An Adaptation Method in Noise Mismatch Conditions for DNN-based Speech Enhancement

Xu* Si-Ying, Niu Tong, Qu Dan, Long Xing-Yan

National Digital Switching System Engineering & Technological R&D Center P.R., China
[e-mail: xusiying_2015@163.com]
[e-mail: jerry_newton@sina.com]
[e-mail: qudanqudan@sina.com]
[e-mail: lxy_120999@qq.com]
*Corresponding author: Siying Xu

*Received December 26, 2017; revised April 9, 2018; accepted May 3, 2018;
published October 31, 2018*

Abstract

The deep learning based speech enhancement has shown considerable success. However, it still suffers performance degradation under mismatch conditions. In this paper, an adaptation method is proposed to improve the performance under noise mismatch conditions. Firstly, we advise a noise aware training by supplying identity vectors (i-vectors) as parallel input features to adapt deep neural network (DNN) acoustic models with the target noise. Secondly, given a small amount of adaptation data, the noise-dependent DNN is obtained by using L_2 regularization from a noise-independent DNN, and forcing the estimated masks to be close to the unadapted condition. Finally, experiments were carried out on different noise and SNR conditions, and the proposed method has achieved significantly 0.1%-9.6% benefits of STOI, and provided consistent improvement in PESQ and segSNR against the baseline systems.

Keywords: Noise-aware Training; identity-vector, L_2 regularization, speech enhancement, DNN, condition mismatch

http://doi.org/10.3837/tiis.2018.10.017 **ISSN : 1976-7277**

1. Introduction

Speech enhancement is an essential part of speech signal processing in noisy environment, aiming at separating useful clean speech from noisy speech and improving the intelligibility and quality of contaminated speech [1]. It is broadly applied in many domains, such as speech interaction, teleconferencing, automatic speech recognition (ASR), speaker identification systems etc. Classical speech enhancement methods including spectral subtraction [2], Wiener filtering [3], statistical model-based methods [4], non-negative matrix factorization algorithms [5] and recently proposed deep neural network(DNN)-based enhancement [6-10] usually degrade rapidly in mismatch conditions. DNN is a multiple layer's perceptron, which is composed of an input layer, an output layer and three or more hidden layers. So far, speech enhancement in mismatch condition is still a very challenging problem.

In the situation of environment mismatch, noise adaptation can help to improve the modeling accuracies under the unseen type of noise by using a small amount of adaptation data.

There are mainly two kinds of noise adaptation algorithms. One is using Gaussian mixture models (GMMs), the other is based on DNN framework. Traditional GMM based noise adaptation methods include parameter adaptation method like vector Taylor series (VTS) [11-14] adaptation and feature normalization such as feature-space maximum likelihood linear regression (fMLLR) [15]. In VTS adaptation, an estimated noise model is used to adapt the Gaussian parameters of the recognizer based on a physical model that defines how noise corrupts clean speech. The nonlinear relationship is often approximated with the first-order VTS in GMMs. FMLLR applies an affine transform to the feature vector so that the transformed feature matches the model better. For GMM-HMMs, fMLLR transformations are estimated to maximize the likelihood of the adaptation data given to the model. [15] proposed feature-space discriminative linear regression (fDLR) in DNN framework, where cross-entropy (CE), a discriminative function, is used as optimization criterion. The fDLR in DNN is just like fMLLR in GMM acoustic model, and optimization criterion is the only difference.

In [16-17], a Noise-aware Training (NaT) is proposed, in which the DNN is being given noise estimation in order to automatically learn the mapping from the noisy speech and noise to the ideal mask labels, implicitly through a clean speech estimation. Noise information can be derived in many different ways. It can be jointly learned with the rest of the model parameters, or it can be estimated completely independent of the DNN training. For example, it may be learned from a separate DNN from which either the output node or the last hidden layer can be used to represent the noise information. In recent years, some speaker adaptation methods have been successfully utilized for noise adaptation. In [18-19], speaker-code based method is proposed to perform speaker adaptation in model space without using any adaptation neural networks. In [20], the probabilistic principle component analysis (PPCA) is proposed to provide not only the speaker space models but also a priori distribution, which can be directly applied to the maximum a posteriori (MAP) estimation scheme of the model parameters. Among the speaker adaptation methods, in [21], i-vector [22-23] method is used. I-vector is a popular technique for speaker verification and recognition. It encapsulates the most important information about a speaker's, noise's or device's identity in a low-dimensional fixed-length representation and thus is an attractive tool for speaker adaptation techniques for ASR. Since a single low-dimensional i-vector is estimated from all the utterances of the same speaker, the same type of noise or the same device, i-vector can be reliably estimated from less data than other approaches. I-vector has become the common speaker adaptation feature, rarely used in speech enhancement as a representation of environment information.

Some ideas in DNN speech recognition can be used in speech enhancement for reference. Although the adapted DNN-based acoustic model shows more performance gain than the traditional acoustic models, they are mainly used for ASR performance promotion. The attractiveness of NaT and these previous works (notably [9]) motivated us to look at their applicability to noise adaptation of DNNs for speech enhancement.

In this paper, a noise adaptation method based on DNN for speech enhancement is proposed to ameliorate the mismatching problem under multi-type noise condition. We combine noise-aware training (NaT) and L_2 regularization. We use NaT to obtain noise information as auxiliary features, and the DNN can tune model parameters with it. And L_2 regularization is added to original adaptation criterion, so that we can use a small amount of adaptation data, improving the performance of mismatching system and ameliorating the noise adaptation.

The rest of the paper is organized as follows. Section 2 describes the supervised speech enhancement system in DNN framework. In Section 3, noise-aware training (NaT) with identity-vector (i-vector) and L_2 regularization for speech enhancement are proposed. In Section 4, we show experimental settings and report some results. Section 5 concludes this paper.

2. Supervised Speech Enhancement System

Speech enhancement can be interpreted as the process that maps a noisy signal to a separated signal with improved intelligibility and/or perceptual quality. Without considering the impact of phase, this is often treated as the estimation of clean speech magnitude or ideal masks. Supervised speech enhancement formulates this as a supervised learning problem that the mapping is explicitly learned from data. Acoustic features extracted from a mixed signal, along with the corresponding desired outputs are fed into a learning machine for training. Enhanced speech is obtained by sending estimated outputs and mixture phase into a resynthesizer.

Fig. 1 shows the diagram of the evaluation system, which consists of the feature extraction component and the multiple layers' perceptron (MLP) classification component. We extract acoustic features from an input signal at the frame level, which are sent to an MLP classifier for ideal mask estimation. Common masks contain ideal binary mask (IBM) [24], target binary mask (TBM) [25], ideal ratio mask (IRM) [26] etc. IRM was the best target in mask–based speech enhancement proved in [27].

Several classical acoustic features are used in the baseline system, including Amplitude Modulation Spectrogram (AMS)[28], Relative Spectral Transformed Perceptual Linear Prediction Coefficients (RASTA-PLP) [29-30], and Gamma-tone filterbank power spectra (GF) [31].

To further incorporate temporal context, a 5-frame window of features are input to the DNNs. The output of the network is composed of the corresponding 5-frame window of IRM. The enhanced signals are resynesized by the IRM prediction.

Additionally, we use two DNN training strategies, Restricted Boltzmann Machine (RBM) based pretraining [6] and dropout with ReLU [32-33]. Restricted Boltzmann Machine (RBM) pre-training is used to avoid falling into local minima. Dropout is to overcome the over-fitting in DNN training. ReLU can realize parameter sparsity through simple thresholding activation. Hence, it can speed DNN training, improve generalization and alleviate gradient vanish problem.

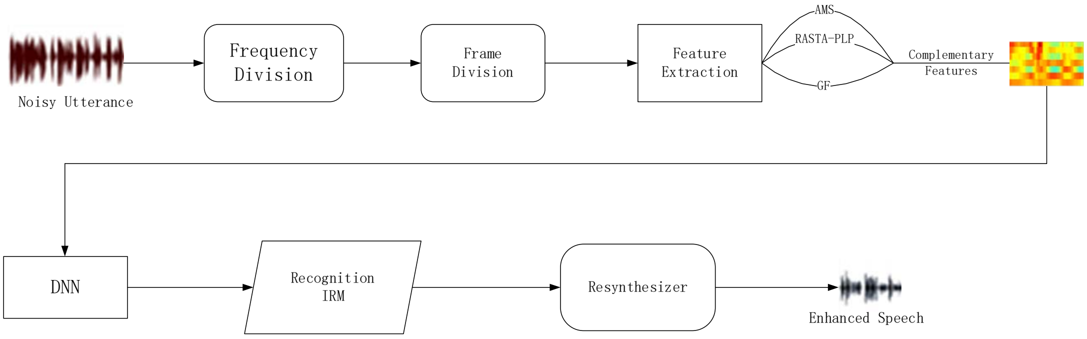


Fig. 1. Speech Enhancement System based on DNN and masks

3. Proposed Methods

In this paper, we propose to adapt deep neural network (DNN) acoustic models to a target noise type or signal-to-noise ratio (SNR) by supplying noise identity vectors (i-vectors) as input features to the network in parallel with the regular acoustic features as mentioned in Section 1. Additionally, to ameliorate the influence of mismatch conditions, we use regularization methods of conservative training. Combing these two methods above, the performance of speech enhancement system in mismatch conditions is expected to improve significantly.

3.1 Noise-aware Training (NaT) with i-vectors

Some sub-space adaptation methods explicitly estimate the noise or speaker information from the utterance and provide this information to the network. It is hoped that the DNN training algorithm can automatically figure out how to adjust the model parameters to exploit the noise, speaker, or device information. We call such approaches noise-aware training (NaT) when the noise information is used, which is very similar to speaker aware training (SaT) or device aware training (DaT). The only difference lies in the auxiliary input.

In speech recognition, i-vectors are used to represent speaker information. That model is trained of training sentences of many speakers and few noise types, so that the i-vectors contain more speaker information and work well in speaker adaptation. In speech enhancement, we can transfer this idea to make i-vectors represent noise information for speech enhancement. In this paper, we choose training sentences of few speakers to mix many types of noises, in order that the extracted i-vectors can well represent noise information. By adding i-vectors as a complementary feature, noise information is fed into the DNN along with other acoustic features. The IRM is learned from all the features, so it contains environment information which is helpful in mismatch conditions.

3.1.1 Extracting i-vector

I-vector maps the input high dimensional feature vector into a low dimensional feature space, retaining most information of the input feature. Let $M(h)$ denotes the mean supervector in a GMM which is related to languages and channels. Then for a voice segment:

$$\boldsymbol{M}(h) = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w}(h) \tag{1}$$

where $\boldsymbol{m}$ is a mean supervector independent with languages and channels, $\boldsymbol{T}$ is a matrix representing R spanning subspace with important variability in the mean supervector space, $\boldsymbol{w}(h)$ is a hidden variable of a normal distribution, and i-vector is the point-estimation of $\boldsymbol{w}(h)$.

Let $\boldsymbol{\chi}\left(h\right) = \left\{\boldsymbol{x}_t^h\right\}, t = 1,2,\cdots,T$ be the feature vector of a training voice segment h. Let $P\left(\boldsymbol{\chi}(s)\middle|\boldsymbol{\Omega},\boldsymbol{M}(s),\boldsymbol{\Sigma}\right)$ be the likelihood of $\boldsymbol{\chi}(s)$ calculated under the model assumption of i-vector, where h is the index of training speech, t is the index of frames of speech feature, and T is the length of the voice segment. The objective function of i-vector is maximizing the overall probability of all the voice segments, i.e.:

$$\prod_h \int_{\boldsymbol{w}(h)} P\left(\boldsymbol{\chi}\left(h\right)\middle|\boldsymbol{\Omega},\boldsymbol{m}+\boldsymbol{T}\boldsymbol{w}(h),\boldsymbol{\Sigma}\right) P(\boldsymbol{w}(h))d\boldsymbol{w}(h) \tag{2}$$

This optimization problem can be solved by EM algorithm as following [6]:

1) The E-Step: For each voice segment h, using the current estimates of $\boldsymbol{T}$, and the prior $\mathcal{N}(w(h)|\mathbf{0},\boldsymbol{I})$ to calculate the posterior distribution of $\boldsymbol{w}(h)$ as

$$P\left(\boldsymbol{w}\left(h\right)\middle|\boldsymbol{\Omega},\boldsymbol{m}+\boldsymbol{T}\boldsymbol{w}(h),\boldsymbol{\Sigma},\boldsymbol{\chi}\left(h\right)\right) \tag{3}$$

2) The M-Step: Update $\boldsymbol{T}$ by maximizing Equation (4):

$$\sum_h \int_{\boldsymbol{w}(h)} \left\{ \begin{array}{l} P\left(\boldsymbol{w}\left(h\right)\middle|\boldsymbol{\Omega},\boldsymbol{m}+\boldsymbol{T}\boldsymbol{w}(h),\boldsymbol{\Sigma},\boldsymbol{\chi}\left(h\right)\right) \\ \quad \times \log P\left(\boldsymbol{\chi}\left(h\right)\middle|\boldsymbol{\Omega},\boldsymbol{m}+\boldsymbol{T}\boldsymbol{w}\left(h\right),\boldsymbol{\Sigma}\right) \end{array} \right\} d\boldsymbol{w}\left(h\right) \tag{4}$$

Then $\boldsymbol{w}(h)$ can be obtained using Equation(5):

$$\boldsymbol{w}\left(h\right) = \arg\max P\left(\boldsymbol{\chi}\left(h\right)\middle|\boldsymbol{\Omega},\boldsymbol{m}+\boldsymbol{T}\boldsymbol{w},\boldsymbol{\Sigma}\right) \tag{5}$$

In previous researches, i-vectors are often used for representing speaker information. In our experiments, we choose training sentences of few speakers in many noise conditions in order to make the extracted i-vector represent more about noise environment rather than the speaker information.

3.1.2 I-vectors as auxiliary features

Classical DNN-based ideal mask estimation uses common short time acoustic features without any speaker and channel information. In our method, to reduce the influence of noise mismatch conditions, i-vector is extracted as a long-time feature to represent noise characteristics. The typical process of extracting i-vector feature and adding it to DNN input is showed in **Fig. 2**.

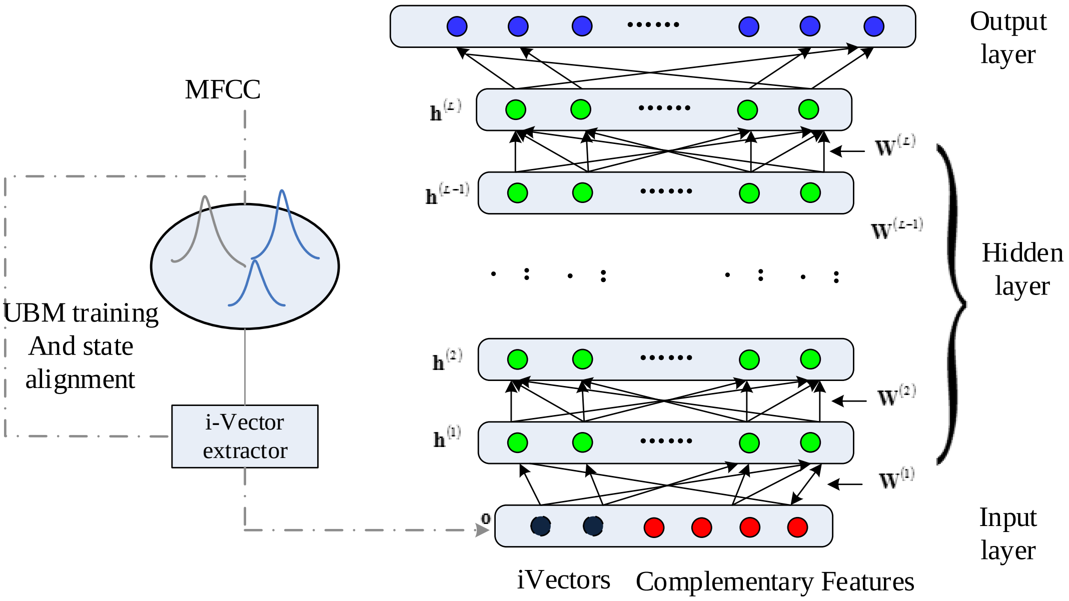


Fig. 2. extracting i-vector based on DNN

The right network in **Fig. 2** shows the DNN estimating ideal masks for speech enhancement. Classical features, including AMS, RASTA-PLP and GF, along with their delta, are concatenated to obtain the F dimensional input features. The ideal ratio mask (IRM), which is obtained by a G-channel Gammatone filterbank and in the range of [0, 1], is used as the output of the network.

The universal background model (UBM) are constructed using all the training data with feature of MFCCs. Using the algorithm mentioned in Section 3.1.1, utterance-level C-dimensional i-vectors are extracted. And for both training and testing, the i-vector for a given noise condition is concatenated to every-frame features. In order to further incorporate temporal context, for the other several features, we splice a 5-frame window of features as the input of DNN to estimate masks. So the DNN input is $C+5*F$ dimensional. The method only needs to increase the number of neurons of input without changing original algorithm.

I-vector, as an additional utterance-level feature, is conducive to speaker, channel and background normalization. Recently, i-vector has achieved great success in speech recognition domain. In this paper, we apply i-vector to speech enhancement.

3.2 L_2 regularization

An obvious approach to adapting DNNs is adjusting all the DNN parameters with the adaptation data, starting from the noise-independent (NI) model. However, doing so may destroy previously learned information and overfit the adaptation data, esp. if the adaptation set is small. To prevent this, adaptation needs to be done conservatively. The technique here does exactly this.

Conservative training (CT)[35-37] is a kind of adaptation techniques often used in the situation that the adaptation set size is small comparing with the number of DNN parameters. CT can be achieved by adding regularizations to the adaptation criterion, such as Kullback-Leibler divergence (KLD) [37] regularization, L_1 regularization[38], and L_2 regularization[35]. An alternative CT technique is adapting only selected weights [39]. Adaptation with very small learning rate and an early stop can also be considered as CT.

The technique developed in this paper adapts the model conservatively by forcing the output vector estimated from the adapted model to be close to that from the unadapted model. By doing this, the trained model can be tuned to adapt the test environment with a small amount of adaptation data, which avoids large data and long training time and can improve the test performance in mismatch conditions. This constraint is realized by adding regularization to the adaptation criterion, because the DNN of our system works as a regressor, aiming to estimate the IRM as the output. Comparing to KLD normalization in [37], we select a more suitable normalization function for mask estimating. We have chosen three linear regularization functions: L_1 regularization, L_2 regularization and elastic net regularization[40], and tested them respectively like that of [41]. The results showed that the L_2 regularization has a better performance and is easy to calculate, so we select L_2 regularization and apply it to speech enhancement domain firstly.

The intuitive explanation of L_2 regularization is: the distance of the output vectors estimating from adaptive model should not differ too greatly from that estimating from unadapted model. The output of DNN is a vector, a measurement of whose distance is L_2 norm. We add L_2 regularization to the adaptive criteria as a regular term to obtain the regular term as follows:

$$J_{L_2}(W,b;N)=(1-\lambda)J(W,b;N)+\lambda R_{L_2}(W_{NI},b_{NI};W,b;N) \qquad (11)$$

where λ is the regularization weight.

$$R_{L_2}(W_{NI},b_{NI};W,b;N)=\frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\|v_{NI}^{L}-v_{m}^{L}\|^2 \qquad (12)$$

where v_{NI}^{L} and v_{m}^{L} are probability that the m^{th} output vector estimating from noise independent DNN and adaptation DNN respectively. The mean square error (MSE) criterion for regression tasks is usually used.

$$J_{MSE}(W,b;N)=\frac{1}{M}\sum_{m=1}^{M}J_{MSE}(W,b;v^{m},y^{m}) \qquad (13)$$

where

$$J_{MSE}(W,b;v^{L},y)=\frac{1}{2}\|v^{L}-y\|^2=\frac{1}{2}(v^{L}-y)^{T}(v^{L}-y) \qquad (14)$$

After adding L_2 regularization, the regularized adaptation criterion can be converted to

$$\begin{aligned} J_{L_2-MSE}(W,b;N)&=(1-\lambda)J_{MSE}(W,b;N)+\lambda R_{L_2}(W_{NI},b_{NI};W,b;N)\\ &=\frac{1}{2M}\sum_{m=1}^{M}(\,1\text{-}\lambda)\|v_{m}^{L}-y\|^2+\lambda\|v_{NI}^{L}-v_{m}^{L}\|^2 \end{aligned} \qquad (15)$$

3.3 Combination of NaT and L_2 regularization

The NaT method consists in providing noise i-vectors alongside aforementioned features as inputs to the neural net. The training and test data are augmented with these i-vectors which are constant for a given noise and change across different noises.

The L_2 regularization is added to the MSE criterion to adapt the model. This can be applied to DNN adaptation via back propagation (BP) algorithm, only modifying the error signals. The interpolation weight, which is directly derived from the regularization weight λ, can be adjusted, typically using a development set, based on the size of the adaptation set, the learning rate, and whether the adaptation is supervised or unsupervised. When $\lambda=1$, we trust completely the NI model and ignore all new information from the adaptation data. When $\lambda=1$, we adapt the model solely on the adaptation set, ignoring information from the NI model except using it as the starting point. Intuitively, we should use a large λ for a small adaptation set and a small λ for a large adaptation set. In our experiment, we try different λ for every situation and test the performance separately.

This paper combines NaT with L_2 regularization for noise adaptive speech enhancement showed in **Fig. 3**. NaT can help bring noise and channel information into account, and L_2 regularization can use data in development set to adapt the DNN. The combinations of these two methods can effectively utilize the advantages of each other, leading to better speech enhancement results. **Fig. 4** and **Fig. 5** is a diagram of speech enhancement. **Fig. 4** is the comparison of the waveforms of noisy speech (the upper) and the enhanced speech (the lower). **Fig. 5** is the comparison of the spectrums of noisy speech (the upper) and the enhanced speech (the lower). From the two figures, we can see that our method does work in reducing the noise.

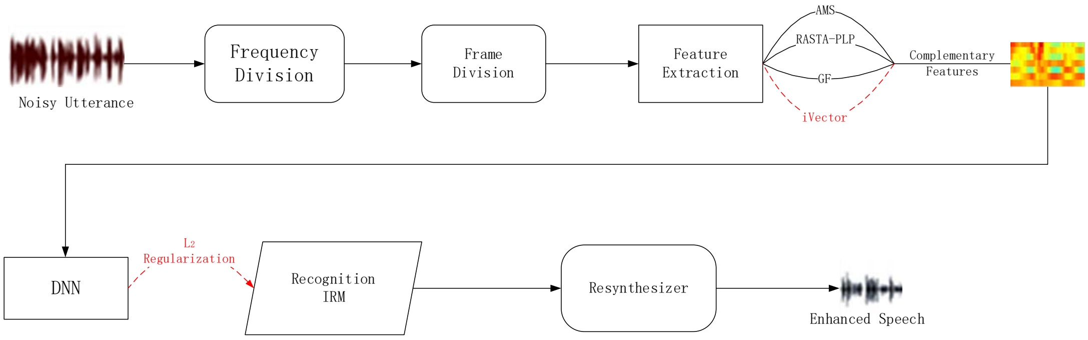


Fig. 3. Proposed System

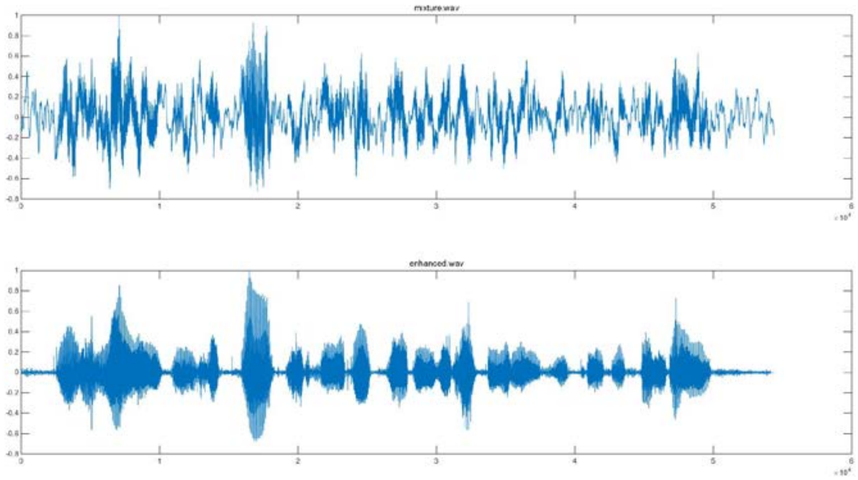


Fig. 4. Waveform Comparison

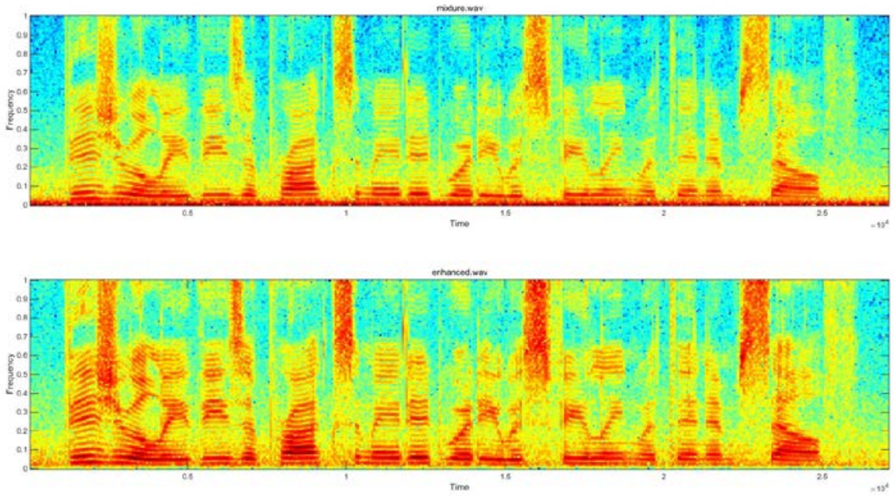


Fig. 5. Spectrum Comparison

4. Experimental and Results

4.1 Experimental Settings

4.1.1 Data

All the experiments are based on Voice Bank corpus [42] speech data set. We choose eight types of noises from NOISEX-92[43]: factory1, destroyer engine, Volvo, f16, m109, destroyer ops, and white for baseline system. And we choose 2520 clean speech utterances of 6 speakers (including both genders) from Voice Bank corpus to mix with the noises aforementioned at different SNRs (-5dB, 0dB, 5dB, 10dB, 15dB and 20dB) to obtain a parallel training dataset and choose 100 utterances to obtain adaptive dataset. Other 200 clean utterances are chosen to mix with these noises at different SNRs to constitute the core testing dataset.

All the noises are 235 seconds long. To avoid overlapping between training and testing noise, we cut the noise into two halves. When generating training set, we use random cuts from the first 2 minutes of each noise to mix with the training utterances at -5, 0, 5, 10, 15 and 20 dBs. The testing mixtures are constructed by mixing random cuts from the last 2 minutes of each noise with the testing utterances at -5, 0, 5, 10, 15 and 20 dBs The adaptive set is generated as the training set.

4.1.2 Parameters settings

As showed in **Fig. 1** in Section 2, the noisy signals with 16 kHz sample rate are passed through the predesigned 64-channel gamma-tone filter bank. After obtaining 64 sub bands, speech signals are divided into frames with 20ms frame length and 10ms frame shift. Then frame-level features are extracted and concatenated with their corresponding delta portions. To encode more information, we use a series of features (15-dimensional AMS, 13-dimensional RASTA-PLP and 64-dimensional GF). In total, the features are 92 dimensional. Adding their delta, the features are 184 dimensional. So as mentioned in Section 3.1.2, the feature dimensional F is equal to 184. Splicing 5-frame window, the feature is 5*184 =920 dimensional. [21] discussed the effect of having different i-vector dimensions and proved that having an i-vector dimension of 100 is a reasonable choice. So using the algorithm mentioned in Section 3.1.1, utterance-level 100-dimensioanl i-vectors are extracted, indicating that C is equal to 100. So the input features are 5*184+100=1020 dimensional. The DNN, as a regressor, is used to learn the estimated ratio mask (ERM) of every frequency band. The DNN includes 3 hidden layers, each of which has 1024 units, and an input layer and an output layer. The output of the DNN is 64 dimensional Gamma-tone filterbank. IRM, ranging from 0 to 1, is used as the target function. Sigmoid function is chosen to be the output function.

Short time Fourier analysis is used to compute the coefficients of each overlapped frame. We use the adaptive gradient descent (AGD) [44] along with a momentum term as the optimization technique. A momentum rate of 0.5 is used for the first 5 epochs, after which the rate increases to and is kept as 0.9. The DNNs are trained to predict the desired outputs across all frequency bands, and the mean squared error (MSE) is used as the cost (loss) function for this regression task.

4.2 Evaluation Criteria

For evaluation metrics, we use the Short-Time Objective Intelligibility score (STOI) [45]to measure the objective intelligibility. We also evaluate objective speech quality using the Perceptual Evaluation of Speech Quality (PESQ) [46] score. Like STOI, PESQ is obtained by comparing the separated speech with the corresponding clean speech. The STOI score ranges from 0 to 1 and PESQ score −0.5 to 4.5. Segmental SNR (segSNR) [47-48] is used to evaluate SNRs of every segment.

4.2.1 Segmental SNR

$$SegSNR = \frac{1}{T}\sum_{t=0}^{T-1} \varsigma_1 \left\{ 10\log \frac{\sum_{l=0}^{L-1}\left[x^t(l+tR)\right]^2}{\sum_{l=0}^{L-1}\left[x^t(l+tR)-\widehat{x}^t(l+tR)\right]^2} \right\} \tag{11}$$

where T represents for the number total frames, and ς is a meaningful range of SNRs for human auditory. Segmental SNR, the ratio of the signal to its delta, is a time-domain metric representing for the extent of denoising.

4.2.2 PESQ

PESQ is an application guide for objective quality measurement based on recommendations P.862 proposed by International Telecommunication Union (ITU). It is an objective metric while being consistent with mean opinion score (MOS) [49]. We regular the voltage and time, then transform the human auditory, including to Bark domain, and use cognitive modeling and distance measure. As a main target of speech enhancement, PESQ measures the quality of the speech.

4.2.3 STOI

STOI was proposed recently to evaluate the intelligibility of speech. The two aspects of speech are the quality and intelligibility. STOI is more meaningful in low SNRs because it is not difficult to understand the content in high SNRs. But in low SNRs, the speech is destroyed heavily, so it is important to improve the intelligibility to make people understand it.

4.3. Results and Analysis

To compare the performance in different situations, we train two sets of DNNs. The first set uses mixtures at variable SNRs and test at different SNRs. The second set uses mixtures of variable types of noises and test with different types of noise. For every experiment, we test the performance after adjusting the weight of normalization λ and choose the best evaluation results to fill up the tables.

4.3.1 Comparison between Various Systems in Mismatching SNRs

For the first set, aiming at the SNR mismatching problem, we make 3 experiments. In the 1st experiment, the DNN is trained with 5 dB mixtures, and tested with -5 dB mixtures. 8 types of noises are used to do the experiment separately and the results are showed in **Table 1**. In the 2nd experiment, the DNN is trained with 0, 5 and 10 dB mixtures of the same type of noise, and tested by -5 dB mixtures. 6 types of noises are used to separately do the experiment and the results are showed in **Table 2**. In the 3rd experiment, the training mixtures are 10, 15 and 20 dB mixtures, and the testing mixtures are 5 dB mixtures of the same type of noise. Other experimental setups are exactly the same as the 2nd experiment and the results are showed in **Table 3**.

Table 1. Performance Comparisons between Various Systems in Mismatching SNRs
(5 dB training and -5 dB testing)

System	**M109**			**Factory1**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.6790	1.0516	-6.0876	0.5355	1.0416	-6.1544
baseline	0.7223	1.2686	-1.5862	0.5952	1.1233	-2.2387
baseline+iVector	0.7265	1.2759	-1.6137	0.5934	1.0980	-2.6466
Baseline+L_2	**0.7547**	**1.3578**	**-0.2009**	**0.6112**	**1.1475**	**-1.2613**
Baseline+iVector+ L_2	0.7540	1.3486	-0.4217	0.6003	1.1283	-1.8179
System	**Volvo**			**F16**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.8881	1.2875	-5.3890	0.5707	1.0466	-6.2958
baseline	0.8965	1.9560	5.5632	0.5794	1.0735	-5.2537
baseline+iVector	0.8942	1.9538	5.3847	0.5830	1.0755	-5.2833
Baseline+L_2	**0.9135**	**2.5192**	**8.5799**	**0.6726**	**1.2342**	**-1.9867**
Baseline+iVector+ L_2	0.9076	2.3616	7.0332	0.6685	1.2079	-1.9892
System	**Destroyer engine**			**White**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.5711	1.0818	-6.3445	0.6048	1.0339	-6.3173
baseline	0.6578	1.1831	-3.1487	0.6402	1.0446	-4.8384
baseline+iVector	0.6686	1.1874	-3.1298	0.6455	1.0443	-5.0906
Baseline+L_2	**0.6900**	**1.2911**	**-1.3051**	**0.6627**	**1.0514**	**-4.6073**
Baseline+iVector+ L_2	0.6897	1.2183	-2.2845	0.6619	1.0484	-4.8821

System	babble			Destroyer ops		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.5482	1.0636	-6.1778	0.5950	1.0381	-6.2963
baseline	0.5490	1.0821	-5.4004	0.5970	1.0773	-4.4975
baseline+iVector	0.5470	1.0846	-5.4464	0.5954	1.0814	-4.5150
Baseline+L_2	**0.5743**	**1.1126**	**-4.0502**	**0.6728**	**1.1783**	**-1.7490**
Baseline+iVector+ L_2	0.5737	1.1085	-4.1008	0.6590	1.1424	-2.9623

Table 2. Performance Comparisons between Various Systems in Mismatching SNRs (0, 5, 10 dB training and -5 dB testing)

System	M109			Factory1		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.6790	1.0516	-6.0876	0.5355	1.0416	-6.1544
baseline	0.6795	1.1237	-3.2301	0.6179	1.1203	-2.1028
baseline+iVector	0.6786	1.1226	-3.3632	0.6222	1.1228	-1.9644
Baseline+L_2	**0.7449**	**1.3504**	**-0.4237**	**0.6223**	**1.1303**	**-1.9131**
Baseline+iVector+ L_2	0.7434	1.3160	-0.7653	**0.6223**	1.1224	-2.2653

System	Volvo			F16		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.8881	1.2875	-5.3890	0.5707	1.0466	-6.2958
baseline	0.8964	1.9586	5.5892	0.5856	1.0761	-5.2650
baseline+iVector	0.8938	1.9468	5.3933	0.5809	1.0721	-5.2639
Baseline+L_2	**0.9135**	**2.3869**	**7.4149**	**0.6736**	**1.2023**	**-1.8900**
Baseline+iVector+ L_2	0.9125	2.3729	7.3493	0.6712	1.1900	-2.1219

System	Destroyer engine			White		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.5711	1.0818	-6.3445	0.6048	1.0337	-6.3173
baseline	0.5765	1.0876	-5.4134	0.6003	1.0700	-4.2564
baseline+iVector	0.5779	1.0858	-5.4668	0.6002	1.0682	-4.3443
Baseline+L_2	**0.6724**	**1.2666**	**-1.1864**	**0.6838**	**1.1570**	**-1.3488**
Baseline+iVector+ L_2	0.6665	1.2217	-2.0689	0.6765	1.1223	-1.4014

Table 3. Performance Comparisons between Various Systems in Mismatching SNRs (10, 15, 20 dB training and 5 dB testing)

System	M109			Factory1		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.8532	1.2944	-1.0543	0.7702	1.1656	-1.1478

baseline	0.8492	1.4834	2.1093	0.7650	1.2157	-0.3740
baseline+iVector	0.8508	1.4836	2.1526	0.7669	1.2189	-0.3498
Baseline+L_2	**0.8824**	**1.9730**	**4.9517**	**0.8075**	**1.4327**	**2.0270**
Baseline+iVector+ L_2	0.8796	1.8994	4.7737	0.8053	1.4193	2.0259
System	**Volvo**			**F16**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.9522	2.6021	8.9953	0.8013	1.1898	-1.3014
baseline	0.9527	2.6823	10.0384	0.8085	1.2670	0.0305
baseline+iVector	0.9508	2.6541	9.8302	0.8084	1.2610	-0.0981
Baseline+L_2	**0.9589**	**3.0688**	**12.2628**	**0.8552**	**1.6563**	**2.5135**
Baseline+iVector+ L_2	0.9579	3.0263	12.1459	0.8501	1.6165	2.4575
System	**Destroyer engine**			**White**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.8032	1.2496	-1.3883	0.8324	1.0853	-1.4292
baseline	0.8110	1.2855	-0.5887	0.8847	1.7131	4.1640
baseline+iVector	0.8050	1.2755	-0.7296	0.8844	1.7129	4.1624
Baseline+L_2	**0.8624**	**1.6364**	**2.0512**	**0.8867**	**1.8480**	**4.6512**
Baseline+iVector+ L_2	0.8543	1.6316	1.9150	0.8866	1.8469	4.6234

Comparing the results in Section 4.3.1, we can reach the following conclusions.

For the noise mismatch condition of SNRs in the 1st set, on all of these noises, L_2 regularization improves the performance consistently while the i-vector method does not. The improvements show that i-vectors cannot reflect the variation of the SNRs, while L_2 regularization can take adaptation information into account to make the original model adapt the test set well. And the decrease of i-vector system also shows that as the dimensional of features increases, it becomes harder to tune the network for adaptation.

Table 1 shows the results under mismatch noise conditions, where the model is trained of the same type of noise under one higher SNR (5dB) and tested under one lower SNR (-5dB). Eight types of noises are used. Among them, babble noise performs worst, whose STOI score is less than 0.6 and PESQ score is around 1.1. It sounds hard to understand and bears signal distortion. This is probably because babble noise itself is speech noise which is similar to speech, the only difference lies in SNRs. When the environment of SNRs changes, the learned information doesn't work in testing. So the performance decreases seriously. Besides babble noise, for m109 and volvo noises, the performance is better than other types of noises. Especially the Volvo noise, the STOI of which is greater than 0.9 and the PESQ is greater than 2.5 and it sounds natural with little degradation and little noticeable noise. The m109 and Volvo noise are recorded in the tank in the speed of 30 km/h and the Volvo 340 car in the speed of 120 km/h respectively. So these two types of mechanical noises are more stationary than others. Adding L_2 regularization, f16 and destroyer ops noises increase most: the STOI promotion percentages are 9.32% and 7.58%, and the PESQ promotion are both greater than 0.1. This shows that the more unstationary, the L_2 regularization works better. The performance with L_2 regularization on white noise is not so obvious, because the correlation of white noise spectrum is not so close.

Table 2~3 show the under mismatch noise conditions, where the model is trained of the same type of noise under three higher SNR (0, 5, 10dB) and tested under one lower SNR (-5dB). In **Table 2**, STOI increases most on destroyer engine noise (9.59%), and the PESQ increases most on Volvo noises (0.43). So we can see on unstationary noises, L_2 regularization can promote human intelligibility obviously and make people understand the speech content better. In the environment with stationary noise where the speech is not so bad and easier to understand, the L_2 regularization can improve the quality of the speech, making the speech more natural and clearer. In **Table 3**, STOI increases most on destroyer engine noise (5.14%), and the PESQ increases most on m109 noise (0.49). From it we can get similar conclusions with it in **Table 2**. Comparing **Table 2~3**, we can see the promotion of the percentage of STOI is getting smaller as the SNR increases. The promotion of STOI in the 3rd experiment is smaller than that in the 2nd experiment. The trend of the other two criteria, PESQ and segSNR, is consistent with the STOI.

In **Table 1~3**, the performance of Volvo noise is better than other noises in all the evaluation criteria. For example, in **Table 1**, the STOI of Volvo noise can reach more than 0.9 while others can reach only 0.7 at most. It is probably because the Volvo noise is a rather stationary noise recorded in the car in the stable road.

4.3.2 Comparison between Various Systems on Mismatching Noise Types

For the second set, aiming at the multi-type noises problem, a DNN is trained with three types of noises, including Volvo, factory1 and f16 noises, and tested with m109 noise, destroyer engine noise， white noise and destroyer ops noise, whose results are showed in **Table 4~7** respectively.

In the second set, comparing **Table 4~7**, we use the same model trained by three types of noises to test other four types of noises separately. Among the test noises, factory1 noise, destroyer engine noise, Volvo noise, m109 noise and f16 noise have similarities in spectrum. Destroyer ops noise has the similar characteristic of these noises, and it contains human speech at the same time. White noise is a stationary noises acquired by sampling high-quality analog noise generator. So in the second set, three of them are chosen to train the DNN, while the other two, along with white noise and destroyer ops noise are chosen to test.

Table 4. Performance Comparisons between Various Systems on Mismatching noise types (Volvo, factory1, f16 training and m109 testing)

System	**-5dB**			**0dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.6790	1.0516	-6.0876	0.7714	1.1156	-3.9650
baseline	0.6812	1.1200	-3.2957	0.7714	1.2492	-0.7065
baseline+iVector	0.6821	1.1219	-3.2725	0.7733	1.2456	-0.7079
Baseline+L_2	0.7316	1.3515	-0.4835	0.8146	1.5958	1.8081
Baseline+iVector+ L_2	**0.7319**	**1.3655**	**-0.2407**	**0.8150**	**1.6089**	**2.0690**
System	**5dB**			**10dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.8532	1.2944	-1.0543	0.9153	1.6010	2.3919

baseline	0.8882	1.8982	4.6374	0.9345	2.3742	7.6857
baseline+iVector	0.8901	1.9740	5.1397	0.9349	2.4620	8.2688
Baseline+L_2	0.8910	2.0187	5.0738	0.9350	2.4681	8.0933
Baseline+iVector+ L_2	**0.8913**	**2.0467**	**5.5093**	**0.9354**	**2.5365**	**8.5326**
System	**15dB**			**20dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.9587	2.0264	6.2021	0.9824	2.5266	10.2699
baseline	0.9672	2.9138	11.1998	0.9849	3.3764	14.4958
baseline+iVector	0.9672	3.0259	11.6788	0.9847	3.5200	14.9065
Baseline+L_2	**0.9676**	2.9902	11.5471	**0.9851**	3.4442	14.7668
Baseline+iVector+ L_2	**0.9676**	**3.1184**	**12.0644**	**0.9851**	**3.5695**	**15.0562**

Table 5. Performance Comparisons between Various Systems on Mismatching noise types (Volvo, factory1, f16 training and destroyerengine testing)

System	**-5dB**			**0dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.5711	1.0818	-6.3445	0.6906	1.1321	-4.2368
baseline	0.5665	1.1265	-6.1808	0.6865	1.1425	-4.0871
baseline+iVector	0.5681	1.1299	-6.1139	0.6908	1.1711	-4.1593
Baseline+L_2	0.6643	1.1760	-2.5357	0.8002	1.5327	0.4656
Baseline+iVector+ L_2	**0.6724**	**1.2799**	**-1.0798**	**0.8054**	**1.1731**	**0.7320**
System	**5dB**			**10dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.8532	1.2944	-1.0543	0.8897	1.4805	2.0355
baseline	0.7995	1.3332	-1.3552	0.8861	1.5686	1.8701
baseline+iVector	0.8001	1.3358	-1.2231	0.8863	1.5718	1.9493
Baseline+L_2	0.8082	1.3157	-0.9444	0.9173	2.1726	6.1442
Baseline+iVector+ L_2	**0.8485**	**1.5246**	**0.9305**	**0.9233**	**2.2403**	**6.4789**
System	**15dB**			**20dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.9474	1.8443	15.0103	0.9773	2.3532	9.8870
baseline	0.9428	1.9198	5.2521	0.9743	2.4000	8.6020
baseline+iVector	0.9437	1.9203	5.2022	0.9756	2.4038	8.7217
Baseline+L_2	0.9592	2.6586	8.9157	0.9812	3.1315	12.2010
Baseline+iVector+ L_2	**0.9604**	**2.6710**	**9.2597**	**0.9813**	**3.1727**	**12.4489**

Table 6. Performance Comparisons between Various Systems on Mismatching noise types (Volvo, factory1, f16 training and white testing)

System	**-5dB**			**0dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.6048	1.0339	-6.3173	0.7222	1.0467	-4.2430
baseline	0.5976	1.0473	-5.5414	0.7173	1.0577	-4.0562
baseline+iVector	0.6008	1.0485	-5.3931	0.7217	1.0569	-4.2098
Baseline+L_2	0.6581	1.0576	-4.9445	0.7760	1.0860	-2.3037
Baseline+iVector+ L_2	**0.6680**	**1.0642**	**-4.7134**	**0.7770**	**1.0969**	**-1.7046**
System	**5dB**			**10dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.8324	1.0853	-1.4292	0.9123	1.1944	1.9483
baseline	0.8269	1.1181	-0.7996	0.8863	1.5686	1.8701
baseline+iVector	0.8252	1.1191	-0.6143	0.8861	1.5718	1.9493
Baseline+L_2	0.8506	1.1548	0.0094	0.9206	1.3021	3.2077
Baseline+iVector+ L_2	**0.8528**	**1.1636**	**0.1558**	**0.9211**	**1.3091**	**3.3096**
System	**15dB**			**20dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.9609	1.4305	5.7321	0.9844	1.8310	9.7921
baseline	0.9571	1.4810	5.3151	0.9828	1.8717	8.6855
baseline+iVector	0.9559	1.4839	5.4216	0.9814	1.8710	8.5745
Baseline+L_2	**0.9644**	1.5727	6.6155	**0.9840**	2.0207	10.2580
Baseline+iVector+ L_2	**0.9644**	**1.5812**	**6.6251**	**0.9840**	**2.0431**	**10.3858**

Table 7. Performance Comparisons between Various Systems on Mismatching noise types (Volvo, factory1, f16 training and destroyerops testing)

System	**-5dB**			**0dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.5950	1.0382	-6.2963	0.6959	1.0702	-4.1983
baseline	0.5966	1.0769	-4.4769	0.6972	1.1058	-2.4657
baseline+iVector	0.5969	1.0776	-4.4122	0.6984	1.1035	-2.5758
Baseline+L_2	0.6646	1.1533	-2.1086	0.7513	1.2477	-1.1689
Baseline+iVector+ L_2	**0.6694**	**1.1563**	**-1.8750**	**0.7575**	**1.2606**	**-0.1320**
System	**5dB**			**10dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.7908	1.1738	-1.3151	0.8655	1.3907	2.1103

baseline	0.7904	1.3016	0.6779	0.8636	1.5674	3.6694
baseline+iVector	0.7942	1.3064	0.5874	0.8669	1.5657	3.7479
Baseline+L_2	0.8299	1.5028	2.4903	0.8887	1.9423	5.9214
Baseline+iVector+ L_2	**0.8333**	**1.5097**	**2.5930**	**0.8929**	**1.9672**	**6.2085**
System	**15dB**			**20dB**		
	STOI	PESQ	segSNR	STOI	PESQ	segSNR
mixture	0.9246	1.7892	5.8962	0.9640	2.3340	9.9634
baseline	0.9233	1.9646	6.8020	0.9629	2.4723	9.8982
baseline+iVector	0.9245	1.9693	6.8543	0.9637	2.4783	10.0437
Baseline+L_2	0.9360	2.4283	8.8215	0.9671	2.9086	12.0769
Baseline+iVector+ L_2	**0.9371**	**2.4531**	**9.0679**	**0.9686**	**2.9543**	**12.0911**

For the noise mismatch condition of noise types in the 2nd set, on all of these types of noises, i-vectors and L_2 separately improve all the three evaluation criteria. The range of promotion of L_2 regularization is larger than that of i-vector method. And it can get better performance when combing these two methods. The factor contributed to this phenomenon is that i-vectors can reflect noise type information, but not so sensitive to SNRs. And L_2 regularization, which brings about testing information, is still effective when the dimensional of features increases.

Table 4~7 show the under mismatch noise conditions, where the model is trained of three different types of noises (Volvo, factory1 and f16 noises) under the same SNR and tested with another type of noise. **Table 4~7** show the results of m109, destroyer engine, white and destroyer ops noises testing respectively, and experiments in six different SNRs (-5, 0, 5, 10, 15 and 20dBs) are done in each of them. Volvo, factory1 and f16 noises are all mechanical noises, which are similar to m109 and destroyer engine noises. Comparing **Table 4~7**, m109 noise in **Table 4** performs best, in which the STOI can reach 0.7 in -5dB. This is because m109 noise is a similar type with training noises and it is stationary than other testing noises. Before enhancement, destroyer engine mixture performs the worst, because the destroyer engine noise is an unstationary noise and the speech is destroyed heavily with it. White noise is different from mechanical noises, so when testing, the evaluation scores are even worse than mixture. But when adding i-vector and L2 regularization, it improves 7.04% at most. So this paper concludes that the test sets of similar noises perform better than white noise. Probably the similarities contribute to congenial DNN parameters, which is helpful for enhancement. Among the four types of noises, the STOI improves most on destroyer engine, whose promotion percentage is 11.89% at most, and the PESQ improves most on destroyer engine and destroyer ops, whose promotion are 0.75 and 0.49 respectively at most. So we can see the i-vector and L2 regularization are effective in improving both human intelligibility and speech quality, and it works better on extremely unstationary noises. Notably, destroyer ops noise itself contains mechanical noise and speech noise, making the improvement slightly worse than destroyer engine noise. The reason may lie in that i-vector contains the speaker and environment information, but destroyer ops noise itself includes human speech, so even advanced complementary-feature system is difficult to distinguish target speech from speech noise. In all the conclusions above, the trend of segSNR is consistent with the other two criteria.

In **Table 4~7**, the promotion of the percentage of STOI is getting smaller as the SNR increases on all the testing noise types, which is similar to the results in Section 4.3.1. For example, on m109 noise in **Table 4**, the largest promotion of percentage of STOI is 5.07% at -5dB, while it is less than 0.1% when the SNR is higher than 10dB. Note that in low SNR conditions, STOI improvement is more meaningful. The trend of the other two criteria, PESQ and segSNR, is consistent with the STOI.

From the all the results above in 4.3, some conclusions can be summarized for i-vector and L2 systems. On all the situations above, the improvements decrease as the SNRs increase. Among all experiments, we find that more robust performance and better results appear when λ ranges from 0.125 to 0.5. And the best value of λ is often less than 0.2 for PESQ and segSNR. This is probably because for supervised adaptation system whose labels are relatively reliable, a smaller λ means a larger proportion of adaptation data taking into account, leading to a better results. An exception is that on white noise, the best result appears when λ is greater than 0.5, sometimes reaches 0.9. This is probably because the white noise is a stationary and weakly-relating noise and the regularization does not work well in this particular situation.

5. Conclusion and Expectation

To overcome the mismatching problem between training and testing sets in speech enhancement, we have presented an effective way to perform noise adaptation for neural network acoustic models. We proposed to use NaT in ideal mask estimation system based on DNN to bring environment information into account. We also test the performance of L_2 regularization, which using a small amount of adaptation data to adapt the network. The two methods can be combined to take advantages of both.

There are several potential research directions. In this study, i-vector is extracted by MFCC and considered as an utterance-level feature. Future works we may use dynamic i-vectors to further improve the method. And we can try other CT methods for comparison. And supervised algorithms always face with a problem of lacking effective labels. Sometimes user-provided tags are incomplete, subjective and noisy. Under the circumstances, weakly supervised deep metric learning is proposed and has been applied in image understanding successfully[50-51]. This is worth considering and is hoped to lead to better performance in speech domain.

References

[1] Philipos. C. Loizou, “Speech Enhancement: Theory and Practice,” *2nd ed. Boca Raton, FL, USA: CRC Press*, Inc., 2013. Article (CrossRef Link)

[2] Steven Boll. “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113-122, 1979. Article (CrossRef Link)

[3] J. Lim and A. Oppenheim, “All-pole modeling of degraded speech,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 3, pp. 197–210, Jun 1978. Article (CrossRef Link)

[4] Y. Ephraim, “Statistical-model-based speech enhancement systems,” in *Proc. of Proceedings of the IEEE*, vol. 80, no. 10, pp. 1526–1555, Oct 1992. Article (CrossRef Link)

[5] Kevin W Wilson, Bhiksha Raj, and Paris Smaragdis, “Regularized non-negative matrix factorization with temporal dependencies for speech denoising,” *Interspeech*, pp. 411-414, 2008a. Article (CrossRef Link)

[6] Y. Bengio, “Learning deep architectures for AI,” Found. Trends Mach. Learn., vol. 2, no. 1, pp. 1–127, 2009. Article (CrossRef Link)

[7] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, “Speech enhancement based on deep denoising autoencoder,” *INTERSPEECH*, pp. 436-440, 2013. Article (CrossRef Link)

[8] Bing-yin Xia and Chang-chun Bao, “Speech enhancement with weighted denoising auto-encoder,” *INTERSPEECH*, pp. 3444-3448, 2013. Article (CrossRef Link)

[9] Bingyin Xia and Changchun Bao, “Wiener filtering based speech enhancement with weighted denoising auto-encoder and noise classification,” *Speech Communication*, vol. 60, pp.13-29, 2014. Article (CrossRef Link)

[10] D. L. Wang and G. J. Brown, “Computational auditory scene analysis: Principles, algorithms, and applications,” *Wiley-IEEE Press*, 2006. Article (CrossRef Link)

[11] Kim, D.Y., Kwan Un, C., Kim, N.S., “Speech recognition in noisy environments using first-order vector Taylor Series,” *Speech Communication*, vol. 24, no. 1, pp. 39-49, 1998. Article (CrossRef Link)

[12] Li, J., Deng, L., Yu, D., Gonf, Y., Acero, A., “A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions via vector Taylor series,” in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 65-70, 2007. Article (CrossRef Link)

[13] Li, J., Deng, L., Yu, D., Gong, Y., Acero, A., “HMM adaptation using a phase-sensitive acoustic distortion model for environment-robust speech recognition,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4069-4072, 2008. Article (CrossRef Link)

[14] Moreno, P.J., Raj, B., Stern, R.M., “A vector Taylor series approach for environment-independent speech recognition,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, pp. 733-736, 1996. Article (CrossRef Link)

[15] Seide, F., Li, G., Chen, X., Yu, D., “Feature engineering in context-dependent deep neural networks for conversational speech transcription,” in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ARSU)*, pp. 24-29, 2011. Article (CrossRef Link)

[16] Seltzer, M., Yu, D., Wang, Y., “An investigation of deep neural networks for noise robust speech recognition,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013. Article (CrossRef Link)

[17] Yu, D., Seltzer, M.L., Li, J., Huang, J.T., Seide, F., “Feature learning in deep neural networks-studied on speech recognition tasks,” in *Proc. of International Conference on Learning Representation (ICLR)*, 2013. Article (CrossRef Link)

[18] Abdel-Hamid, O., Jiang, H., “Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7942-7946, 2013. Article (CrossRef Link)

[19] Shaofei Xue, Ossama Abdel-Halmid, Hui Jiang, Lirong Dai, “Direct Adaptation of Hybrid DNN/HMM Model for Fast Speaker Adaptation in LVCSR Based on Speaker Code,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6389-6393, 2013. Article (CrossRef Link)

[20] D. K. Kim and N. S. Kim, “Baysian speaker adaptation based on probabilistic principal component analysis,” *INTERSPEECH*, pp. 734-737, 2000. Article (CrossRef Link)

[21] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, “Speaker adaptation of neural network acoustic models using i-vectors,” in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ARSU)*, pp. 55-59, 2013. Article (CrossRef Link)

[22] Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788-798, 2011. Article (CrossRef Link)

[23] Glembek, O., Burget, L., Matejka, P., Karafiat, M., Kenny, P., “Simplification and optimization of i-vector extraction,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4515-4519, 2011. Article (CrossRef Link)

[24] Wang D, “On ideal binary mask as the computational goal of auditory scene analysis,” *Divenyi, P., editor. Speech Separation by Humans and Machines. Norwell*, MA, USA: Kluwer: pp.181-197, 2005. Article (CrossRef Link)

[25] Kjems U, Boldt, J, Pedersen M, Lunner T, Wang D, “Role of mask pattern in intelligibility of ideal binary-masked noisy speech,” *The Journal of the Acoustical Society of America*, vol. 126, pp. 1415-1426, 2009. Article (CrossRef Link)

[26] Li N, Loizou P, “Factors influencing intelligibility of ideal binary masked speech: Implications for noise reduction,” *The Journal of the Acoustical Society of America*, vol. 123, no. 3, pp. 1673-1682, 2008. Article (CrossRef Link)

[27] Yuxuan Wang, Arun Narayanan, Deliang Wang, “On Training Target for Supervised Speech Separation. IEEE/ACM Trans Audio Speech Lang Process,” vol. 22, no. 12, pp. 1849-1858, Dec 2014. Article (CrossRef Link)

[28] G. Kim, Y. Lu, Y. Hu, and P. C. Loizou, “An algorithm that improves speech intelligibility in noise for normal-hearing listeners,” *The Journal of the Acoustical Society of America*, vol. 126, pp. 1486–1494, 2009. Article (CrossRef Link)

[29] H. Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” *The Journal of the Acoustical Society of America*, vol. 87, pp. 1738–1752, 1990. Article (CrossRef Link)

[30] H. Hermansky and N. Morgan, “RASTA processing of speech,” *IEEE Transactions on Speech, Audio Process*, vol. 2, no. 4, pp. 578–589, Oct 1994. Article (CrossRef Link)

[31] Timo Gerkmann and Richard C Hendriks, “Unbiased mmse-based noise power eatimation with low complexity and low tracking delay,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 4, pp. 1383-1393, 2012. Article (CrossRef Link)

[32] Hinton G.E, Srivastava N, Krizhevsky A, Sutskever I and Salakhutdinov R, “Improving neural networks by preventing co-adaptation of feature detectors,” Canada: Cornell University, [2013-07-3]. Article (CrossRef Link)

[33] Miao Yajie, Metze Florian, “Improving Low-Resource CD-DNN-HMM using Dropout and Multilingual DNN Training,” in *Proc. of Proceedings of 14th Annual Conference of the International Speech Communication Association (INTERSPEECH). Lyon*, France: ISCA, pp. 2237-2241, 2013. Article (CrossRef Link)

[34] Albensano, D., Gemello, R., Laface, P., Mana, F., Scanzio, S., “Adaptation of artificial neural networks avoiding catastrophic forgetting,” in *Proc. of International Conference on Neural Networks (IJCNN)*, pp. 1554-1561, 2006. Article (CrossRef Link)

[35] Li, X., Bilmes, J., “Regularized adaptation of discriminative classifiers,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp. I-I, 2006. Article (CrossRef Link)

[36] Stadermann, J., Rigoll, G., “Two-stage speaker adaptation of hybrid tied-posterior acoustic models,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2005. Article (CrossRef Link)

[37] Yu, D., Yao, K., Su, H., Li, G., Seide, F., “Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7892-7897, 2013. Article (CrossRef Link)

[38] Tibshirani, R., “Regression shrinkage and selection via the lasso,” *Journal of the royal statistical society series b-statistical methodology*, vol. 58, no. 1, pp. 267–288, 1996. Article (CrossRef Link)

[39] J. Stadermann and G. Rigoll, “Two-stage speaker adaptation of hybrid tied-posterior acoustic models,” in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. I, pp. 997-1000, 2005. Article (CrossRef Link)

[40] Christine De Mol, Ernesto De Vito and Lorenzo Rosasco,“Elastic-net regularization in learning theory,” *Journal of Complexity*, vol. 25, issue. 2, pp. 201-230, Apr 2009. Article (CrossRef Link)

[41] Hui Zou and Trevor Hastie, “Regularization and variable selection via the elastic net,” *Journal of Royal Statistical Society*, Series B, vol. 67, no. 2, pp. 301-320, 2005. Article (CrossRef Link)

[42] C. Veaux, J. Yamagishi, and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," in *Proc. of IEEE International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques, held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pp. 1-4, 2013. Article (CrossRef Link)

[43] A.Varga and H. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun*., vol. 12, pp. 247–251, 1993. Article (CrossRef Link)

[44] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of machine learning research*., pp. 2121–2159, 2011. Article (CrossRef Link)

[45] C. Taal, R. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, Sep 2011. Article (CrossRef Link)

[46] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 749–752, 2001. Article (CrossRef Link)

[47] Cohen and B. Berdugo, "Speech enhancement for nonstationary noise environments," *Signal processing*, vol. 81, no. 11, pp. 2403–2418, 2001. Article (CrossRef Link)

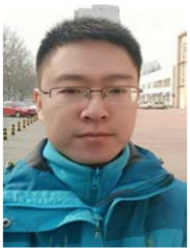
[48] R. Talmon and S. Gannot, "Single-channel transient interference suppression with diffusion maps," *IEEE Transactions on Audio, Speech, and Language Process*, vol. 21, no. 1, pp. 132–144, 2013. Article (CrossRef Link)

[49] Scott Pennock, "Accuracy of the perceptual evaluation of speech quality (pesq) algorithm," *Measurement of Speech & Audio Quality in Networks Line Workshop Mesaqin'*, vol. 25, 2002. Article (CrossRef Link)

[50] Zechao Li, and Jinhui Tang, "Weakly-supervised Deep Matrix Factorization for Social Image Understanding. IEEE Transactions On Image Processing," pp.1-13, 2016. Article (CrossRef Link)

[51] Zechao Li and Jinhui Tang, "Weakly Supervised Deep Metric Learning for Community-Contributed Image Retrieval," *IEEE Transactions On Multimedia*, pp.1989-1999, 2015. Article (CrossRef Link)

Si-Ying Xu received the B.S. degree in National Digital Switching System Engineering and Technological R&D Center, Zhengzhou, China in 2015. She is currently working towards the M.S. degree on speech recognition in National Digital Switching System Engineering and Technological R&D Center. Her research interests are in speech signal processing, speech enhancement, and machine learning.



Tong Niu received the master's degree in National Digital Switching System Engineering and Technological R&D Center, Zhengzhou, China in 2009. He did his B. Tech degree in communication engineering in National Digital Switching System Engineering and Technological R&D Center in 2006. His research interests includes speech enhancement and speech recognition.

Dan Qu received the M.S. degree in communication and information system from Xi’an Information Science and Technology Institute, Xi’an, China in 2000 and the Ph.D. degree in in National Digital Switching System Engineering and Technological R&D Center, Zhengzhou, China in 2005. She is an Associate Professor at the in National Digital Switching System Engineering and Technological R&D Center. Her research interests are in speech signal processing and pattern recognition.

Xing-Yan Long received the B.S. degree in automation from the Tsinghua University, Beijing in 2015. He is currently working towards the M.S. degree on speech recognition in National Digital Switching System Engineering and Technological R&D Center, Zhengzhou, China. His research interests are in speech signal processing, end-to-end speech recognition, and machine learning.