RWTH Aachen University
Human Language Technology and Pattern Recognition Group
Prof. Dr.-Ing. Hermann Ney

Selected Topics in Human Language Technology and Pattern Recognition

ANN Supported Source Separation

Ilya Sklyar

Matriculation Number 362 027

23.07.2017

Supervisor: Tobias Menne

Contents

List of Tables

List of Figures

1 Introduction

The human brain is an extremely sophisticated system which has a vast amount of impressive properties that are usually taken for granted. One striking example is our ability to separate individual sound sources in an overlapped and interleaved mixture of sounds. The human auditory system can perform such source separation even in challenging acoustic environments, containing, for instance, music, speaking men and women, clinking glasses, footsteps, etc. Helmholtz was among the first scientists who described this phenomenon in his foundational work on acoustics and the perception of sound [36]. It was later formulated as a "cocktail party problem" by Cherry in 1957 [10]. Apart from describing human ability of following one speaker in the presence of others, Cheery also stated the absence of the machine that could perform this task. This interesting observation has been persistent for more than half of the century. Constructing the device that could segregate the mixture of sounds was especially challenging since we had no idea how human listeners perform this task. Understanding the processes behind the perceptual separation of sound sources has been addressed in many psychophysical studies.

In 1990, Bregman [1] introduced auditory scene analysis (ASA), the process of separating individual sounds in the human auditory system. The proposed framework inspired interest in the computational studies of source separation [37]. Many attempts to solve this problem were made in the last two decades, including both rule-based and machine learning (ML) approaches. Some researchers incorporated biological principles of speech perception to build heuristics for computational auditory scene analysis (CASA) [37, 20]. While being robust to over-fitting, these techniques require very careful tuning which makes them difficult to apply to hard problems. Other approaches relied on speaker-dependent models and employed ML algorithms such as non-negative matrix factorization (NMF) [25, 31] and factorial GMM-HMM [35] to learn models for speech streams. They achieved some success in the limited domain of speaker-dependent source separation. A clustering approach to the source separation featured spectral clustering [2] that was used to learn similarities between different parts of speech in the mixture. However, it suffered from the high computational cost of the spectral clustering paradigm.

Continuous rise of deep learning (DL) in other fields recently motivated researchers to apply artificial neural networks (ANNs) to the problem of source separation. As a result of these studies, two families of ANN-based source separation methods were developed. First family is based on the technique called deep clustering (DC) that employs representation power of an autoencoder for learning similarity structure of speech mixtures. Competing family of methods based on permutation invariant training (PIT) considers source separation as multi-class regression problem with permutation-invariant label assignment. Both approaches show comparable performance in separation experiments and form the state-of-the-art of the source separation. This report aims to describe these approaches in all comprehensive details.

Apart from the scientific challenge of building a system that could perform source separation, research in this direction is also motivated by a large number of possible applications. Some of them are listed below:

- **Virtual assistants.** In the recent years intelligent personal assistants have achieved significant level of recognition performance, especially in close-talking scenarios as in Google Now and Siri. Wide distribution of smart speakers (Google Home, Amazon Echo) increases possibilities of virtual assistants and introduces new challenges to the field. Nowadays these systems perform two-way conversations by means of anchor word detection (such as "Alexa") that distinguishes desired speaker in the presence of others [26]. To enable the scenario of multi-party human-machine interactions, which is crucial in the world of Internet of things (IoT) [44], such systems must act as full cocktail party processors.

- **Hearing aids.** About 1.1 billion people are affected by hearing loss to some degree. For many of them hearing aids are the only appropriate treatment. Apart from the amplification of incoming audio signals, such systems (e.g. cochlear implants) must be capable to perform auditory scene analysis in the complex scenarios, such as speech overlapping and noisy environments [37].
- **Automatic meeting transcription.** Work group effectiveness is a big issue in the industry today. Endless meetings dedicated to discussing same things over and over again waste a lot of time that could be spent in more productive way. New meeting browsing technologies can provide a better way to manage and organise group discussions [7]. Developing systems that could perform automatic meeting transcription would be an important step towards this goal.
- **Automatic captioning for audio/video recordings.** A huge volume of audio and video content is available on the Internet today. Automatic transcription of such multimedia content (e.g. Youtube videos) would provide more diverse user experience and more effective search possibilities. Since audio and video recordings often contain mixture of sounds, the separation of speech mixtures is required for the reliable transcription of these files.

For more formal definition and mathematical formulation of the source separation problem please refer to Section 2 of the current report. Section 3 is dedicated to an overview of the main DL building blocks used in the state-of-the-art source separation algorithms. In Section 4 several conventional methods for source separation are discussed. The rest of the paper covers main part of this report. Section 5 introduces main problems from which suffered previous DL-based approaches to source separation and describes how current state-of-the-art methods address these issues. Subsequent Section 6 compares these methods in terms of their architectures and performance. Finally, Section 7 summarizes all key points of the report and provides discussion on the possible future work.

2 Source Separation Problem

2.1 Definition

The source separation problem has been described in many different ways in the literature. One of the most general definitions was given by Cardoso in [6], where this problem was determined as the blind signal separation (BSS). BSS is the task of recovering a set of individual unobserved signals or "sources" from a set of the observed mixtures of signals. The important adjective "blind" here points to the fact that no information about the mixture is available and, hence, no assumptions about underlying sources can be made. While BSS can take into account different types of multidimensional data, such as images or tensors, only blind audio source separation (BASS) [34] is concerned here.

To further restrict the discussed problem it is highly important to mention that this report only considers speaker-independent multi-talker speech separation. This constraint thus corresponds to the adjective "blind" in the definition of BASS, implying that no assumptions about the number of speakers or their properties can be made. In addition, all approaches presented below use only single microphone signals, hence making beamforming inapplicable to this problem. In the subsequent sections the term source separation will be used as a shortening for the term single-microphone speaker-independent multi-talker speech separation. This task was shown to be the most challenging speech separation task and can be also referred to the general cocktail party problem, discussed in Section 1.

2.2 Mathematical Formulation

Given the above definition of the source separation this task can be described with the following mathematical formalism. The goal of source separation is to recover C individual source signals $s_c(t), c = 1, ..., C$ from the input mixed signal $x(t)$:

$$x(t) = \sum_{c=1}^{C} s_c(t) \tag{1}$$

Since speech separation is normally performed in the time-frequency (TF) domain, $x(t)$ is processed with short-time Fourier transformation (STFT). As a result, a set $X = \{X_i\} \in \mathbb{R}^{F \times T}$ of F-dimensional spectral magnitudes for all time frames T is obtained, where $i = (t, f)$ is TF bin for time frame t and frequency f. The task of the source separation system is to estimate STFT spectral magnitudes of the source signals $S_{c,i}$ for all c and i given X_i:

$$X_i = \sum_{c=1}^{C} S_{c,i} \tag{2}$$

Unfortunately, this problem is ill-posed since there are infinite number of possible combinations of individual spectral magnitudes $S_{c,i}$ that yield same mixed spectral magnitude X_i. However, other ill-posed problems occurred in computer vision and speech recognition have been successfully attacked with different ML algorithms, especially DL models. Following the general supervised ML recipe, some DL model $f_\theta(X)$ with parameters θ can be trained to learn regularities between target source vectors $S_c \in \mathbb{R}^N$ and input mixture vector $X \in \mathbb{R}^N$ from training data. At test time parameters θ are kept fixed and model $f_\theta(\bar{X})$ is used to estimate individual sources $\tilde{S}_c$ given a test mixture $\bar{X}$.

3 Deep Learning Basics

The following chapter gives a high level overview of basic DL concepts and corresponding references, since an in depth discussion is out of the scope of this report.

3.1 Deep Feedforward Networks

Deep feedforward networks (also denoted as deep neural networks or DNNs) are essential building blocks in DL. They are very powerful mathematical models which aim to map some input x to an output $f(x; \theta)$ by learning parameters θ. In other words, they find parameters θ such that $f(x; \theta)$ is the best approximation of some function f^* [12]. This function can describe many real-world tasks, e.g. classification of objects on a image, transcription of a spoken sentence or translation of a written sequence. This approximation power of DNNs accounts for their current success in a variety of different applications.

DNNs compose together many different functions (called neurons, nodes or units) in a directed acyclic graph manner. Output of each unit y_j can be represented as a weighted sum of N input units x_i followed by a non-linear function g:

$$y_j = g(\sum_{i=1}^{N} x_i w_i + b), \tag{3}$$

where b is bias term and w_n is corresponding weight.

Inside DNNs, information flows only in one direction, from input x through hidden units to output $f(x;\theta)$ without any loops. Hidden units in the network are organized in layers. Stacking more layers results in more abstract, high-level representations of input data. The adjective "deep" thus symbolizes the ability of DNNs to represent functions with increasing complexity by adding more layers in the network.

3.2 Recurrent Neural Networks and LSTMs

Recurrent Neural Networks (RNNs) are important family of neural networks used for sequence modelling. RNNs take as an input sequence of observation $x^{(0)}, .., x^{(t)}$, thus making advantage of the continuous nature of many forms of data, such as speech or text [12]. Sequence processing is achieved by the means of loop connection inside of a neural network. Unrolling the loop connection introduces multiple copies of the same neural network in different hidden states $h^{(0)}, .., h^{(t)}$, where information passes not only from input nodes to output nodes, but also to successor nodes, as shown in Figure 1. This structure enables the connection between the previous information and the current task. Hidden state h_t of a RNN is calculated according to the formula:

$$h_t = g(Wx_t + Uh_{t-1} + b), \tag{4}$$

where W denotes weights between hidden states and inputs and U denotes weights between adjacent hidden states.

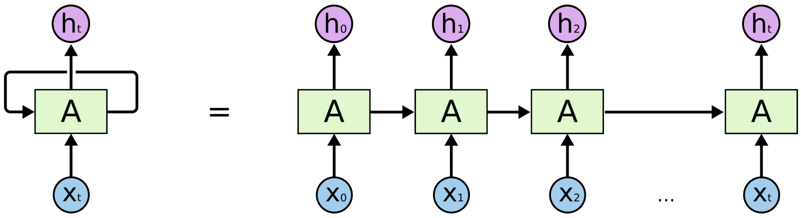

Figure 1: Unrolling a RNN [28]

However, vanilla RNNs have issues with modelling long-term dependencies due to the vanishing gradient problem [3]. To overcome this drawback, more sophisticated type of RNN was proposed, called Long Short Term Memory networks (LSTMs) [19]. They introduce specific repeating module with four different layers that enforces the network to remember information from the past. Each LSTM module consists of one memory cell C_t, three sigmoid gate layers and one tanh layer $\tilde{C}_t$. Memory cell C_t works as a conveyor belt that enables activation and error flow along the module. The input gate i_t and the output gate o_t define amount of activation that is added and deleted from the memory cell at the time step t. The forget gate f_t controls how much information about previous steps is retained at the current time step. To reserve place for future updates, tanh layer

$\tilde{C}_t$ creates a vector of new candidate values. This architecture is presented in Figure 2. For more detailed explanation of the ideas behind LSTMs please refer to [28].

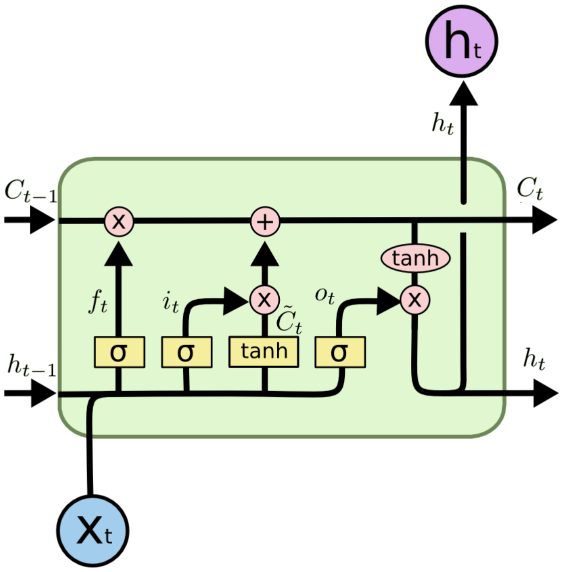

Figure 2: Architecture of an LSTM module [28]

LSTM networks successfully solve the vanishing gradient problem by enabling the gradient flow in the memory cell. They are especially useful for exploring long-term dependencies in input data. To further enhance this ability, Bi-directional LSTM (BLSTM) networks were proposed to propagate information in both time directions. This property of BLSTM networks allows to find better connections in the context which leads to better performance in some tasks.

3.3 Autoencoders

An autoencoder is an ANN that maps original data to reconstructed data obtained from a hidden representation. It contains a hidden layer h (also denoted as coding or code layer) which describes an efficient coding of the input data x: $h = f(x)$. Function h is called an encoder function and is trained to capture useful properties of the data. The second part of an autoencoder is a decoder function $r = g(h)$ that aims to reconstruct the original data. However, perfect reconstruction of the dataset is not the goal of the autoencoder network. On the contrary, it is trained to be unable to just copy the input to output, but to extract useful information about data distribution instead [12]. The general architecture of an autoencoder is presented in Fig. 3.

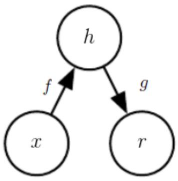

Figure 3: The general architecture of an autoencoder [12]

Autoencoders have been studied extensively in the literature in the recent years. Several ways were proposed to obtain useful representations in a hidden layer h. One possible solution is to restrict the coding to have smaller dimension than the original data. This type of autoencoder is called undercomplete. It learns a low-dimensional manifold that represents the principal sub-space of the training data. However, when complexity of encoder and decoder functions is high, undercomplete autoencoder fails to learn salient features of the data and just performs useless identity function. To overcome this problem another family of models - regularized autoencoders - were developed. They provide the ability to choose the coding dimension and model flexibility with respect to complexity of given data.

One of the most popular regularized autoencoders is sparse autoencoder. It introduces a sparsity penalty $\Omega(h)$ on a code layer h that forces a model to learn unique statistical properties of the dataset in addition to an identity function. Training criterion of a sparse autoencoder will thus include both reconstruction error and sparsity penalty:

$$L(x, g(f(x))) + \Omega(h) \tag{5}$$

Another way to force an autoencoder to learn useful features is to directly change reconstruction error. Denoising autoencoders (DAE) minimize objective function

$$L(x, g(f(\widetilde{x}))) \tag{6}$$

where input to the network $\widetilde{x}$ is copy of original input x corrupted by noise. The goal of autoencoder is therefore to recover the original input data from the corrupted signal. To achieve this goal DAE implicitly learns underlying structure of the data distribution.

Autoencoders are heavily used in practice for representation learning tasks, such as dimensionality reduction and information retrieval. Historically, dimensionality reduction was among the first successful applications of autoencoders and DL in general [17]. Recently they are also studied in the scope of different unsupervised learning problems like clustering [21]. Applicability of autoencoders to spectral clustering is particularly important for the task of source separation, since it is often treated as a segmentation problem.

4 Conventional Methods

In the following subsections several state-of-the-art conventional methods for source separation are described. Although most of them suffer from domain limitations and lack of generalization they still show descent performance in some tasks when properly tuned. Moreover, they provide good historical perspective on the problem and inspire modern ANN-based approaches.

4.1 Computational Auditory Scene Analysis

In the last two decades many attempts have been made to solve the challenging problem of source separation. One of the most popular family of approaches of the past was computational auditory scene analysis (CASA). These techniques are based on the idea of building a machine system that could perform ASA by adopting (to some extent) the biological principles of processing sound in human auditory system [37]. Similar to ASA, most CASA approaches divide the task of source separation into two stages: segmentation and grouping. During segmentation input auditory signal is decomposed into time-frequency (TF) regions. The grouping stage then combines these regions to streams that correspond to sound sources. These streams are later used to generate a mask that separates the sources. Segmentation rules in CASA methods are often manually designed based on perceptual Gestalt grouping cues [41]. These hand-crafted similarity features are usually based on proximity in time and frequency of the signal, such as common amplitude and frequency modulation or harmonicity using pitch tracking. Therefore CASA methods are rule-based, which makes them more robust to overfitting. However, they require very careful tuning, which causes strong limitations in their areas of application.

In this report a state-of-the-art CASA-system developed in [20] is included as a baseline. It introduces a novel unsupervised approach to cochannel speech separation. This method utilizes two different strategies for voiced and unvoiced speech in the mixture. For separation of the voiced speech it firstly performs simultaneous grouping to organize components of sound across frequency axis and then clusters resulting streams to sequentially group them into two speakers across time. Unvoiced speech is segregated through onset/offset analysis and produced segments are grouped based on already segregated voiced speech, in case of unvoiced-voiced overlaps, or just splitted equally into two speakers for portions of unvoiced-unvoiced speech. This approach has demonstrated impressive performance in separating cochannel speech, but has been never generalized to multi-speaker mixtures.

4.2 Spectral Clustering

Spectral clustering is a state-of-the art clustering method that partitions data points into disjoint clusters based on the eigenstructure of a similarity matrix. It performs an Eigenvalue Decomposition (EVD) on the normalized Laplacian matrix and then represents the original data through k eigenvectors that correspond to the k smallest non-zero eigenvalues of the Laplacian matrix. Pre-defined k thus corresponds to the number of clusters in data, and running k-means on the resulting representation solves the clustering problem. [32]

Spectral clustering has gained huge popularity in machine learning community in the last decades due to its elegant mathematical derivation and global optimum solution. It was successfully applied to speech separation of two-speaker mixture in [2]. Their method combines several perceptual grouping affinity features, motivated by CASA and psychophysical properties of speech. Multiple kernel learning works as a front-end processor that approximates similarity matrices for these features. Spectral clustering is applied to the similarity matrices to generate separated speech. The resulting segmentation is presented in Fig. 4.

Despite the success of spectral clustering approach in separating equal-strength mixture of two speakers, it has never been extended to more general conditions. It can be explained by high computational cost of EVD decomposition and shallow learning of similarity matrices. Moreover, both factors are interdependent since simple kernels yield sparse affinity matrices that require more expensive spectral methods to produce clusters [15]. This problem discouraged further research in the clustering approaches to speech separation due to its tricky chicken and egg nature. It can be actually described as Catch-22 paradox. To produce reliable segmentation one needs to take into account more neighbouring features of the same source. However, these context regions are overlapped with features belonging to other sources. Therefore, front-end processor should possess segmentation beforehand to select only relevant features for future segmentation, which leads to a logical contradiction. Solving this problem with an ANN is a main idea of DC approach, described in 5.1.1.

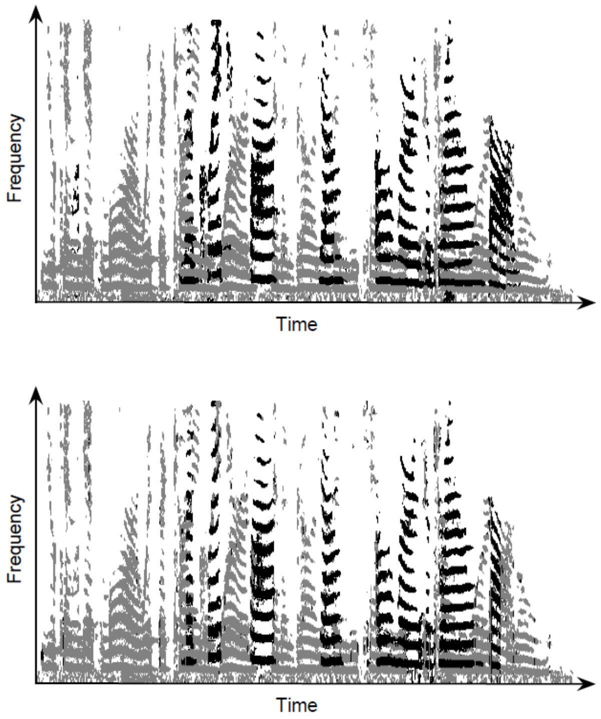

Figure 4: Two-speaker (black and grey) segmentation of the mixture spectrogram obtained with spectral clustering (bottom) and the corresponding optimal segmentation (top) [2].

4.3 Non-negative Matrix Factorization

Another important family of methods that proved their ability to attack source separation problem in limited set of domains is model-based methods. Two most popular approaches from this family are worth mentioning in this report: NMF and factorial GMM-HMM. The former is a dimensionality reduction technique which learns a set of non-negative basis functions and their activation coefficients that are used to reconstruct each source during evaluation [25]. The latter exploits factorial HMM to model the target and competing sources and their temporal dynamics [35]. Both approaches make several assumptions about the input signal and, hence, are fundamentally incapable of solving general cocktail-party problem. However, sparse NMF (SNMF) approach is considered as the baseline in the closed condition source separation experiments in [15, 44], therefore it is described here more broadly.

NNF has been applied to a wide of range of different tasks for analysing non-negative data. It is normally used as dimensionality reduction technique which aims to split non-negative matrix $\mathbf{M} \in \mathbb{R}^{F \times T}$ into two non-negative factors $\mathbf{W} \in \mathbb{R}^{F \times R}$ and $\mathbf{H} \in \mathbb{R}^{R \times T}$, where inner dimension R is in the order of magnitude smaller than original dimensions F and T. Hence, sparsity penalty on R, similar to autoencoder, forces $\mathbf{W}$ and $\mathbf{H}$ to learn meaningful properties of the original data as a by-product of factorization. In the context of source separation, data $\mathbf{M}$ is a matrix of F-dimensional spectral magnitudes for all time frames T in the mixture. Each source $c \in \{1, ..., C\}$ is represented as a product of a matrix $\mathbf{W}^c$, containing R_c non-negative basis functions and a matrix of activations $\mathbf{H}^c$ for all time frames T. A common way to solve the source separation problem is then to learn $\mathbf{W}^c$ from data during training and estimate activation matrices $\mathbf{H}^c$ at test time. Optimal activation matrices $\hat{\mathbf{H}} = [\hat{\mathbf{H}}^1; ...; \hat{\mathbf{H}}^C]$ are obtained via formula:

$$\hat{\mathbf{H}} = \arg\min_{\mathbf{H}} L(\mathbf{M}, \mathbf{W}\mathbf{H}) + \lambda \left\|\mathbf{H}\right\|_1, \tag{7}$$

where L is a loss function that is minimized when $\mathbf{M} = \mathbf{W}\mathbf{H}$ and $\lambda \left\|\cdot\right\|_1$ denotes L^1 sparsity penalty with weight λ. This approach is called supervised NMF [25].

5 Deep Learning Based Techniques for Source Separation

ANNs have recently taken a vast amount of computational fields by storm. They are quintessential part of DL approaches, that have shown impressive effectiveness especially in AI-complete tasks, e.g. processing of images [24] and speech [18]. It became possible due to more efficient training algorithms [4], deeper architectures [14] and sophisticated node structures [19]. In spite of several successes achieved by DL-approaches in different speech processing tasks such as end-to-end automatic speech recognition [8, 13], robust speech recognition [40, 27] and raw speech feature extraction [11, 30], the problem of source separation has remained unresolved. Researchers encountered two main difficulties while attacking the source separation problem with DL.

The first difficulty is so called ”permutation problem” [15]. It arises from the fact that the order of sources in the mixture is unknown and irrelevant. For instance, both (A,B) and (B,A) solutions describe the correct source separation for mixture A+B. However, neural networks are by definition trained to map an input data to a unique target output. Training with multiple target labels for each sample will lead to serious convergence issues due to conflicting gradients.

The second issue is ambiguity of number of sources in the mixture, also referred to ”output dimension mismatch problem” [9] General formulation of source separation problem implies that cocktail party processor must be able to separate speech signals belonging to arbitrary many sources. However, if the size of output layer in the neural network is fixed, it will be unable to adapt to the arbitrary number of sources, thus violating one of the main principles of source separation.

In the last years different approaches were proposed to overcome these difficulties. They can be split into two families: deep clustering and permutation invariant training. Both of them are described in corresponding subsections.

5.1 Deep Clustering

Deep clustering (DC) was originally proposed by Hershey et al. [15] in 2015, making it a first major success in the history of ANN-supported source separation. This method brought back the interest to the field of source separation by showing how DL could be effectively applied to this long-lasting problem. It is based on generating discriminative embeddings for each time-frequency bin that approximate an ideal similarity matrix between the sources. Hence, this method combines both spectral clustering and autoencoder frameworks due to similarities of what they optimize [32]. It also employs k-means clustering to generate the source assignments at test time. For detailed model description of the original DC approach see Section 5.1.1. A more advanced version of DC algorithm, named Deep Clustering with End-to-End training, was developed to enable joint training of the DC network with enhanced network for signal reconstruction. It is presented in section 5.1.2. Another type of DC named deep attractor network (DANet) was proposed in [9]. It extends the idea of deep clustering by introducing biologically inspired attractor points in the embedding space. For the relevant description please take a look at Section 5.1.3.

5.1.1 Deep Clustering Model Description

DC treats source separation as a clustering problem. It represents raw input signal x as a collection of complex spectrogram values X_i in each TF bin $i = (t, f), i \in \{1, .., N\}$, where $N = F \times T$. The goal of clustering is therefore to find a partition of i into subsets of bins that belong to particular sources. Moreover, these subsets are assumed to be disjoint, implying that each time-frequency bin gets a hard assignment to the speaker that dominates in the corresponding bin. Resulting partition allows to infer binary masks for each of the source used for extracting separated features.

As mentioned in Section 4.2, the clustering approach to source separation with shallow learning methods suffers from high complexity due to Catch-22 paradox. On the contrary, deep ANNs have shown their tremendous ability to alleviate this problem by learning useful representations of data. Following recent successes of autoencoders producing unsupervised embeddings for clustering algorithms [32, 21], authors of DC proposed to use an ANN $V = f_\theta(X) \in \mathbb{R}^{N \times D}$ to directly generate a $N \times N$ lower-rank affinity matrix VV^T. In other words, ANN $f_\theta(X)$ performs a mapping of global input signal $X \in \mathbb{R}^N$ into embedding space $V \in \mathbb{R}^{N \times D}$ where D is embedding dimension. Embedding is constrained to have unit norm, such that $|v_i| = 1$ for all elements i.

This transformation enables the encoding to implicitly learn similarity structure of input data in permutation-independent way. It is achieved by introducing objective function $C_Y(V)$ that forces learned affinity matrix VV^T to match the target binary affinity matrix YY^T:

$$C_Y(V) = \left\| VV^T - YY^T \right\|_F^2 \tag{8}$$

Here $Y = \{y_{i,c}\}$ 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 $y_{i,c} = 0$ otherwise. Such one-hot encoding enables target binary affinity matrix YY^T to represent 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 $c \neq c'$. Since the order of the sources c, c' is completely disregarded, this formulation successfully solves long-lasting permutation problem. Moreover, the number of clusters C does not affect training objective $C_Y(V)$, hence also avoiding the necessity to specify number of sources at the stage of neural network training.

Form 8 can be expanded further using Frobenius norm definition:

$$\begin{aligned} C_Y(V) = \left\| VV^T - YY^T \right\|_F^2 = \sum_{i,j} (< v_i, v_j > - < y_i, y_j >)^2 = \sum_{i,j: y_i = y_j} (< v_i, v_j > -1)^2 \\ + \sum_{i,j: y_i \neq y_j} < v_i, v_j >^2 = \sum_{i,j: y_i = y_j} (1 - 2 < v_i, v_j >) + \sum_{i,j} < v_i, v_j >^2 \end{aligned} \tag{9}$$

Applying polarization identity to dot product $< v_i, v_j >$ in Equation 9 leads to more intuitive formulation of training criterion:

$$C_Y(V) = \sum_{i,j: y_i = y_j} (|v_i - v_j| - 1) + \sum_{i,j} < v_i, v_j >, \tag{10}$$

where the first sum pulls encodings of elements i and j closer if they belong to the same cluster and the second pushes all encodings apart to avoid trivial solution.

One can also optimize the original training criterion from Equation 8 for efficient implementation:

$$C_Y(V) = \left\|V^T V\right\|_F^2 - 2\left\|V^T Y\right\| + \left\|Y^T Y\right\|, \tag{11}$$

where expensive estimation of $N \times N$ affinity matrix is avoided by constructing low-rank $D \times D$ matrix instead. Therefore DC training criterion can be viewed as an efficient direct optimization of a low-rank affinity matrix in spectral clustering.

During evaluation embeddings $V = f_\theta(\bar{X})$ are generated on the test mixture $\bar{X}$. The rows $v_i \in \mathbb{R}^{\mathbb{D}}$ of the matrix V are then clustered using k-means loss function:

$$\bar{Y} = \arg\min_Y K_V(Y) = \|V - YM\|_F^2 \tag{12}$$

Means of the clusters are defined as $C \times D$ matrix M:

$$M = UA, \tag{13}$$

where the normalizer $U = (Y^T Y)^{-1}$ is a $C \times C$ diagonal matrix that contains the inverse number of elements in each cluster $\frac{1}{c_i}$ on the main diagonal and the accumulator $A = Y^T V$ is a $C \times D$ matrix that sums all embeddings belonging to each cluster.

Inferred cluster assignments $\bar{Y}$ are used as binary masks that separate the mixture $\bar{X}$ into different sources $\tilde{S}_c$:

$$\tilde{S}_c = \bar{Y}_c \circ \bar{X}, \tag{14}$$

where $\circ$ denotes element-wise multiplication. True cluster labels $\mathring{Y}$ obtained from the binary mask with optimal SNR are used for estimating clustering error by the means of different statistical measures (e.g. chi-squared test χ^2):

$$d_{\chi^2}(\bar{Y}, \mathring{Y}) = \left\|\bar{Y}\bar{U}\bar{Y}^T - \mathring{Y}\mathring{U}\mathring{Y}^T\right\|_F^2 \tag{15}$$

It is easy to notice that all three objective functions (training loss from Equation 8, k-means loss from Equation 12 and clustering error from Equation 15) are optimized when learned affinity matrix matches ideal binary affinity matrix: $VV^T = \mathring{Y}\mathring{Y}^T$, resulting in $\bar{Y} = \mathring{Y}$.

While this approach to source separation problem provides comprehensive mathematical foundation and successfully avoids both ANN-related caveats, it also has some limitations, such as binary nature of inferred masks. Hard assignment of TF bins to dominating sources actually leads to the loss of the features belonging to competing sources. This problem is addressed in the subsequent section.

5.1.2 Deep Clustering with End-to-End Training

Isik et al. [22] proposed two main improvements to the standard DC architecture. Firstly, in order to recover missing sources in regions prevailed by stronger sources, second-stage enhancement network was built on top of the DC model. For each source c separated amplitude spectrogram $\hat{S}_c$ obtained from the DC clustering algorithm is concatenated with the amplitude spectrogram of the original input signal X and passed to the enhancement network. The network contains one BLSTM and one feedforward layer and produces output Z_c. Outputs from all sources are combined via softmax function that generates final mask for each TF bin i:

$$M_{c,i} = e^{Z_{c,i}} / \sum_{c'} e^{Z_{c',i}} \tag{16}$$

Final mask is then applied to the original input signal, yielding enhanced separated signals: $\tilde{S}_{c,i} = M_{c,i} X_i$. They are optimized during training by enhancement cost function

$$C_E = \min_{\pi \in \mathcal{P}} \sum_{c,i} \left(S_{c,i} - \tilde{S}_{\pi(c),i} \right)^2, \tag{17}$$

where $\mathcal{P}$ represents all possible permutations on the set of sources $\{1, ..., C\}$ and $S_{c,i}$ is target separated source. This formulation thus allows to directly optimize separation error, instead of optimizing affinities VV^T.

However, the ability to perform this optimization in the end-to-end manner, jointly with the DC objective function, is restricted by two factors. First, binary masks obtained from the original DC algorithm cannot be optimized directly with enhancement network since optimal signal separation is usually continuous. Second, direct optimization of both networks is impossible due to undifferentiable hard clustering operation between these networks. Therefore authors of [22] proposed to substitute hard k-means clustering used in the original method [15] with soft weighted k-means algorithm. This algorithm can be viewed as a weighted expectation maximization (EM) algorithm for a Gaussian mixture model with pooled covariances between the clusters.

Weights w_i for each embedding v_i are introduced in this clustering framework in order to disregard the silence regions. They are set to 1 for TF bins with significant energy and to 0 in silence regions. EM clustering is divided into two stages. At expectation step soft assignment $\gamma_{i,c}$ of each embedding v_i to each cluster c is performed:

$$\gamma_{i,c} = \frac{e^{-\alpha |v_i - \mu_c|^2}}{\sum_{c'} e^{-\alpha |v_i - \mu_{c'}|^2}}, \tag{18}$$

where parameter α defines hardness of clustering. At maximization step mean μ_c for each cluster is recomputed with respect to these assignments:

$$\mu_c = \frac{\sum_i \gamma_{i,c} w_i v_i}{\sum_i \gamma_{i,c} w_i}, \tag{19}$$

Steps of EM algorithm are organized as layers in clustering network using deep unfolding framework, presented in [16].

Final model, referred at the rest of the paper to DC+, consists of three networks. First network is the ANN that learns discriminative embeddings from input complex spectrogram values, as presented in 5.1.1. Unfolded EM-clustering network then performs clustering on the output of the first network. Third network takes inferred masks from the second network as input and generates final enhanced separated sources, optimized by separation error function 17. Since gradient is guaranteed to pass throughout the whole model, it can be trained end-to-end with the standard back-propagation algorithm. However, this approach was reasonably criticized due to its complicated architecture and inefficient mapping between input signal and separated sources [9, 44].

5.1.3 Deep Attractor Network

More efficient end-to-end training recipe for DC algorithm was proposed by Chen et al. in [9]. Their approach is biologically inspired by the well-known phenomenon in human speech perception called perceptual magnet effect. It describes an ability of human brain to warp stimulus space in such a way that speech signals near category prototype (perceptual magnets) of some sound are drawn together, resulting in general representation of the corresponding sound. Following this idea, DANet was developed to form a perceptual magnet (attractor) for each source in the embedding space that draws together all TF bins belonging to this source. Similarity between TF bins and attractor points in embedding space is then used to estimate masks for all sources in the mixture. Finally, these masks are applied to the original mixture and resulting separated sources are compared with the targets. Mask learning thus enables more efficient end-to-end training routine than in previous DC-based approaches.

Training phase of DANet, as mentioned above, consists of three main parts:

- **Attractor estimation.** Here embeddings V are firstly generated from the given mixture using the standard DC network with a slight variation in the objective function:

$$\begin{aligned} C_Y(V) &= \left\|VV^T - YY^T\right\|_F^2 = \left\|UY^TYY^T - UY^TVV^T\right\|_F^2 \\ &= \left\|Y^T - UAV^T\right\|_F^2 = \left\|Y^T - MV^T\right\|_F^2, \end{aligned} \tag{20}$$

  where UY^T is multiplied to both terms, yielding an objective that minimizes discrepancy of the similarity between each embedding vector and cluster mean MV^T with target binary mask Y^T.

  Given computed embeddings V attractors $A \in \mathbb{R}^{C\times D}$ for each of the sources are then estimated according to the centroid calculation formula:

$$A_{c,d} = \frac{\sum_i V_{d,i} \cdot Y_{c,i}}{\sum_i Y_{c,i}} \tag{21}$$

- **Mask estimation.** Next, separation masks $M \in \mathbb{R}^{N\times C}$ for all sources are computed by finding the similarity between each attractor and each embedding in embedding space. This metric can be represented as inner product followed by an activation function:

$$M_{c,i} = g(\sum_d A_{c,d} \cdot V_{i,d}), \tag{22}$$

  where g is sigmoid for 2-speaker mixtures and softmax for 3 speakers and more.

- **Separation error minimization.** When separations masks are estimated, one can finally optimize separation error using standard $L2$ reconstruction loss:

$$L = \sum_{i,c} \|S_{c,i} - X_i \cdot M_{c,i}\|_F^2 \tag{23}$$

  This training criterion takes into account actual difference between clean spectrogram values and masked signal, thus encouraging the network to find better separation.

For inference, attractor estimation procedure is different due to the absence of target assignments Y. Authors of the method proposed two strategies for estimating attractor points at test time. First strategy is similar to the approach used in the original DC algorithm, requiring to perform clustering (e.g. k-means) on the embeddings V. System architecture of DANet, combining both training recipe described above and evaluation procedure with k-means, is presented in Fig. 5.

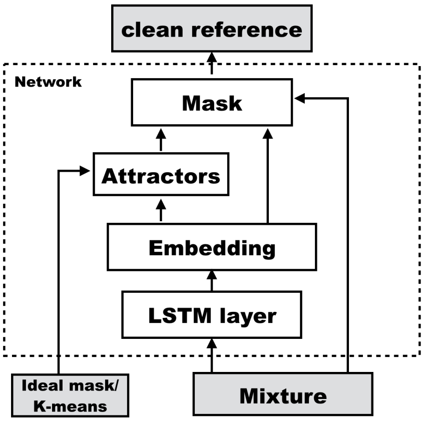

Figure 5: DANet system architecture with k-means as evaluation procedure [9].

Second inference strategy is more interesting. It is based on the discovery that was made while training DANet on the set of 10,000 mixture examples. Projecting resulting attractor points into 3-dimensional space using Principal Component Analysis showed that the location of attractors in embedding space is relatively stable. In particular, two pairs of attractors A1 and A2 were learned by the algorithm, as shown in Figure 6. It confirmed an intuition that DANet has an ability to implicitly discover different number of attractor points in an unsupervised manner. This property can be helpful during evaluation, when one can use pre-set generalized attractor pair obtained from training phase instead of performing costly k-means algorithm. It reduces DANet architecture to a simple classifier with attractor points as output labels and empower real-time performance of the algorithm. However, it brings back the permutation problem, since mapping between masks and attractors is not unique. This problem can be solved with PIT, which is described in the next section.

5.2 Permutation Invariant Training

Except DC and DANet, previous attempts to solve source separation with ANNs were made in multi-class regression framework. Unfortunately, most of them failed due to already discussed permutation problem. To overcome this problem, Permutation Invariant Training (PIT) algorithm was proposed by Yu et al. in [44]. This approach is described in Section 5.2.1 under the name of segment-based PIT (or PIT-S) due to the way it handles input-output meta-frame mapping. Enhanced version of PIT, called PIT with speaker tracing (PIT-T)[23] is presented in Section 5.2.2 Finally, Yu et al. in [43] recently integrated PIT framework into automatic speech recognition (ASR) system, thus enabling direct recognition of multiple streams of speech. For the comprehensive description please see Section 5.2.3.

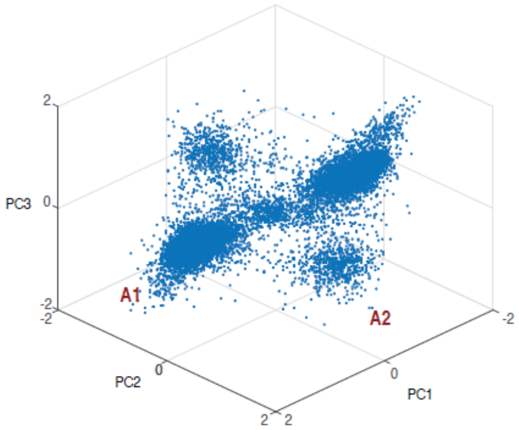

Figure 6: Location of attractors in the embedding space[9].

5.2.1 Segment-based Permutation Invariant Training

In the recent years significant progress has been made in the field of speech enhancement with class-based methods [38, 42, 39]. Following this success, source separation was also formulated as multi-class regression problem. In this framework network is trained on parallel sets of mixtures and their corresponding target sources, learning to predict the source belonging to the class reference for each TF bin. This architecture for two-talker separation is demonstrated in Figure 7.

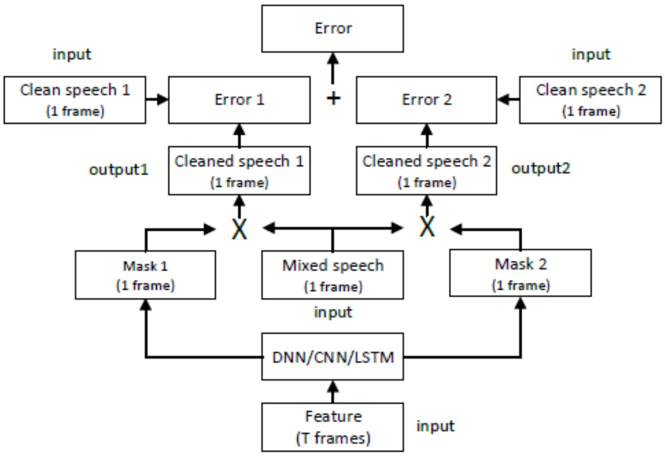

Figure 7: Two-talker speech separation as conventional multi-class regression problem [23].

Input features to the system are presented as a meta-frame with contextual information of T successive spectral magnitudes of the mixture $X \in \mathbb{R}^{F\times T}$, where F is the number of frequency bins. Specific ANN model $f(X;\theta)$, such as DNN or RNN is used to infer one frame t of the mask $M_c(t)$ for each of the sources c. Obtained masks are then utilized to build one frame of reconstructed one-source spectral magnitude for each talker: $\tilde{S}_c(t) = M_c(t) \circ X(t)$.

To learn parameters θ of the network $f(X;\theta)$ during training correct magnitudes S_1 and S_2 need to be compared with the corresponding output layers of the network $\tilde{S}_1$ and $\tilde{S}_2$. However, mapping between target and generated magnitudes is not unique and not known by the system. It forces the network to randomly assign reference to the output, leading to label permutation problem. This problem is especially vivid for training sets with many number of speakers.

Solution to this problem, illustrated in Fig. 8, is based on representing reference streams in a set instead of an ordered list, as it was before in conventional approach. Since the order of sources in the set is irrelevant, same result is obtained for all possible orderings of sources. It is achieved with permutation invariant building on the ANN output, as demonstrated in the dashed rectangular in Fig. 8.

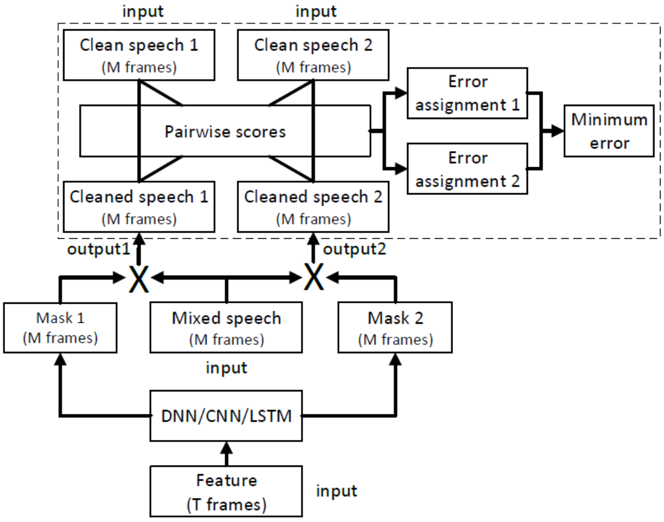

Figure 8: PIT for two-talker speech separation [23].

PIT strategy performs label assignment simultaneously with error evaluation. First, it computes C^2 pairwise mean squared errors (MSE) between each target source S_c and each reconstructed source $\tilde{S}_c(t)$:

$$J_{r,t} = \frac{1}{T \cdot F} \left\| \tilde{S}_r - S_t \right\|_F^2, \tag{24}$$

where $r, t \in \{1, ..., C\}$. Then a set of $C!$ possible assignments between target source t and estimated sources s is constructed and total error of each assignment a is estimated:

$$J_a = \frac{1}{C} \sum_{(r,t)\in a} J_{r,t} \tag{25}$$

Finally, assignment with the least error a_{opt} is chosen to optimize network parameters:

$$a_{opt} = \arg\min_a J_a \tag{26}$$

Hence, PIT directly minimizes the optimal separation error J_{opt} among all possible permutations.

Another distinctive feature of PIT algorithm is an ability to specify dimensions of output and input windows M and T. Input T frames of mixed speech produce M frames of source labels, where input meta-frame is shifted by one or several frames, thus shifting output meta-frame as well. Since output meta-frames have overlapping regions, output-to-speaker assignments for the same frames in different output meta-frames can be different. While the original segment-based PIT makes an assumption that they do not change, it was shown to be suboptimal. For this reason authors of [23] integrated speaker tracing algorithm in the original PIT approach to further improve separation performance.

5.2.2 Permutation Invariant Training with Speaker Tracing

In general, speaker tracing aims to assign each frame of the network's output to a specific source. Main idea of the specific speaker tracing approach presented here is based on the investigation that perfect source reconstruction is obtained when all frames belonging to the same speaker are aligned to the same output layer. To achieve this goal, two main changes to the original PIT technique were made.

First, segmentation of input and output streams needs to be omitted to avoid generation of overlapping output masks, which caused different across-meta-frames assignments in the first place. Speaker tracing integration forces the network to make the decision of speaker assignment based on the whole utterance of dimension N, not on the M-frame segment of it. Input meta-frame dimension T should thus also be changed to utterance length N.

Second, such restriction on the feature dimension introduces new challenges, such as high variability in the utterance length and the necessity to operate with long-term dependencies between contextual information. However, LSTMs showed their impressive ability to attack these problems. Therefore, they are applied here as building blocks of the ANN that infers masks out of the presented mixed speech. Moreover, input to the network does not need to include contextual frames no more, since LSTMs implicitly learn dependencies in the whole context, which leads to the input layer with only F units.

Separation performance with PIT can be further improved by stacking second LSTM network on top of the results obtained with the first system $M_c^{(1)} = h^{(1)}(X)$ combined with mixed speech X: $M_c^{(2)} = h^{(2)}(X, M_c^{(1)} \circ X)$, where h denotes an LSTM layer. Final reconstruction mask is then averaged between outputs of two systems:

$$M_c = \frac{M_c^{(1)} + M_c^{(2)}}{2} \tag{27}$$

5.2.3 Permutation Invariant Training with ASR

Effective training algorithm of PIT avoids extra clustering step of DC approaches by elegantly solving both permutation and speaker tracing problems at once. It motivated researchers to integrate PIT algorithm directly in the standard ASR pipeline. Significant progress in this direction was made in [43], where MSE criterion of the original PIT was substituted with cross-entropy (CE) criterion that directly minimizes error between target and estimated senone (tied states of context-dependent phones) posterior probabilities. Their model for two-talker speech recognition (PIT-ASR) is presented in Fig. 9.

Proposed architecture consists of N stacked BLSTM layers $\mathbf{H}_i, i = 1, ..., N$. First LSTM layer $\mathbf{H}_1 = h(X)$ takes feature representation X of the mixed speech x as input.

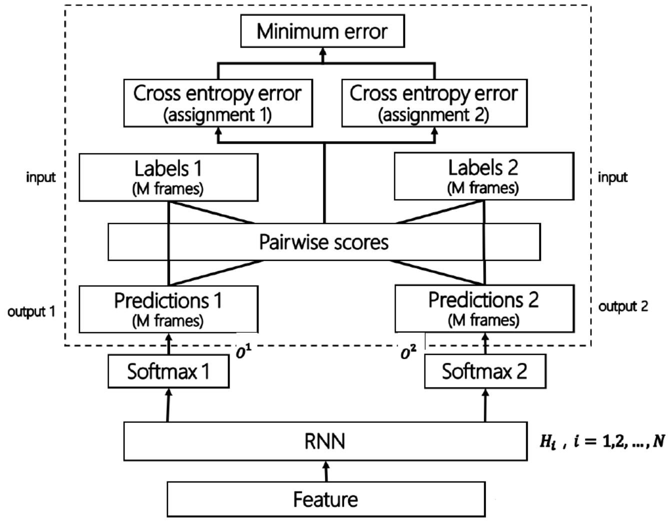

Figure 9: PIT-ASR model for two-talker speech recognition [43].

Output of the last layer $\mathbf{H}_N = h(\mathbf{H}_{N-1})$ is used to compute C output layers of excitations for each source:

$$\mathbf{H}_o^c = q(\mathbf{H}_N), c = 1, ..., C, \tag{28}$$

where q denotes linear transformation. Excitations are then normalized via softmax, resulting in final C output layers with senone posterior probabilities for each stream c:

$$\mathbf{O}^c = g(\mathbf{H}_o^c), c = 1, ..., C, \tag{29}$$

where g is softmax. Once senone probabilities for each stream $\mathbf{O}^c$ are computed by the network, they need to be compared with corresponding correct label sequences l_c. Here label ambiguity problem comes in place. Moreover, speaker tracing has to be performed, since all posteriors probabilities within one layer need to be assigned with the same speaker. Fortunately, in previous sections PIT has shown its ability to deal with similar problems in speech separation framework. Here PIT also comes to the rescue, but it is augmented with different optimization criterion:

$$J = \frac{1}{C} \min_{c' \in \mathcal{P}(C)} \sum_c \sum_t CE(l_t^{c'}, \mathbf{O}_t^c), c = 1, ..., C, \tag{30}$$

where $\mathcal{P}(C)$ denotes a permutation of C sources. This objective function takes into consideration both label assignment problems. First, it forces the system to choose for optimization label assignment with minimum loss, disregarding ordering of the labels. Second, it computes the loss for each assignment on the whole utterance to ensure that speaker assignments are consistent within one layer. Hence, this model manages to perform direct recognition of mixed speech by implicitly separating signals in permutation-free way.

6 Evaluation

This chapter is dedicated to evaluation and comparison of the described ANN-based approaches to source separation. Evaluation takes into account speech separation experiments in two- and three-speaker conditions. In addition, ASR performance for separated streams obtained by different methods is evaluated. Experimental setup includes description of the datasets used in the experiments and configurations of competing methods.

6.1 Experimental Setup

Hershey et al. in the original DC paper [15] created a new corpus of 2 speaker mixed speech utterances: WSJ0-2mix. It consists of 30 hours of training, 10 hours of validation and 5 hours of evaluation data generated from the Wall Street Journal corpus by randomly mixing utterances from different speakers at signal-to-noise ratios (SNR) in the diapason between 0 dB and 10 dB.

Baselines used in evaluation include 2 conventional approaches: supervised sparse NMF [25] provided with oracle information about the speakers and unsupervised CASA-based system [20]. Competing ANN-supported source separation approaches include DC, DC+, DANet, PIT-S and PIT-T. Their configurations are listed below. Ideal ratio mask (IRM) result is provided as a reference for upper bound performance achievable on this task[23]. All methods were evaluated using the most common source separation metric: signal-to-distortion ratio (SDR), measured in dB [34]. It compares the level of a desired signal to the level of an interfering signal and a background noise. In all separation experiments input features were obtained with 129-dimensional magnitude Short-time Fourier transform (STFT).

All DC-based approaches compared here share the same embedding network architecture between each other. Network contains 4 BLSTM layers with 600 hidden units in each layer followed by one feed-forward layer with 2580 units ($20 \cdot 129$), corresponding to the embedding dimension $D = 20$ and number of STFT spectral coefficients $S = 129$. Experiments were conducted with RMSprop optimization algorithm [33] in all three cases. Other training details for DC-based approaches are more method-specific.

While DC network was trained on non-overlapping windows of 100 frames, authors of DC+ proposed more efficient training recipe for this task, a specific type of the curriculum learning [5]. They observed that shorter segments of speech provide better starting point for training due to increasing diversity of the batch. However, since network has to perform inference on the whole utterance at test time, feeding the network with longer segments of speech in the end yields better results. Therefore they firstly pre-trained the network on 100-frames segments and fine-tuned it on segments of length 400 afterwards. This strategy was applied by authors of DANet as well. Enhancement network was built on top of the embedding network in DC+, consisting of 2 BLSTM layers each with 300 hidden units. Tricks of the trade for training both networks included feed-forward and recurrent dropout and gradient normalization.

Two different configurations for DANet are presented here for comparison with other approaches. It follows the dichotomy of evaluation strategies described in Section 5.1.3. First model, simply labelled as DANet, performs k-means on the output of the embedding network. Second one (fix-DANet) uses a fixed pair (A1 from Fig. 6) of attractor points obtained from the training phase.

Permutation invariant family of separation methods is presented here with two models: segment-based PIT-S and PIT-T with speaker tracing. In PIT-S input and output meta-frames have length of 51. Network configuration of PIT-S contains 11 convolution layers, one pooling layer and one 1024-unit ReLU layer. Output of the model is C layers with 6579 ($129 \cdot 51$) hidden units in each layer. PIT-T model has 3 BLSTM layers with 896 hidden units in each layer. Its input and output layers both have 129 units. Regularization

Table 1: Evaluation results for two speaker separation

Method	**SDR**
Oracle NMF [15]	5.1
CASA [15]	3.1
DC [9]	9.1
fix-DANet [9]	9.5
DANet [9]	10.5
DC+ [22]	**10.8**
PIT-S [44]	7.6
PIT-T [23]	10.0
IRM [23]	12.7

was applied via feed-forward dropout with rate control. Second stage network was applied on top of the original model, following the approach described in Section 5.2.2.

6.2 Results

Table 1 summarizes experimental results on WSJ0-2mix dataset obtained with described separation methods. Several important observations can be made based on the presented results. First, all listed ANN-based approaches outperform conventional baselines by a large margin. Second, the most algorithmically complicated DC+ achieves the best performance among all other approaches, holding the current state-of-the-art result in 2-speaker source separation. However, DANet yields comparable result, while being less complex than DC+. DANet with pre-set attractors compensated effective real-time implementation with slightly worse performance. The original DC method demonstrated the worst performance among all DC-based approaches, as expected. In general, DC-based method compared favourably with PIT-based algorithms. Experiments have confirmed the idea that segment-based approach to PIT drastically lacks correct output-to-speaker assignment, resulting in relatively poor performance. Introducing speaker tracing into PIT routine brings a significant improvement of 2.4 dB.

Table 2 summarizes experimental results on WSJ0-3mix dataset, constructed from WSJ corpus following the same principle as in WSJ0-2mix but with 3 speakers. CASA, fix-DANet and PIT-S are excluded from the comparison since they were not evaluated in three-speaker conditions. As shown in the table, separation quality degrades for all approaches on several dB in more complex auditory environments. The most significant drop in performance (3.7 dB) is shown by DC+, leaving it behind DANet and PIT-T in this experiment. DANet, on the contrary, demonstrated the strongest generalization ability (1.7 dB decrease in SDR) among all methods, achieving the best overall performance. PIT-T compares favourably with both simple DC and complex DC+, yielding descent result with relatively small performance drop of 2.3 dB.

Only two of the described approaches were evaluated in terms of their ASR performance: DC+ and PIT-ASR. Authors of DC+ used Kaldi toolkit [29] with GMM-based clean speech models to decode reconstructed streams. Achieved result in terms of word error rate (WER) is reported in Table 3 along with results of the corresponding noisy baseline and clean speech [22]. This outcome represents an unprecedented performance gain in 63.2% relative WER. However, there is still a significant margin by ASR performance on clean speech.

Table 2: Evaluation results for three speaker separation

Method	**SDR**
Oracle NMF [15]	4.5
DC [9]	6.3
DC+ [22]	7.1
DANet [9]	**8.8**
PIT-T [23]	7.7
IRM [23]	12.8

Table 3: WER improvements on WSJ0-2mix dataset separated with DC+ [22]

Method	**WER**
Baseline	89.1
DC+	30.8
Clean	19.9

In the contrast to DC+, PIT-ASR is able to perform speech recognition directly on mixed speech. To evaluate their approach Yu et al.[43] artificially generated new two-talker dataset based on the AMI IHM corpus of meetings [7]. New corpus consists of 80 hours of training and 8 hours of evaluation data with speech mixture of two different speakers under 5 different signal-to-noise (SNR) conditions: 0dB, 5dB, 10dB, 15dB and 20dB. Baseline setup for the original AMI corpus includes acoustic model with 3-layer 512-unit BLSTM-RNN and trigram language model. PIT-ASR configuration contains 10 BLSTM layers with 768 hidden units in each layer. Both baseline and PIT-ASR use 40-dimensional log filter bank features as input and senone alignment obtained with standard Kaldi [29] LDA-MLLT-SAT-GMM-HMM model as target labels. Final recognition results are presented in Table 4. They show impressive 34.7% relative WER improvement of the PIT-ASR compared to the baseline. However, this result is insufficient for any practical use and needs further development.

7 Conclusion and Discussion

In this report an overview of the current state-of-the art methods in ANN supported source separation was performed. Historical perspective on this problem has shown it to be very challenging. Conventional approaches to source separation used to make too many assumptions about underlying properties of the mixed speech streams, which led to poor generalization ability. Following the rise of DL, researchers tried to attack this issue with the representation power of ANNs, but faced source permutation problem. It was successfully solved by integrating source separation problem into DC framework in [15].

Table 4: WER improvements on AMI mixed dataset achieved with PIT-ASR [43]

Method	**WER**
Baseline	83.9
PIT-ASR	54.8
Clean	26.6

This method dramatically outperformed conventional baselines and restored the faith in feasibility of the cocktail-party problem. It provoked further research in the field of ANN supported source separation, which resulted in impressive number of publications on the topic in the last year. Some of them were dedicated to enhance the initial DC approach, as DANet [9] and DC+ [22], while the others formed competing family of methods based on PIT (PIT-S [44], PIT-T [23], PIT-ASR [43]). Their comparison revealed that both families have their own merits and demerits.

All DC-based methods manage to solve both permutation and output dimension mismatch problems during training. However, they require prior knowledge about number of sources at test time to perform k-means clustering algorithm. Original DC approach also suffers from suboptimal binary mask inferring, leading to poor results in three-speaker separation experiments. DC+ demonstrates the overall best separation performance on WSJ0-2mix dataset among all methods and performs quiet well in the recognition of the separated sources. On the other hand, its excessive algorithmic complexity and inefficient stepwise pre-training discourages from integrating it in a multi-speaker recognition pipeline. Here DANet can be a helpful extension to the standard DC algorithm since it introduces more efficient end-to-end training recipe with attractor points estimation. This method has an ability to implicitly discover arbitrary many attractors in an unsupervised way. This property makes DANet highly applicable to more complex separation scenarios, which results in the first place in three-speaker separation task. Moreover, fixing attractor points during evaluation can enable efficient real-time separation without clustering step, thus avoiding the main drawback of the DC-based separation school. These factors make DANet the most promising DC-based approach. It would be very interesting to investigate its performance in speech recognition experiments and in three-speaker separation tasks with fixed attractor points. Further improvements of this method can be achieved with [9]:

- picking representative embeddings for robust attractor estimation via attention encoder-decoder network
- hierarchical clustering of the embeddings where better representation of audio signal is achieved with hierarchical grouping of sources
- generating attractor codebook for more challenging tasks and using it for real-time separation experiments

In contrast to DC-based approaches, PIT methods deal only with label permutation problem, leaving output dimension mismatch problem aside. Hence, their network architectures depend on the pre-set number of sources in the mixture. However, it is compensated by algorithmically simpler training and evaluation recipe. Unsatisfactory performance of the segment-based PIT-S underlines the importance of speaker tracing for this separation approach. PIT-T naturally integrates it to the separation pipeline and avoids additional clustering step. Its performance is slightly worse in comparison to DC approaches, but this can be explained by the lack of regularization, i.e. recurrent dropout. Simple model architecture of PIT allows it to be directly integrated in multi-talker speech recognition framework, as it is done in PIT-ASR. It performs speech separation implicitly and yields recognized sequences for each speaker. While achieved WER is still far away from the single-talker recognition quality, it can be further improved by a variety of ways [43]:

- finer feature extraction guideline: STFT spectral magnitudes instead of log filter bank features
- passing language model information down from the recognition to the separation stage and taking into account all speech streams when searching for the optimal recognized sequence
- extending single microphone source separation to multi-channel setup with beamforming technique for better speaker tracing

To sum up, possible future work on ANN-supported source separation seems very promising. After all these approaches being listed, solving cocktail party problem does not look as fantastic as it was before. Concerning many possible areas of application, it is desirable that more academic and industrial research groups would be involved in improving the state-of-the-art of source separation. Literature research conducted in this report can be useful as a starting point in this direction.

References

[1] A.S. Bregman. *Auditory scene analysis: The perceptual organization of sound.* The MIT Press, 1990.

[2] Francis R. Bach and Michael I. Jordan. Learning spectral clustering, with application to speech separation. *J. Mach. Learn. Res.*, 7:1963–2001, December 2006.

[3] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Trans. Neur. Netw.*, 5(2):157–166, March 1994.

[4] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. pages 153–160, 2007.

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA, 2009. ACM.

[6] Jean-Francois Cardoso. Blind signal separation: Statistical principles, 2003.

[7] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Maël Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain A. McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The ami meeting corpus: a pre-announcement. Idiap-RR Idiap-RR-82-2005, IDIAP, 0 2005.

[8] W. Chan, N. Jaitly, Q. Le, and O. Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964, March 2016.

[9] Zhuo Chen, Yi Luo, and Nima Mesgarani. Deep attractor network for single-microphone speaker separation. *CoRR*, abs/1611.08930, 2016.

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

[11] Pavel Golik, Zoltán Tüske, Ralf Schlüter, and Hermann Ney. Convolutional neural networks for acoustic modeling of raw time signal in LVCSR. In *Interspeech*, pages 26–30, Dresden, Germany, September 2015.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[13] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1764–1772, 2014.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[15] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 31–35. IEEE, 2016.

[16] John R. Hershey, Jonathan Le Roux, and Felix Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. *CoRR*, abs/1409.2574, 2014.

[17] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[18] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[20] K. Hu and D. Wang. An unsupervised approach to cochannel speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(1):122–131, Jan 2013.

[21] P. Huang, Y. Huang, W. Wang, and L. Wang. Deep embedding network for clustering. In *2014 22nd International Conference on Pattern Recognition*, pages 1532–1537, Aug 2014.

[22] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey. Single-channel multi-speaker separation using deep clustering. *CoRR*, abs/1607.02173, 2016.

[23] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen. Multi-talker Speech Separation and Tracing with Permutation Invariant Training of Deep Recurrent Neural Networks. *ArXiv e-prints*, March 2017.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[25] Jonathan Le Roux, Felix J. Weninger, and John R. Hershey. Sparse nmf – half-baked or well done? Technical Report TR2015-023, Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA, March 2015.

[26] Roland Maas, Sree Hari Krishnan Parthasarathi, Brian King, Ruitong Huang, and Björn Hoffmeister. Anchored speech detection. *Interspeech 2016*, pages 2963–2967, 2016.

[27] Tobias Menne, Jahn Heymann, Anastasios Alexandridis, Kazuki Irie, Albert Zeyer, Markus Kitza, Pavel Golik, Ilia Kulikov, Lukas Drude, Ralf Schlüter, Hermann Ney, Reinhold Haeb-Umbach, and Athanasios Mouchtaris. The rwth/upb/forth system

combination for the 4th chime challenge evaluation. In *The 4th International Workshop on Speech Processing in Everyday Environments*, pages 39–44, San Francisco, CA, USA, September 2016.

[28] Christopher Olah. Understanding lstm networks. `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`, 2015.

[29] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[30] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and Andrew. Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 30–36, Dec 2015.

[31] M. N. Schmidt and R. K. Olsson. Single-channel speech separation using sparse non-negative matrix factorization, sep 2006.

[32] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1293–1299. AAAI Press, 2014.

[33] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[34] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, July 2006.

[35] Tuomas Virtanen. Speech recognition using factorial hidden markov models for separation in the feature space, 2006.

[36] H. von Helmholtz and A.J. Ellis. *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. Longmans, Green, 1885.

[37] DeLiang Wang and Guy J. Brown. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006.

[38] Yuxuan Wang, Arun Narayanan, and DeLiang Wang. On training targets for supervised speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 22(12):1849–1858, 2014.

[39] C. Weng, D. Yu, M. L. Seltzer, and J. Droppo. Deep neural networks for single-channel multi-talker speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(10):1670–1679, Oct 2015.

[40] Felix Weninger, Hakan Erdogan, Shinji Watanabe, Emmanuel Vincent, Jonathan Le Roux, John R Hershey, and Björn Schuller. Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 91–99. Springer, 2015.

[41] M. Wertheimer. *Laws of organization in perceptual forms.* Harcourt, Brace & Jovanovitch, London, 1938.

[42] Y. Xu, J. Du, L. R. Dai, and C. H. Lee. An experimental study on speech enhancement based on deep neural networks. *IEEE Signal Processing Letters*, 21(1):65–68, Jan 2014.

[43] D. Yu, X. Chang, and Y. Qian. Recognizing Multi-talker Speech with Permutation Invariant Training. *ArXiv e-prints*, March 2017.

[44] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. *CoRR*, abs/1607.00325, 2016.