Multi-channel Speech Enhancement Based on Deep Stacking Network

Hui Zhang, Xueliang Zhang, Guanglai Gao

Department of Computer Science, Inner Mongolia University, Hohhot, China, 010021

alzhu.san@163.com, {cszxl,csggl}@imu.edu.cn

Abstract

Beamforming enhances sound components coming from a direction specified by a steering vector. Some beamforming methods use the time-frequency masks for the steering vector estimation. Better masks lead to better beamforming results. Meanwhile, the beamforming results carry cross-channel information which make the mask estimation easier. Therefore, the beamforming and the mask estimation can boost each other, and can be treated as a "chicken-and-egg" problem. In this work, we embed the beamforming and the mask estimation into a deep stacking network architecture as the speech separation front-end. Together with the state-of-the-art speech recognition back-end, the proposed method obtains 11.00% and 6.00% WER for the real test data in the 4th CHiME Challenge 2 channels and 6 channels tracks.

1. Background

This paper introduces the speech separation and recognition system designed for the 4th CHiME Challenge [1] 2 channels and 6 channels tracks.

From the review of the last CHiME Challenge, we find that the success is mostly relative to the time-varying minimum variance distortionless response (MVDR) beamforming [2].

A beamformer enhances the sound components coming from a direction which specified by a steering vector. The accurate steering vector estimation is the key to effective beamforming. Recently, a beamforming method was proposed that uses the time-frequency masks to estimate the steering vector [3], where the masks represent the probabilities of background noise dominating the corresponding time-frequency points. In this method, the accurate mask estimation is the key to effective steering vector estimation. Better mask estimations lead to better steering vector estimations and better beamforming results. Mask estimation is helpful for the beamforming. Beamforming is also helpful for the mask estimation. The beamforming results are built from multi-channel microphone array, so that they contain cross-channel information which is useful for the mask estimation of a certain single channel.

Because the beamforming and the mask estimation can boost each other, they can be treated as a "chicken-and-egg" problem. In [4], the authors proposed using the deep stacking network (DSN) architecture to solve the "chicken-and-egg" problem. In DSN, each basic module is used to process a "chicken-and-egg" step. DSN stacks these basic processing modules to build forward deep architectures. With the increasing of the number of stacked modules, the system's performance is improving. We consider the mask estimation and beamforming as a "chicken-and-egg" step, process them with a basic module, and embed them into a DSN to form the speech separation front-end. Specifically, we first obtain the estimated masks from a basic module. Then these estimated masks are used to perform the beamforming. Next these beamforming results are used to obtain new estimated masks by another basic module. Then these new estimated masks are used for beamforming, estimating new masks, and so on.

2. Contributions

2.1. Mask Estimation

Before getting any beamforming results, we need a initial mask estimation. We use deep neural network (DNN) as a basic module to estimate the ideal ratio mask (IRM):

$$IRM = \sqrt{\frac{|STFT^{\{speech\}}|^2}{|STFT^{\{speech\}}|^2 + |STFT^{\{noise\}}|^2}} \tag{1}$$

where $|STFT^{\{speech\}}|$ and $|STFT^{\{noise\}}|$ is the short time Fourier transform (STFT) features of the premixed speech and noise. We obtain the STFT features by applying 320-point Fourier transform on each hamming window frame which length of 20-ms and shift with 10-ms, and using the absolute value of the first 161-D Fourier coefficients.

The DNN contains three 1024-node ReLU hidden layers, and the output transform is sigmoid. The inputs of the DNN is the STFT features of the mixtures. Before feeding into the DNN, the STFT features are compressed by a cubic root operation. The input features also contain a context window of previous 2 and subsequent 2 frames. Therefore, the input is a $161 \times 5 = 805$ dimensional vector.

The DNN is trained with all of the simulated training data with early stop controlled by a 10% left out develop set.

2.2. Beamforming

After obtaining the estimated mask, we get the beamforming results using the the time-frequency mask based MVDR beamforming method [3], where the masks represent the probabilities of background noise dominating the corresponding time-frequency points. We obtain this mask base on the estimated IRM:

$$mask = 1 - max\{IRM_1, \ldots, IRM_N\} \tag{2}$$

where IRM_i is estimated IRM in channel i. N is number of channels. For 2 channels track, $N = 2$, and for 6 channels track, $N \leq 5$, where we drop the backward channel 2, and remove failed channels with the scripts offered by the official baseline.

2.3. Mask Estimation with Beamforming

After getting the beamforming results, we can use them to improve the mask estimation. In this step we use another DNN basic module to estimate the IRM. The DNN's structure is same as the one in Sect. 2.1 except the inputs. The inputs of the DNN contain three parts: the estimated IRM from the last DNN module, the STFT features of the mixtures, and the STFT features of the corresponding beamforming results. These beamforming results may contribute to the improvement of the mask estimation. Before feeding into the DNN, all of the STFT features are compressed by a cubic root operation. All of the STFT features are extended with its previous and subsequent 1 frames as context. Therefore, the input is a $161 + 161 \times 3 + 161 \times 3 = 1127$ dimensional vector.

We use the same DNN for the 2 channels and 6 channels tracks. The beamforming results used for training are generated as follows. We first divide the simulated training utterances randomly into two sets whose size are almost the same. One part for the 6 channels track, and another for the 2 channels track. In the one for 2 channels track, we further pick 2 channels randomly for each utterance, and remove others. Then the beamforming results are generated from these two sets.

The DNN is trained with all of the simulated training data with early stop controlled by a 10% left out develop set.

2.4. Combining Mask Estimation and Beamforming

We perform the mask estimation and beamforming alternantly and iteratively by embedding them into a DSN, where we stack basic modules one by one, and as illustrated in Fig. 1. We first obtain the initial estimated IRM by the module described in Sec. 2.1. Then we get the beamforming results as described in Sec. 2.2. Next the beamforming results are used for updating the estimated IRM by the module described in Sec. 2.3. Then these updated estimated masks are used for beamforming, estimating new masks, and so on.

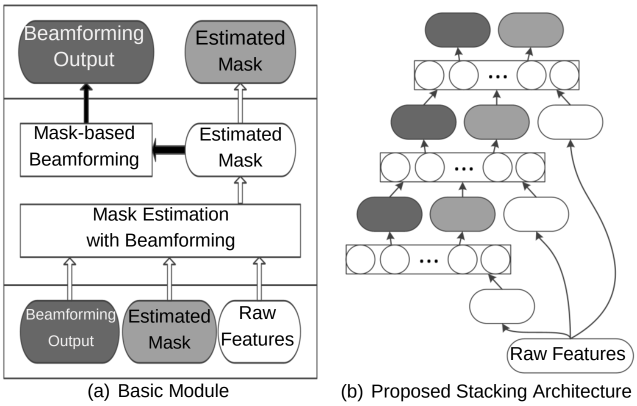

Figure 1: Schematic diagram of the proposed system.

2.5. ASR Back-end

We can further improve the performance of ASR systems by increasing the amount of training data, so that we use scripts offered by the official baseline to train a new ASR back-end with all of the 6 channels training data.

3. Experimental evaluation

In the official baseline, four types of ASR back-ends are involved, which are GMM-based (denoted as "GMM"), DNN-based (denoted as "DNN"), DNN-based with a larger language model (denoted as "5kng") and DNN-based with RNN-based language model (denoted as "RNNML"). We report the results using all of these four ASR back-ends, and compare the proposed system with the official baseline front-end "BeamformIt" system. The proposed front-end is named as "model-N", where N indicates the number of the stacked modules. The average WER of all systems with the baseline ASR back-end and with the new ASR back-end in Tab. 1 and Tab. 3. And the detail WER of the best system are given for each noisy environment in Tab. 2 and Tab. 4.

Compared the Tab. 3 with Tab. 1, we can see that the new ASR back-end can generate better ASR results than the baseline ASR back-end. From Tab. 1, compared with the 6 channel the model-1 system with RNNML ASR back-end and the one reported in [3], the WER in the real test data is 7.44 and 8.86, respectively. It indicates that the DNN is powerful than the complex Gaussian mixture model (CGMM) used in [3] for mask estimation. And compared among the proposed system with different numbers of the stacked modules, we find the performance of the system is improving with the increasing of the number of stacked modules. In addition, the single channel signal and the corresponding beamforming result are often mismatch in the time axis. The experimental results show that the mask estimation can benefit from the beamforming results although the inputs do not match strictly.

4. Conclusion

Because mask estimation and beamforming can boost each others, we treat them as a "chicken-and-egg" problem, and iterate them alternatingly in a DSN. The experimental results show that the proposed method can improve the ASR performance in noisy environment, and the performance of the system is improving with the increasing of the number of stacked modules. The proposed method obtains a comparable performance without any advanced language model or speaker adaptation which are the primary weapons of other participators.

5. Acknowledgments

This research was supported in part by the China national nature science foundation (No. 61365006, No. 61263037).

6. References

[1] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language*, to appear.

[2] 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.

[3] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 5210–5214.

[4] X. Zhang, H. Zhang, S. Nie, G. Gao, and W. Liu, "A pairwise algorithm using the deep stacking network for speech separation and pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 6, pp. 1066–1078, June 2016.

Table 1: Average WER (%) for the tested systems with baseline ASR back-end.

Track	System		Dev		Test	
			real	simu	real	simu
2ch	GMM	BeamformIt	16.23	19.14	29.05	27.56
		model-1	14.53	16.25	24.49	19.50
		model-2	14.47	15.97	23.84	19.10
		model-3	14.61	15.83	24.47	19.73
	DNN	BeamformIt	10.90	12.36	20.44	19.03
		model-1	9.29	10.03	17.39	12.86
		model-2	9.08	9.89	16.58	12.91
		model-3	9.04	9.91	16.80	12.72
	5kng	BeamformIt	9.63	10.72	18.08	16.88
		model-1	7.77	8.65	15.07	10.68
		model-2	7.71	8.53	14.27	10.62
		model-3	7.74	8.63	14.38	10.71
	RNNML	BeamformIt	8.23	9.49	16.58	15.34
		model-1	6.74	7.66	13.54	9.46
		model-2	6.57	7.57	12.92	9.55
		model-3	6.57	7.57	12.75	9.37
6ch	GMM	BeamformIt	13.04	14.30	21.83	21.29
		model-1	9.64	10.10	15.08	11.81
		model-2	9.55	10.12	14.53	11.99
		model-3	9.48	10.17	14.48	11.87
	DNN	BeamformIt	8.14	9.07	15.04	14.19
		model-1	6.25	5.96	10.22	7.62
		model-2	6.10	6.08	10.07	7.87
		model-3	6.01	6.20	10.10	8.02
	5kng	BeamformIt	6.85	7.74	13.18	12.33
		model-1	4.91	5.09	8.69	6.17
		model-2	4.82	5.07	8.52	6.29
		model-3	4.91	5.03	8.47	6.60
	RNNML	BeamformIt	5.75	6.77	11.47	10.91
		model-1	4.12	4.20	7.44	5.44
		model-2	3.99	4.41	7.17	5.32
		model-3	4.03	4.42	7.15	5.58

Table 2: WER (%) per environment for the best system with baseline ASR back-end.

Track	Envir.	Dev		Test	
		real	simu	real	simu
2ch	BUS	8.17	6.06	20.15	7.08
	CAF	6.30	10.16	12.07	10.38
	PED	4.60	6.39	9.38	9.54
	STR	7.20	7.67	9.39	10.46
6ch	BUS	5.21	3.89	11.57	4.54
	CAF	3.55	5.06	5.42	5.36
	PED	3.38	3.91	5.64	5.32
	STR	4.00	4.84	5.96	7.10

Table 3: Average WER (%) for the tested systems with new ASR back-end.

Track	System		Dev		Test	
			real	simu	real	simu
2ch	GMM	BeamformIt	15.21	16.86	26.23	25.80
		model-1	13.17	14.76	22.06	18.14
		model-2	12.92	14.46	21.43	17.78
		model-3	12.94	14.45	21.50	17.78
	DNN	BeamformIt	9.52	10.58	17.59	16.94
		model-1	7.99	8.37	14.79	11.02
		model-2	7.79	8.36	14.32	10.84
		model-3	7.83	8.26	14.51	11.09
	5kng	BeamformIt	7.97	8.95	15.31	14.57
		model-1	6.65	6.99	12.86	9.17
		model-2	6.47	7.09	12.37	8.95
		model-3	6.37	7.13	12.36	8.89
	RNNML	BeamformIt	7.01	8.02	13.70	13.28
		model-1	5.58	6.25	11.47	7.99
		model-2	5.48	6.26	11.02	7.80
		model-3	5.56	6.32	11.00	7.80
6ch	GMM	BeamformIt	12.25	12.97	19.99	19.53
		model-1	9.13	9.42	14.13	10.91
		model-2	9.01	9.51	13.47	11.33
		model-3	8.97	9.52	13.62	11.29
	DNN	BeamformIt	7.30	8.27	13.08	12.79
		model-1	5.53	5.30	8.90	6.76
		model-2	5.45	5.21	8.65	7.17
		model-3	5.44	5.27	8.66	7.09
	5kng	BeamformIt	6.04	6.71	11.23	10.95
		model-1	4.44	4.17	7.38	5.24
		model-2	4.25	4.28	7.08	5.39
		model-3	4.28	4.33	6.92	5.59
	RNNML	BeamformIt	5.07	6.08	9.88	9.47
		model-1	3.74	3.56	6.23	4.40
		model-2	3.62	3.65	6.05	4.58
		model-3	3.62	3.66	6.00	4.83

Table 4: WER (%) per environment for the best system with new ASR back-end..

Track	Envir.	Dev		Test	
		real	simu	real	simu
2ch	BUS	7.11	5.31	17.22	5.85
	CAF	5.34	8.48	10.14	9.19
	PED	4.07	5.13	8.05	7.92
	STR	5.71	6.37	8.61	8.24
6ch	BUS	4.59	3.32	9.46	3.88
	CAF	3.08	4.23	4.15	4.87
	PED	3.11	3.17	4.93	4.74
	STR	3.70	3.92	5.47	5.81