Practical applicability of deep neural networks for overlapping speaker separation

Pieter Appeltans[1]*, Jeroen Zegers*[2]*, Hugo Van hamme*[2]

[1]Department of Computer Science - KU Leuven - Belgium
[2]Processing Speech and Images - ESAT - KU Leuven - Belgium
{pieter,jeroen,hugo}.{appeltans,zegers,vanhamme}@kuleuven.be

Abstract

This paper examines the applicability in realistic scenarios of two deep learning based solutions to the overlapping speaker separation problem. Firstly, we present experiments that show that these methods are applicable for a broad range of languages. Further experimentation indicates limited performance loss for untrained languages, when these have common features with the trained language(s). Secondly, it investigates how the methods deal with realistic background noise and proposes some modifications to better cope with these disturbances. The deep learning methods that will be examined are deep clustering and deep attractor networks.

Index Terms: Source Separation, Recurrent neural networks, Artificial neural networks

1. Introduction

The overlapping speaker separation problem consists of separating the utterances of multiple speakers from a mixture. Many cues, such as the identity, position, and lip movement of the speakers could be used to tackle this problem. This paper however will focus on methods that only use a mono recording of the mixture.

The current state-of-the-art methods to address the speaker separation problem are based on deep neural networks. These methods are able to obtain good text-independent separations with no or limited prior information [2]. This is a big improvement compared to previous methods like hidden Markov models [3, 4, 5], independent component analysis [6], computational auditory scene analysis [7, 8] and non-negative matrix factorisation [9, 10], which have limited separation performance or impose restrictions on the speakers and vocabulary. This improved performance comes at the cost of needing a lot of labelled training data (mixtures for which the desired separation is known) and demanding computations. The former can be tackled by artificially generating mixtures from two separate sources. The latter becomes feasible due to the increasing parallel computation power of graphical processing units.

This paper will focus on two such methods, namely deep clustering (DC) [2] and deep attractor networks (DAN) [11]. Both use (bidirectional) recurrent neural networks with long-short term memory (LSTM) cells to map each bin in the log magnitude spectrogram of the mixture to an embedding vector. This mapping is learned from training data and is such that embedding vectors associated with bins dominated by the same speaker are close. These vectors are then used to generate masks to filter out the individual speakers from the mixture. By using these intermediate embedding vectors instead of directly outputting the masks, the so called permutation problem [11] is avoided.

This paper is organised as follows. In the remainder of this introduction the specific details of the two examined method are further discussed. Section 2 presents experiments to asses their performance for six different languages. Subsequently, Section 3 will examine how well a model trained for one language generalises to another language and how this generalisation changes when multiple languages are used for training. Section 4 discusses the applicability of these methods in the presence of background noise and proposes some modifications to improve their performance. Finally, Section 5 gives some overall conclusions.

1.1. Deep clustering [2]

In DC, the network is trained by minimising the following loss function:

$$\sum_{n=1}^{N} \frac{1}{K_n^2} ||\mathbf{V}_n \mathbf{V}_n^T - \mathbf{Y}_n \mathbf{Y}_n^T||_F^2 \tag{1}$$

with N the number of mixtures in the training set, K_n the number of time-frequency bins in the spectrogram of mixture n, $\mathbf{V}_n$ a $K_n \times D$ (with D the size of the embedding vectors) dimensional matrix with the embedding vectors, outputted by the network, each normalised to (euclidean) norm 1, $\mathbf{Y}_n$ a $K_n \times C_n$ (with C_n the number of speaker in mixture n) dimensional matrix with $y_{(t,f),c} = 1$ if speaker c dominates the time-frequency bin and $y_{(t,f),c} = 0$ else. This cost function can be understood as follows.

$\mathbf{Y}_n \mathbf{Y}_n^T$ and $\mathbf{V}_n \mathbf{V}_n^T$ are $K_n \times K_n$ dimensional matrices. $\{\mathbf{Y}_n \mathbf{Y}_n^T\}[(t,f),(t',f')]$ is equal to one when time-frequency bins (t,f) and (t',f') are dominated by the same speaker and zero in the other case and the $((t,f),(t',f'))^{\text{th}}$ element of $\mathbf{V}_n \mathbf{V}_n^T$ is the euclidean inner product of $\mathbf{v}^{(t,f)}$ and $\mathbf{v}^{(t',f')}$. Minimising the cost function will thus tend to map embedding vectors associated with time-frequency bins dominated by the same speaker near each other ($\mathbf{v}^{(t,f)^T} \mathbf{v}^{(t',f')} \approx 1$) and vectors associated with different speakers will tend to be orthogonal ($\mathbf{v}^{(t,f)^T} \mathbf{v}^{(t',f')} \approx 0$).

After training, the network is used to separate new unseen mixtures. This is done by applying its log magnitude spectrogram to the network and clustering the resulting embedding vectors with K-means. Each cluster represents one speaker and is used to create a binary mask to reconstruct the original utterance of the speaker.

1.2. Deep attractor networks [11]

In DANs the network is trained by minimising:

$$\sum_{n=1}^{N} \frac{1}{K_n * C_n} \sum_{c=1}^{C_n} ||\mathbf{S}_{n,c}^{mag} - \mathbf{X}_n^{mag} \odot \mathbf{M}_{n,c}||_F^2 \tag{2}$$

with $\mathbf{X}_n^{mag}$ the magnitude spectrogram of mixture n, $\mathbf{S}_{n,c}^{mag}$ the original magnitude spectrogram of speaker c in mixture n, $\odot$ the element wise product, and $\mathbf{M}_{n,c}$ the estimated mask for speaker c that is obtained as follows from the output of the network:

$$M_{n,c}[t,f] = \frac{1}{1+\exp(-\mathbf{a}_c \cdot \mathbf{v}(t,f))} \tag{3}$$

with $\mathbf{a}_c$ the attraction point of speaker c, which is calculated as the mean of the embedding vectors associated with the speaker:

$$\mathbf{a}_c = \frac{\sum_{(t,f)} \mathbf{v}(t,f) y_{(t,f),c}}{\sum_{(t,f)} y_{(t,f),c}} \tag{4}$$

By minimising the above mentioned loss function, the network learns to form an attraction point in embedding space for each speaker, that attracts embedding vectors associated with time-frequency bins of this source.

To separate an unseen mixture, its log magnitude spectrogram is fed to the network and the obtained embedding vectors are used to create a ratio mask for each speaker using Eq. (3). Because the partitioning of the bins ($y_{(t,f),c}$) is not known (this is exactly what we are looking for), Eq. (4) cannot be used to calculate the attraction points. These are therefore approximated by the cluster centres found by K-means clustering of the embedding vectors.

2. Different languages

In [2] and [11] the separation performance of the above mentioned methods is only examined for mixtures of English speakers. This section presents experiments with six other languages, including a tonal language. It is structured as follows: first the experiment design is explained; subsequently the separation scores are presented and discussed.

2.1. Experiment set-up

The mixtures are generated using the global phone corpus [15] by overlaying utterances of two different speakers. To compare with the results in [2] and [11], we used a similar set-up: the signals were subsambled to 8kHz (to limit memory requirements and computation time); we calculate the (log magnitude) spectrogram using the short time Fourier transformation with a cosine window of 32 milliseconds and an overlap of 8 milliseconds; the neural network consisted of two layers of 600 bidirectional LSTM cells, followed by a fully connected layer of neurons with linear activation function; a 20 dimensional embedding space was used. For each language the training set consisted of 20 000 training mixtures, which each contained 2 speakers randomly sampled from a pool of 70 speakers, the development set 3 000 mixtures sampled from 10 speakers and the test set 3 000 mixtures sampled from 20 speakers. The speakers in the different data sets are non-overlapping and in each set there were as many male as female speakers.

The quality of the separations is quantified by the signal to distortion ratio (SDR) which measures the retrieved source energy relative to the energy of interfering sources and artifacts.

2.2. Results

Table 1 gives the average SDR for DC and DAN for mixtures of two speakers in respectively Arabic, French, Mandarin, Portuguese, Spanish, and Swedish. These scores are in line to with the results in [2] and [11] for English. In our experiments deep attractor networks outperform deep clustering for every language and therefore seems the better choice. Both methods obtain their best score for Mandarin, which is the only tonal language in our test set. This might indicate that tonality is a useful feature for speaker separation but more research with other tonal languages is needed to support this thesis.

Table 1: *The average SDR (in dB) when trained and tested on the same language.*

language	deep clustering	deep attractor networks
Arabic	7.50	7.97
French	7.46	8.20
Mandarin	8.54	8.86
Portuguese	7.24	8.27
Spanish	6.72	7.76
Swedish	6.93	7.83

3. Generalisation to an unseen language

This section will examine how well a network can separate mixtures of an untrained language. The reasons for these experiments are threefold. Firstly, it may not be reasonable to assume the speaker's language is know, e.g. when deploying a conferencing service over the internet or when built into a mobile phone. Secondly, these results give an indication of the robustness against different accents and dialects of a language. Lastly, they might give some information on what cues, such as phonetic, phonotactic, lexical or grammatical, the methods exploit to separate speakers. In Section 3.1 the set-up of the experiments is described. Next, experiments with networks trained with one language are presented in Section 3.2. Section 3.3 examines whether the performance for trained and untrained languages improves when more than one training language is used.

3.1. Experimental set-up

In Section 3.2 we reuse the networks from Section 2 trained with respectively French and Swedish speakers. The French network is tested for mixtures with respectively Portuguese and Mandarin speakers. The Swedish network is tested with mixture of respectively Arabic and Spanish speakers. In Section 3.3 new networks are trained with {French, Turkish}, {French, Turkish, Japanese}, {Swedish, Turkish} and {Swedish, Turkish, Japanese} datasets. For each network the training set consisted of 20 000 two-speaker mixtures sampled from a pool of 70 speakers, equally balanced between languages and genders. The development set consisted of 3 000 mixtures sampled from 10 speakers. Mixtures consisted only of speakers of the same language. The test sets are the same as in Section 2.

3.2. Network trained with one language

Table 2 gives the average separation performance of the methods for untrained languages. Also the difference with the score of the network trained with the considered language (Table 1) is given.

We noticed that for all languages there is a significant decrease in separation quality compared to the network trained with the test language itself. Portuguese, Arabic and Spanish have a decrease of about 1dB, and the obtained separation are still of good quality. For Mandarin on the other hand the decrease is more significant, around 4dB. This seems to indicate

that the performance for untrained languages depends on the relation of the training and test language (closer related is better).

The fact that the methods do not break completely implies that they do not create grammatical or lexical models, but at most phonotactic or phonetic models. They do seem to do more than tracking formants or pitch, which would make them almost language independent.

Table 2: *The average SDR in dB for DC and DAN for an unseen test language and the difference with the SDR for matched language training.*

language	deep clustering	deep attractor networks
	French network	
Mandarin	4.59 (-3.95)	4.86 (-4.00)
Portuguese	6.33 (-1.13)	7.22 (-0.98)
	Swedish network	
Arabic	5.98 (-1.52)	7.01 (-0.96)
Spanish	6.01 (-0.71)	7.20 (-0.55)

3.3. Network trained with multiple languages

Table 3 gives the separation quality for the networks trained with multiple languages. The average SDR is reported for both trained (t) languages and untrained (u) languages and the difference with scores of the networks trained with the language itself (Table 1). From the results we observe that for trained languages it is in most cases disadvantageous to replace a part of the training data with mixtures in other languages. For untrained languages on the other hand, it is in some cases advantageous to include multiple training languages instead of one. Only for Portuguese there is a consistent decrease in performance compared to the results of the previous subsection.

Table 3: *The average SDR in dB for DC and DAN trained with multiple languages for trained and untrained languages and the difference with the SDR for matched language training.*

language	deep clustering	deep attractor networks
	{French, Turkish} network	
French (t)	6.92 (-0.55)	7.75 (-0.45)
Mandarin (u)	4.89 (-3.65)	5.32 (-3.54)
Portuguese (u)	6.31 (-1.15)	7.24 (-0.96)
	{French, Turkish, Japanese} network	
French (t)	6.35 (-1.11)	7.27 (-0.93)
Mandarin (u)	4.57 (-3.97)	5.34 (-3.52)
Portuguese (u)	5.94 (-1.53)	6.77 (-1.44)
	{Swedish, Turkish} network	
Swedish (t)	7.03 (0.10)	6.97 (-0.87)
Arabic (u)	6.45 (-1.02)	6.71 (-1.26)
Spanish (u)	6.39 (-0.33)	6.99 (-0.77)
	{Swedish, Turkish, Japanese} network	
Swedish (t)	6.75 (-0.18)	7.58 (-0.25)
Arabic (u)	6.49 (-1.02)	7.31 (-0.66)
Spanish (u)	6.16 (-0.56)	7.34 (-0.42)

4. Coping with background noise

In this section we examine the usability of deep clustering and deep attractor networks in the presence of realistic background noise and propose some modifications. This section is organized as follows. First, the modifications to the original methods are presented. Subsequently, the set-up of the experiments is discussed. To conclude, the performance of the original and modified methods are compared.

4.1. Proposed modifications

4.1.1. Modified network architecture

Figure 1 shows the modified network architecture. Besides an embedding vector, it now has a (scalar) mask output for each time-frequency bin. This scalar is an estimated ratio mask to suppress the noise in that bin. Because noise and speech signals have different roles and structures, there is no need for permutation invariance and the network can therefore directly output a noise filter mask.

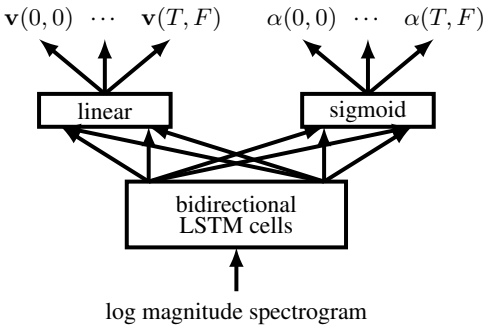

Figure 1: *Modified network architecture to better cope with background noise. It takes as input the log magnitude spectrogram of the mixture and has as output an embedding vector and a noise mask for each bin in the spectrogram.*

4.1.2. Deep clustering

Loss function Eq. (1) is modified to:

$$\sum_{n=1}^{N} \frac{1}{\tilde{K}_n^2} ||\tilde{\mathbf{V}}_n \tilde{\mathbf{V}}_n^T - \tilde{\mathbf{Y}}_n \tilde{\mathbf{Y}}_n^T||_F^2 + \gamma \frac{1}{K_n} ||\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_n^{\text{ideal}}||_F^2 \quad (5)$$

with K_n as defined previously, $\tilde{K}_n$ the number of bins not dominated by noise, $\tilde{\mathbf{Y}}_n$ and $\tilde{\mathbf{V}}_n$ as defined previously but the rows associated with bins dominated by noise are set to zero, $\boldsymbol{\alpha}$ the ratio mask estimated by the network, and $\boldsymbol{\alpha}^{ideal}$ the optimal ratio mask to filter the noise. The first term is similar to Eq. (1). The second term trains the network to generate ratio masks to filter out the noise by penalizing the distance between the estimated and the optimal mask. The hyper-parameter γ weighs the importance of separating the speakers and filtering out noise. In our experiments in Section 4.3 γ is arbitrarily set to one.

Also the procedure to separate unseen mixtures is modified. Firstly, before separating the speakers the estimated noise mask is used to suppress the noise. Secondly, only the embedding vectors for which are the associated α is greater than 0.75 are used in the clustering algorithm. The remaining bins are assigned to one of the speakers based on the distance between their embedding vector and the cluster centres of the speakers.

Based on these clusters, a binary mask to separate the speakers is generated.

4.1.3. Deep attractor networks

For deep attractor networks the loss function Eq. (2) is modified to:

$$\sum_{n=1}^{N} \frac{1}{K_n * C_n} \sum_{c=1}^{C_n} ||\mathbf{S}_{c,n}^{mag} - (\mathbf{X}_n^{mag} \odot \mathbf{M}_n^{noise}) \odot \mathbf{M}_{n,c}||_F^2 \quad (6)$$

with $\mathbf{M}^{noise}[t,f] = \alpha(t,f)$ the estimated noise mask. Also Eq. (4) is modified:

$$\mathbf{a}_c = \frac{\sum_{(t,f)} \mathbf{v}(t,f)\tilde{y}_{(t,f),c}}{\sum_{(t,f)} \tilde{y}_{(t,f),c}} \quad (7)$$

with $\tilde{y}_{(t,f),c}$ equal to one when speaker c dominates the bin, the bin has enough energy and $\alpha(t,f)$ bigger than 0.75 and zero in all other cases. Although this hard cut-off introduces discontinuities and local optima in the cost function, an alternative (smoother) penalty for noisy bins did not lead to improved performance.

To separate new mixtures, a similar strategy as in Section 1.2 is applied, but with two slight modifications. Firstly, prior to separating the speakers, the noise was filtered using the estimated noise mask. Secondly, to estimate the attraction points the K-means clustering is only applied to embedding vectors of time-frequency bins with enough energy and α above 0.75.

4.2. Experiment set-up

The utterances for the experiments in 4.3 were sampled from the 'Wall Street Journal Database'[18]. The noise signals were chosen from the 'third CHiME speech separation and recognition challenge' data set [19], which contains recordings of realistic environment noise. As in the previous sections all signals were first downsampled to 8kHz. Six different two-speaker mixture sets were used:

- A noise free training (20 000 mixtures) and development set (5 000 mixtures). The signals are normalised such that the individual speakers have the same power.
- A noisy training (100 000 mixtures) and development set (5 000 mixtures). The training set reuses each mixture of the noise free training set five times, each time with different noise. The new development set is similar to the noise free variant, only with noise added. The signals of the speakers and the noise are normalised such that they have the same power.
- A noisy test set of 3 000 mixtures with different speakers and utterances than in the training and development sets. The noise comes from different parts of the same recordings as the training and development sets (for the training and development sets noise is sampled from the first 10 minutes of the recording, for the test set from the leftover part). The signals of the speakers and the noise are normalised such that they have equal power.
- A second noisy test set of 3 000 mixtures. Similar to the previous test set but now the signals are normalised such that both speakers have equal power and the noise is 3dB weaker than each speaker.

4.3. Results

Table 4 compares the performance of the following five methods for the two noisy test sets described in 4.2:

- deep clustering trained without noise (DC no noise);
- deep attractor networks trained without noise (DAN no noise);
- deep clustering with noise (DC with noise). During training the noise was considered as third speaker and the network was trained to form three clusters: two associated with speakers and one associated with the noise. During testing three reconstructions were created but only the two that most resembled a speaker were used for scoring;
- modified deep clustering described in Section 4.1.2 (modified DC);
- modified deep attractor networks described in Section 4.1.3 (modified DAN).

For all methods a 20 dimensional embedding space was used. The recurrent part of the networks trained without noise consisted of two layers with 800 bidirectional LSTM cells each. For the networks trained with noise this consisted of four layers with each 800 bidirectional LSTM cells.

The models trained without noise break down on noisy data. Including noise during training as a third speaker already leads to improved performance. The best SDRs are obtained with the modified methods of Section 4.1. The SDR improvement w.r.t. "DC with noise" comes at a cost of a few dB in SNR, which seems less important since noise is not the main source of distortion.

Table 4: *The average SDR and SNR in dB for the test sets with respectively the two speakers and the noise equally loud (0 dB) and the noise 3dB quieter than the speakers (3 dB)*

	0 dB		3 dB	
Method	SDR	SNR	SDR	SNR
DC no noise	-1.75	5.38	1.99	11.5
DC with noise	4.33	16.1	6.17	19.2
modified DC	5.11	12.8	7.43	17.2
DAN no noise	-0.37	5.83	2.67	10.8
modified DAN	5.27	13.5	7.33	17.4

5. Conclusion

Deep clustering and deep attractor networks are applicable to source separation in a wide variety of languages, including tonal languages. Training models with (a combination of) related languages yields only minor performance degradation compared to training on the target language. This observation supports the results in [17], which showed that recurrent networks trained for speech separation mainly exploit information with the time span of a phone and long span information is limited to speaker identity while lexical or grammatical patterns are ignored. Furthermore,we extended deep clustering and deep attractor networks with an estimated spectral mask to cope with noisy mixtures and showed significant improvement over the baselines. A limitation of the current experiments is that they only examine how well the methods perform for noise for which we have training data. Future work will consider "untrained" noise types.

6. References

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953. [Online]. Available: https://doi.org/10.1121/1.1907229

[2] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 31–35.

[3] J. R. Hershey, S. J. Rennie, P. A. Olsen, and T. T. Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Computer Speech and Language*, vol. 24, no. 1, pp. 45 – 66, 2010, speech Separation and Recognition Challenge. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230808000557

[4] R. J. Weiss and D. P. W. Ellis, "Monaural speech separation using source-adapted models," in *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct 2007, pp. 114–117.

[5] G. J. Mysore, P. Smaragdis, and B. Raj, "Non-negative hidden markov modeling of audio with application to source separation." in *LVA/ICA*. Springer, 2010, pp. 140–148.

[6] Z. Koldovsky and P. Tichavsky, "Time-domain blind separation of audio sources on the basis of a complete ica decomposition of an observation space," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 406–416, Feb 2011.

[7] K. Hu and D. Wang, "An unsupervised approach to cochannel speech separation," *IEEE Transactions on audio, speech, and language processing*, vol. 21, no. 1, pp. 122–131, 2013.

[8] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE press, 2006.

[9] J. Le Roux, F. J. Weninger, and J. R. Hershey, "Sparse nmf–half-baked or well done?" *Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA, Tech. Rep., no. TR2015-023*, 2015.

[10] M. N. Schmidt, "Speech separation using non-negative features and sparse non-negative matrix factorization," *Elsevier*, 2007.

[11] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 246–250.

[12] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, Mar 1996.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] T. Schultz, "Globalphone: a multilingual speech and text database developed at Karlsruhe University," in *Seventh International Conference on Spoken Language Processing*, 2002.

[16] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[17] J. Zegers *et al.*, "Memory time span in lstms for multi-speaker source separation," *arXiv preprint arXiv:1808.08097*, 2018.

[18] J. S. Garofolo, D. Graff, D. Paul, and D. Pallett, "Wall street journal dataset," https://catalog.ldc.upenn.edu/ldc93s6a, 1993. [Online]. Available: https://catalog.ldc.upenn.edu/ldc93s6a

[19] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 504–511.