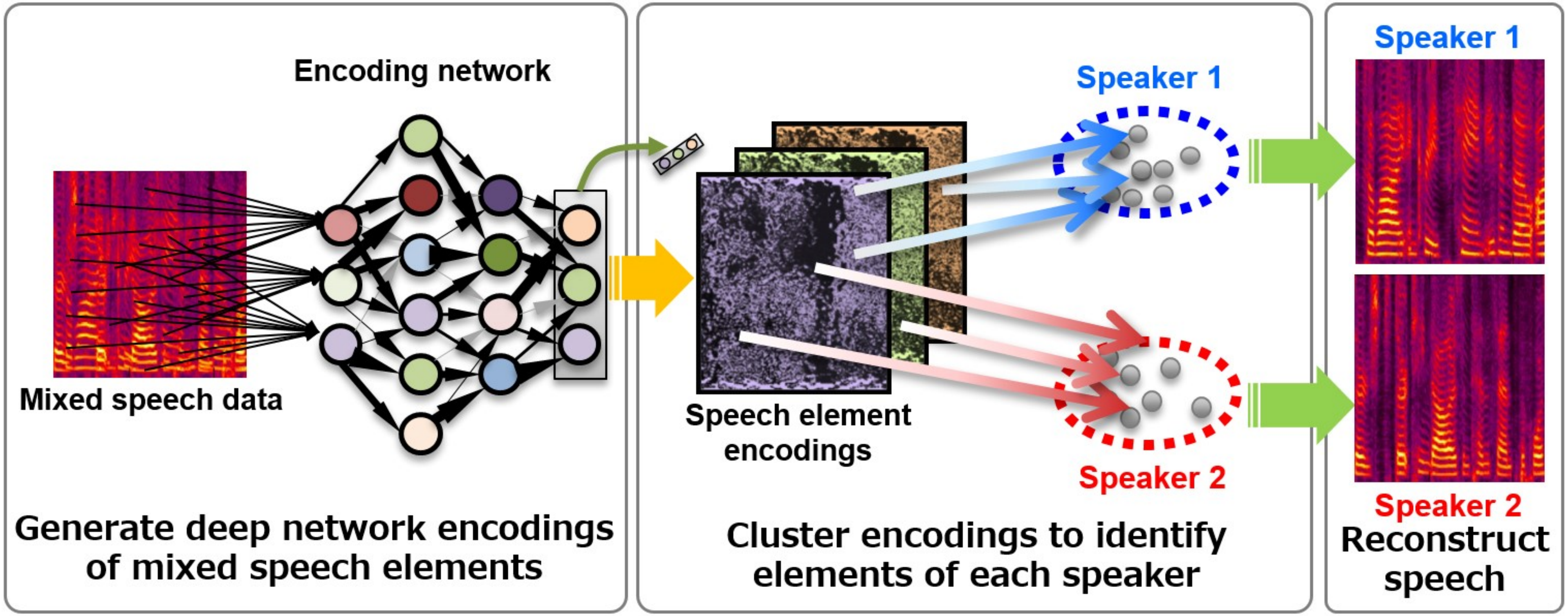
- PIT can be integrated into ASR system to recognize multi-talker speech
- Error between target and estimated senone posterior probabilities is minimized via cross-entropy (CE) criterion, separation is performed implicitly



[Yu & Chang+ 17]

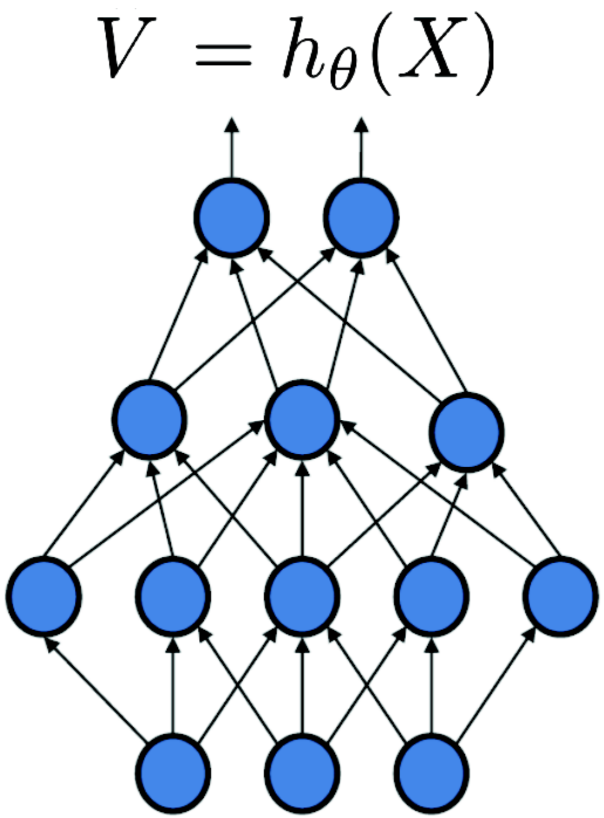
Deep Clustering [Hershey & Chen^{+} 16]

- First major success in the history of ANN-supported source separation
- Solves both permutation and output dimension mismatch problem



[DC]

DC Model Description



[Chen 15]

- Neural network $V = h_\theta(X)$ performs a mapping of global input signal $X \in \mathbb{R}^N$ into embedding space $V \in \mathbb{R}^{N \times D}$ with embedding dimension D and unit norm $|v_i| = 1$ for all i
- Resulting embedding is used to generate a $N \times N$ affinity matrix VV^T that represents similarity structure of input data

DC Training Recipe

- Partition-based objective function forces learned affinity matrix VV^T to match the target binary affinity matrix YY^T:

$$C_Y(V) = \left\| VV^T - YY^T \right\|_F^2$$

- $Y = \{y_{i,c}\}$ indicates a mapping between each time-frequency bin i and one of the C clusters c: $y_{i,c} = 1$ if $i \in c$ and $y_{i,c} = 0$ if $i \notin c$
- Therefore YY^T represents cluster assignments in permutation-independent way : $(YY^T)_{i,j} = 1$ if $i, j \in c$ and $(YY^T)_{i,j} = 0$ if $i \in c, j \in c'$ and $c \neq c'$
- Expanding Frobenius norm and applying polarization identity results in more intuitive formulation of the training criterion:

$$C_Y(V) = \underbrace{\sum_{i,j:y_i=y_j} (|v_i - v_j| - 1)}_{\text{pulls same cluster embeddings closer}} + \underbrace{\sum_{i,j} < v_i, v_j >}_{\text{pushes all embeddings apart}}$$

DC Evaluation Recipe

- During evaluation embeddings $V = h_\theta(\overline{X})$ are generated on the test mixture $\overline{X}$
- Rows $v_i \in \mathbb{R}^{\mathbb{D}}$ of the matrix V are clustered using k-means loss function:

$$\overline{\boldsymbol{Y}} = \arg\min_{\boldsymbol{Y}} \boldsymbol{K_V}(\boldsymbol{Y}) = \|\boldsymbol{V} - \boldsymbol{Y}\boldsymbol{M}\|_{\boldsymbol{F}}^2$$

- Means of the clusters are defined as $C \times D$ matrix $M = UA$:
  - Normalizer $U = (Y^T Y)^{-1}$
  - Accumulator $A = Y^T V$
- Inferred cluster assignments $\overline{Y}$ are used as binary masks that separate the mixture $\overline{X}$ into different sources

Deep Clustering with Enhancement Network [Isik & Roux$^+$ 16]

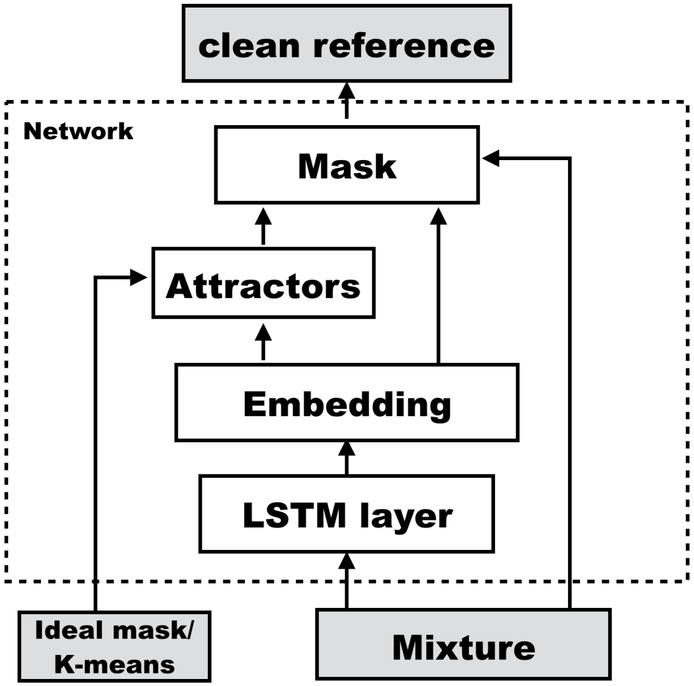
- **Problem: binary masks disregard features from weaker sources**
- **Solution: enhancement network on top of DC is a way to go**
- **DC is extended with following steps:**
  - **For each source c separated amplitude spectrogram $\hat{S}_c$ is concatenated with the mixture X and passed to enhancement network that outputs $z_{c,i}$**
  - **All outputs are normalized via softmax, yielding reconstruction masks: $H_{c,i} = e^{z_{c,i}} / \sum_{c'} e^{z_{c',i}}$**
  - **Enhanced separated signals are computed: $\tilde{S}_{c,i} = H_{c,i} \cdot X_i$**
- **Separation error is directly optimized by enhancement cost function:**

$$C_E = \min_{\pi \in \mathcal{P}} \sum_{c,i} \left(S_{c,i} - \tilde{S}_{\pi(c),i} \right)^2$$

  - **$\mathcal{P}$ represents all possible permutations on the set of sources $\{1, ..., C\}$**

Deep Attractor Network [Chen & Luo$^+$ 16]

- Problem: DC+ suffers from overcomplicated architecture and inefficient mapping between input signal and separated sources
- Solution: more efficient end-to-end training recipe for DC algorithm called Deep Attractor Network (DANet)



[Chen & Luo$^+$ 16]

- Biologically inspired by the Perceptual Magnet Effect
- Forms a perceptual magnet (attractor) for each source in the embedding space that draws together all TF bins belonging to this source
- Masks for the sources are estimated based on the similarity between TF bins and attractor points

DANet Training Recipe

- **Embedding generation with DC-related objective**

$$C_Y(V) = \left\| Y^T - MV^T \right\|_F^2$$

- **Attractor estimation**

$$A_{c,d} = \frac{\sum_i V_{d,i} \cdot Y_{c,i}}{\sum_i Y_{c,i}}$$

- **Mask estimation**

$$H_{c,i} = g(\sum_d A_{c,d} \cdot V_{i,d})$$

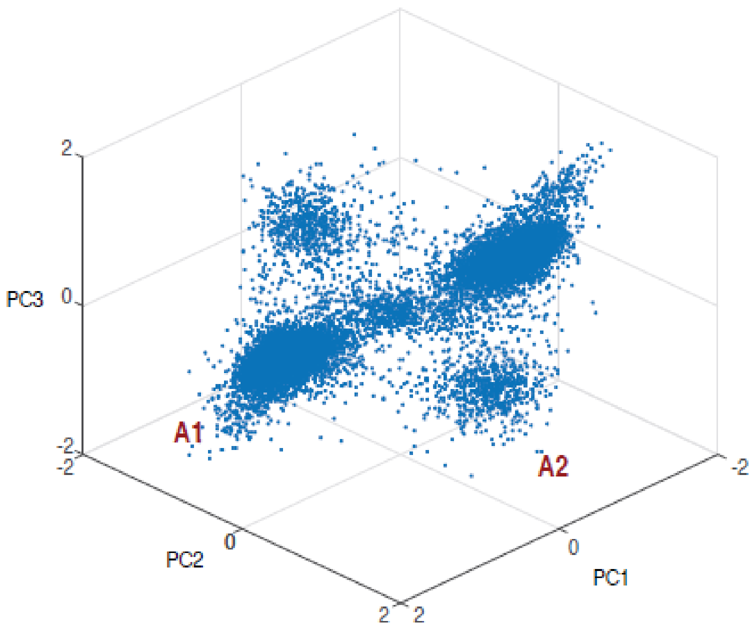
  - g **is sigmoid for two speaker separation and softmax for multi-speaker**

- **Separation error minimization**

$$L = \sum_{c,i} (S_{c,i} - H_{c,i} \cdot X_i)^2$$

DANet Inference Strategies

- With k-means as in DC:
  - Requires attractor estimation at test time
- With fixed attractors:



[Chen & Luo^{+} 16]

- Location of attractors in embedding space is relatively stable
- Two attractor pairs were learned by the algorithm on a set of 10,000 mixture examples
- Fixed attractor pair reduces DANet to a classification network
- Empowers real-time performance, but brings back permutation problem

Experimental setup

- Data is generated from the Wall Street Journal corpus by randomly mixing utterances from different speakers: WSJ0-2mix and WSJ0-3mix
- Most common source separation metric is signal-to-distortion ratio (SDR), measured in dB [Vincent & Gribonval^{+} 06]:
  - Defined as scale-invariant signal-to-noise ratio (SNR)
  - Compares the level of a desired signal to the level of an interfering signal and a background noise
- Ideal ratio mask (IRM) defines an upper bound performance achievable on this task
- All state-of-the-art methods employ BLSTM networks
  - Variable length of utterances
  - Long-range dependencies in the context

Evaluation results for two speaker separation

Method	SDR
Oracle NMF[1]	5.1
CASA [1]	3.1
DC [2]	9.1
fix-DANet [2]	9.5
DANet [2]	10.5
DC+ [3]	**10.8**
PIT [4]	10.0
IRM [4]	12.7

- **ANN-based approaches outperform conventional baselines by a large margin**
- **Most algorithmically complex DC+ achieves the best result**
- **fix-DANet compensates real-time implementation with slightly worse performance**

[1] [Hershey & Chen+ 16]
[2] [Chen & Luo+ 16]
[3] [Isik & Roux+ 16]
[4] [Kolbæk & Yu+ 17]

Evaluation results for three speaker separation

Method	SDR
Oracle NMF [1]	4.5
DC [2]	6.3
DC+ [3]	7.1
DANet [2]	**8.8**
PIT [4]	7.7
IRM [4]	12.8

- **DANet demonstrates the strongest generalization ability**
- **The most significant drop in performance is shown by DC+**
- **PIT performs better than both vanilla DC and DC+**

[1][Hershey & Chen+ 16]
[2][Chen & Luo+ 16]
[3][Isik & Roux+ 16]
[4][Kolbæk & Yu+ 17]

ASR experiments on speech separated with DC+

WER improvements on WSJ0-2mix [Isik & Roux^{+} 16]

Method	WER
Baseline	89.1
DC+	30.8
Clean	19.9

- Kaldi toolkit with GMM-based clean speech models was used to decode reconstructed streams
- Unprecedented performance gain in 63.2% relative WER

WER improvements on AMI mixed dataset achieved with PIT-ASR

[Yu & Chang+ 17]

Method	WER
Baseline	83.9
PIT-ASR	54.8
Clean	26.6

- **Two-talker dataset is generated from the AMI IHM corpus of meetings**
  - **80 hours of training data**
  - **8 hours of evaluation data**
- **Input features are 40-dimensional log filter banks**
- **Baseline setup includes acoustic model with 3-layer 512-unit BLSTM network and trigram language model**
- **PIT-ASR model contains 10 BLSTM layers with 768 hidden units in each layer**
- **Senone alignment is obtained with standard Kaldi model**
- **34.7% relative WER improvement, but still far from single-talker quality**

Future Work

- **Hierarchical clustering of the embeddings**
- **Attractor codebook for more challenging tasks**
- **Representative embeddings for robust attractor estimation**
- **Incorporating spatial information via beam-forming in multi-channel setup**
- **Passing LM down from the recognition to the separation stage and searching for the optimal recognized sequence across all speech streams**

Summary

- Historical perspective on source separation problem has shown it to be very challenging
- First attempts to apply ANNs failed due to permutation problem
- MERL's speech team revolutionized the field with deep clustering and restored the faith in feasibility of the cocktail-party problem
- ANN supported source separation has been studied extensively in the last two years
- Two proposed speech separation paradigms (DC and PIT) have their own merits and demerits: no clear winner yet
- Large number of possible applications
- Further developments of the methods is necessary to make them suitable for practical use

Thank you for your attention

Ilya Sklyar

ilya.sklyar@rwth-aachen.de

http://www-i6.informatik.rwth-aachen.de/

References

[Bach & Jordan 06] F.R. Bach, M.I. Jordan: Learning Spectral Clustering, With Application To Speech Separation. *Journal of Machine Learning Research*, Vol. 7, pp. 1963–2001, Dec. 2006. 8

[Chen 15] Z. Chen: Seminar Talk on Deep Clustering: Discriminative Embeddings for Segmentation and Separation. https://labrosa.ee.columbia.edu/cuneuralnet/chen111815.pdf, Nov. 2015. Accessed: 2017-06-18. 9, 11, 12, 17

[Chen & Luo^{+} 16] Z. Chen, Y. Luo, N. Mesgarani: Deep attractor network for single-microphone speaker separation. *CoRR*, Vol. abs/1611.08930, Nov. 2016. 5, 21, 23, 25, 26

[Cherry 57] C. Cherry: *On Human Communication: A Review, a Survey, and a Criticism*. Studies in communication. Technology Press of Massachusetts Institute of Technology, 1957. 3

[DC] Deep Clustering Model. http://www.merl.com/public/img/news/photo-1203.jpg. Accessed: 2017-06-18. 16

[Dem] Deep Clustering Demo. http://www.merl.com/demos/deep-clustering. Accessed: 2017-06-18. 4

[Hershey & Chen+ 16] J.R. Hershey, Z. Chen, J. Le Roux, S. Watanabe: Deep Clustering: Discriminative Embeddings for Segmentation and Separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 31–35, Shanghai, China, March 2016. IEEE. 5, 16, 25, 26

[Hu & Wang 13] K. Hu, D. Wang: An Unsupervised Approach to Cochannel Speech Separation. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 21, No. 1, pp. 122–131, Jan. 2013. 8

[Isik & Roux+ 16] Y. Isik, J.L. Roux, Z. Chen, S. Watanabe, J.R. Hershey: Single-Channel Multi-Speaker Separation using Deep Clustering. *CoRR*, Vol. abs/1607.02173, July 2016. 5, 20, 25, 26, 27

[Kolbæk & Yu+ 17] M. Kolbæk, D. Yu, Z. Tan, J. Jensen: Multi-talker Speech Separation and Tracing with Permutation Invariant Training of Deep Recurrent Neural Networks. *CoRR*, Vol. abs/1703.06284, March 2017. 6, 10, 25, 26

[Le Roux & Weninger+ 15] J. Le Roux, F.J. Weninger, J.R. Hershey: Sparse NMF – half-baked or well done? Technical Report TR2015-023, Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA, March 2015. 8

[Vincent & Gribonval$^+$ 06] E. Vincent, R. Gribonval, C. Fevotte: Performance Measurement in Blind Audio Source Separation. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, No. 4, pp. 1462–1469, July 2006. 24

[Yu & Chang$^+$ 17] D. Yu, X. Chang, Y. Qian: Recognizing Multi-talker Speech with Permutation Invariant Training. *CoRR*, Vol. abs/1704.01985, March 2017. 6, 15, 28

[Yu & Kolbæk$^+$ 16] D. Yu, M. Kolbæk, Z. Tan, J. Jensen: Permutation Invariant Training of Deep Models for Speaker-Independent Multi-talker Speech Separation. *CoRR*, Vol. abs/1607.00325, July 2016. 6, 13

PIT-ASR Training

- **Output of the BLSTM network with N layers $\mathbf{H}_N$ is used to compute C output layers of excitations for each source:**

$$\mathbf{H}_o^c = Linear(\mathbf{H}_N), c = 1, ..., C$$

- **Final C output layers with senone posterior probabilities for each stream c are computed via softmax:**

$$\mathbf{O}^c = Softmax(\mathbf{H}_o^c), c = 1, ..., C$$

- **Output senone probabilities $\mathbf{O}^c$ are compared with correct label sequences l_c via CE criterion:**

$$J = \frac{1}{C} \min_{c' \in permute(C)} \sum_c \sum_t CE(l_t^{c'}, \mathbf{O}_t^c), c = 1, ..., C$$

  - **Forces the system to choose label assignment with minimum loss**
  - **Computes the loss for each assignment on the whole utterance**

Objective function derivation

$$C_Y(V) = \left\| VV^T - YY^T \right\|_F^2 = \sum_{i,j}(< v_i, v_j > - < y_i, y_j >)^2 = \sum_{i,j:y_i=y_j} (< v_i, v_j > -$$

$$+ \sum_{i,j:y_i \neq y_j} < v_i, v_j >^2 = \sum_{i,j:y_i=y_j} (1 - 2 < v_i, v_j >) + \sum_{i,j} < v_i, v_j >^2$$

Polarization identity:

$$< v_i, v_j >= \frac{1}{2}(|v_i|^2 + |v_j|^2) - |v_i - v_j|^2)$$

Applying polarization identity to dot product $< v_i, v_j >$ leads to more intuitive formulation of training criterion:

$$C_Y(V) = \sum_{i,j:y_i=y_j} (|v_i - v_j| - 1) + \sum_{i,j} < v_i, v_j >$$

Efficient Implemenataion

- **Number of TF bins N is in order of magnitudes larger than embedding dimension D:**
  - **For a 10s audio file processed with 129-dimensional STFT and 10ms window $N = 129000$**
- **Low-rank nature of affinity matrix VV^T allows efficient implementation of the training criterion:**

$$\boldsymbol{C_Y(V) = \left\|VV^T - YY^T\right\|_F^2 = \left\|V^TV\right\|_F^2 - 2\left\|V^TY\right\| + \left\|Y^TY\right\|}$$

- **DC training criterion can be viewed as an efficient direct optimization of a low-rank affinity matrix in spectral clustering**

Deep Clustering with End-to-End Training

- Problem: joint training of embedding and enhancement networks is restricted by undifferentiable k-means clustering step in between
- Solution: substitute hard k-means clustering with a weighted EM algorithm with pooled covariances
  - Expectation step: soft assignment $\gamma_{i,c}$ of each embedding v_i to each cluster c:

$$\gamma_{i,c} = \frac{e^{-\alpha|v_i-\mu_c|^2}}{\sum_{c'} e^{-\alpha|v_i-\mu_{c'}|^2}}$$

    - α defines hardness of clustering
  - Maximization step: recomputation of means μ_c for each cluster with respect to assignments:

$$\mu_c = \frac{\sum_i \gamma_{i,c} w_i v_i}{\sum_i \gamma_{i,c} w_i}$$

    - $w_i = 0$ for silence and $w_i = 1$ for speech
- Steps of EM are unfolded in clustering network that enables gradient flow
- Final model is called DC+

Experimental setup

- Features are obtained with 129-dimensional magnitude STFT spectra
- All DC-based approaches share the same network architecture with 4 BLSTM layers with 600 units and one 2580-unit (20×129) feed-forward layer with embedding dimension $D = 20$
- DC+ and DANet are trained with curriculum learning
  - Pre-training on 100-frames segments
  - Fine-tuning on 400-frames segments
- DC+ employs feed-forward and recurrent dropout
- PIT details
  - 3 BLSTM layers with 896 units
  - Regularization via feed-forward dropout with rate control

Details on SDR and IRM

- **SDR**

$$SDR = 10\log_{10}\frac{\left\|s_{target}\right\|^2}{\left\|e_{interf}+e_{noise}+e_{artif}\right\|^2}$$

  - $\left\|s\right\|^2$ **is energy of the signal s**
  - s_{target} **is part of the signal coming from the wanted source** s
  - e_{interf} **is part of the signal coming from other unwanted sources**
  - e_{noise} **is part of the signal coming from sensor noise**
  - e_{artif} **is part of the signal coming from other causes**
- **IRM**

$$IRM_{c,i} = \frac{S_{c,i}}{\sum_{c\prime=1}^{C} S_{c\prime,i}}$$