INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

An Investigation on the Use of i-vectors for Robust ASR

Dimitrios Dimitriadis[1], *Samuel Thomas*[1] *and Sriram Ganapathy*[2]

[1]IBM T.J. Watson Research Center, Yorktown Heights, USA,
[2]Department of EE Indian Institute of Science, Bangalore, India

[dbdimitr, sthomas]@us.ibm.com, sriram@ee.iisc.ernet.in

Abstract

In this paper we propose two different i-vector representations that improve the noise robustness of automatic speech recognition (ASR). The first kind of i-vectors is derived from "noise only" components of speech provided by an adaptive denoising algorithm, the second variant is extracted from mel filterbank energies containing both speech and noise. The effectiveness of both these representations is shown by combining them with two different kinds of spectral features - the commonly used log-mel filterbank energies and Teager energy spectral coefficients (TESCs). Using two different DNN architectures for acoustic modeling - a standard state-of-the-art sigmoid-based DNN and an advanced architecture using leaky ReLUs, dropout and rescaling, we demonstrate the benefit of the proposed representations. On the Aurora-4 multi-condition training task the proposed front-end improves ASR performance by 4%.

Index Terms: speech recognition, noise robustness, feature extraction, i-vectors

1. Introduction

Despite recent significant advances in acoustic modeling using deep neural networks (DNNs), automatic speech recognition systems (ASR) are still not robust enough to deal with noise, speaker and domain variabilities unseen during training. To improve speech recognition performances in such settings, four directions are actively pursued within the DNN acoustic models framework - feature compensation or signal enhancement, feature or model space adaptation, data augmentation followed by multi-condition style training and training with side information about the undesired variabilities in the signal.

Under the first class of techniques, DNNs are trained using noise robust feature representations compensated for additive and convolutive distortions [1, 2, 3, 4]. The second class of techniques identifies a subset of feature and/or model parameters, which can be adapted to the target speakers and channel characteristics [5, 6, 7, 8]. The third class of noise robustness strategies is the multi-condition style training of the neural networks, after data augmentation with real and artificially generated noises [9] providing significant performance gains, whilst increasing the network training complexities [10]. This approach is ofter either combined with noise robust feature representations described earlier or can be used to train networks directly on acoustic representations learning invariant transformations.

Finally, appending information about undesired noise and speaker variabilities provided by additional features can be considered as the fourth class of techniques. This additional information allows the network to automatically learn compensation transformations during training. One of the early approaches in this direction was the Noise Aware Training (NAT) [11], where noise estimates were concatenated with the acoustic features for improved robustness. Recently, speaker adaptation with speaker codes [12] and i-vectors [13, 14] have been successfully deployed for ASR.

In our earlier work, we demonstrated the usefulness of the i-vector approach for ASR in addressing channel and noise related variabilities in addition to the speaker variability [15], especially in mismatched training and testing conditions. The i-vector extractors in our case are trained at utterance level without any explicit speaker information in training. In this paper we investigate additional aspects of using the i-vectors to capture information about the noise for robust ASR systems. These include: (a) a study on training i-vectors on feature representations from "noise only" signals and also from noisy speech signals containing both speech and noise, (b) an evaluation of the effectiveness of these i-vectors variants in characterizing various noise types, and (c) an assessment of the usefulness of the proposed i-vector representations with various acoustic features and modeling techniques for neural networks. This paper focuses on the "matched" conditions scenario, where noisy training data is used. As mentioned before, this is a more challenging case for further improving the ASR performance, since the multi-condition training has already provided large gains.

ASR experiments in this paper are performed on the Aurora 4 task [16]- a medium vocabulary task, based on the Wall Street Journal corpus. Using the multi-condition experimental framework of this task which utilizes a variety of noise types for train and several test sets containing both seen and unseen noise distortions, we investigate the usefulness of our proposed i-vector representations. Instead of using an adaptive "Minimum-Mean-Square-Error" (MMSE) denoising algorithm to denoise signal, in Sec. 2 we describe how we use it to extract the "noise only" components. Then, the Teager Energy Spectral Coefficients (TESC) are described, in conjunction with the proposed i-vector representations. Sec. 4 describes a factor analysis framework for extracting i-vector representations both from the "noise only" signals and from the noisy speech signals containing both speech and noise. Finally, the experimental results are presented in Sec. 5 followed by a brief discussion. The paper concludes with a summary of the proposed techniques in Sec. 6.

2. Noise Signal Estimation

The Aurora-4 task has 4 different training/testing scenarios, i.e. ranging from matched to heavily mismatched noise conditions.

The work was performed when S. Ganapathy was with IBM Research.

These conditions include additive and channel noise of various types and levels. To compensate for these conditions, we have employed a variation of the MMSE algorithm [17], extracting information about the noise corrupting signals. The denoised signals are not used in the ASR processing pipeline since initial experimental results have shown little or no improvement in terms of ASR performance. Instead, we are using the noise residual signals for extracting i-vectors, similarly to the NAT approach [11].

In most cases, the MMSE denoiser is used to suppress the noise component. Herein, the denoised signals are subtracted from the original audio estimating a residual signal approximating the noise corrupting component. The key factor is that the denoising algorithm has to adapt fast enough to the speech fluctuations minimizing the speech signal leakage to the residual. Thus, we chose a modified version of the MMSE denoiser [18], capable of doing so.

The general idea of an MMSE-based denoiser is that the speech component of noisy audio is obtained by multiplying the noisy power spectrum by a gain

$$\widehat{A^2} = G_{A^2}(\xi, \zeta) \cdot R^2$$

where the gain G_{A^2} depends on the assumed speech and noise models [17], A and R are the denoised and noisy speech spectral amplitudes, and ξ and ζ the priori and posteriori SNR estimates, respectively. However, this process suffers from leakage of the speech power to the noise estimates. In order to minimize it, a time- and frequency-dependent smoothing parameter is proposed in [17], where the estimate of speech presence probability is also investigated. Further, the gain function is trained by an iterative data-driven training method [19] and a look-up table is created based on the speech and noise variance estimates. A safety net is also employed for the cases when the noise levels suddenly increase, as described in [17]. That algorithm provides a fast, adaptive estimate of the speech signals minimizing their leakage to the residual, as shown in [17, 19].

3. Feature Extraction

It is shown that the human hearing physiology [20, 21, 22] can be well modeled by the auditory filters, with bandwidths provided by the $ERB(f)$ curve,

$$ERB(f) = 6.23(f/1000)^2 + 93.39(f/1000) + 28.52$$

where f is the filter center frequency dictated by the Bark frequency scale. As that, the filter placing and bandwidth for the proposed filterbank are described by this curve [23]. Contrary to the typical logmel coefficients estimated over a filterbank of triangular filters with 50% overlap [24], we propose using the auditory-inspired filterbank and incorporate information about the time-varying nature of speech using the instantaneous Teager-Kaiser (TK) energy [25]. The auditory filters are approximated by the Gammatone filters and they are smoother and broader than the triangular filters. The proposed features are shown to be more robust in additive noise and provide additional acoustic information when compared to the logmels.

The *TESC* estimation algorithm is described with the following steps: (i) use a Gammatone filterbank to estimate a sequence of bandpass, speech signals. The number of filters is ranging from 25 to 200 filters, (ii) estimate the mean TK-energy for each one of the framed bandpass signals, (iii) estimate the Spectral coefficients as the log mean energies. The first two steps combine the auditory filtering scheme with a more "natural" approach of the speech TK-energy notion. These steps differentiate the proposed algorithm from the typical logmel extraction algorithm. The ASR results show significant improvement, especially in noisy recognition tasks [25].

4. Factor Analysis Framework

The techniques outlined here are derived from the previous work on joint factor analysis (JFA) and i-vectors [26, 27, 28]. We follow the notations used in [26]. The training data from all the speakers is used to train a GMM with model parameters $\lambda = \{\pi_c, \boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}\}$ where π_c, $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ denote the mixture component weights, mean vectors and covariance matrices respectively for $c = 1, .., C$ mixture components. Here, $\boldsymbol{\mu}_c$ is a vector of dimension F and $\boldsymbol{\Sigma}_c$ is of assumed to be diagonal matrix of dimension $F \times F$.

4.1. I-vector Representations

Let $\boldsymbol{\mathcal{M}_0}$ denote the UBM supervector which is the concatenation of $\boldsymbol{\mu}_c$ for $c = 1, .., C$ and is of dimension of $CF \times 1$. Let $\pm$ denote the block diagonal matrix of size $CF \times CF$ whose diagonal blocks are $\boldsymbol{\Sigma}_c$. Let $\mathcal{X}(s) = \{x_i^s, i = 1, ..., H(s)\}$ denote the low-level feature sequence for input recording s where i denotes the frame index. Here $H(s)$ denotes the number of frames in the recording. Each x_i^s is of dimension $F \times 1$.

Let $\boldsymbol{\mathcal{M}(s)}$ denote the recording supervector which is the concatenation of speaker adapted GMM means $\boldsymbol{\mu}_c(s)$ for $c = 1, .., C$ for the speaker s. Then, the i-vector model is,

$$\boldsymbol{\mathcal{M}}(s) = \boldsymbol{\mathcal{M}_0} + \boldsymbol{V}\boldsymbol{y}(s) \tag{1}$$

where $\boldsymbol{V}$ denotes the total variability matrix of dimension $CF \times M$ and $\boldsymbol{y}(s)$ denotes the i-vector of dimension M. The i-vector is assumed to be distributed as $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

In order to estimate the i-vectors, the iterative EM algorithm is used. We begin with random initialization for the total variability matrix $\boldsymbol{V}$. Let $p_\lambda(c|\boldsymbol{x}_i^s)$ denote the alignment probability of assigning the feature vector $\boldsymbol{x}_i^s$ to mixture component c. The sufficient statistics are then computed as,

$$\begin{aligned} N_c(s) &= \sum_{i=1}^{H(s)} p_\lambda(c|\boldsymbol{x}_i^s) \\ \boldsymbol{S}_{X,c}(s) &= \sum_{i=1}^{H(s)} p_\lambda(c|\boldsymbol{x}_i^s)(\boldsymbol{x}_i^s - \boldsymbol{\mu}_c) \end{aligned} \tag{2}$$

Let $\boldsymbol{N(s)}$ denote the $CF \times CF$ block diagonal matrix with diagonal blocks $N_1(s)\boldsymbol{I}$, $N_2(s)\boldsymbol{I}$,..,$N_C(s)\boldsymbol{I}$ where $\boldsymbol{I}$ is the $F \times F$ identity matrix. Let $\boldsymbol{S}_X(s)$ denote the $CF \times 1$ vector obtained by splicing $\boldsymbol{S}_{X,1}(s)$,..,$\boldsymbol{S}_{X,C}(s)$.

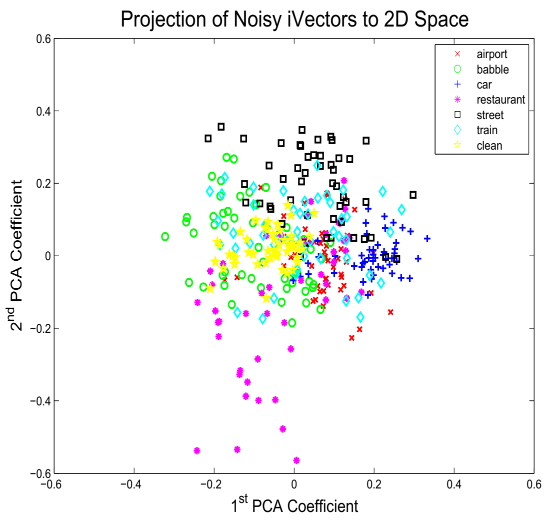
It can be easily shown [26] that the posterior distribution of the i-vector $p_\lambda(\boldsymbol{y}(s)|\mathcal{X}(s))$ is Gaussian with covariance $\boldsymbol{l}^{-1}(s)$ and mean $\boldsymbol{l}^{-1}(s)\boldsymbol{V}^*\pm^{-1}\boldsymbol{S}_X(s)$, where

$$\boldsymbol{l}(s) = \boldsymbol{I} + \boldsymbol{V}^*\pm^{-1}\boldsymbol{N}(s)\boldsymbol{V} \tag{3}$$

The optimal estimate for the i-vector $\boldsymbol{y}(s)$ obtained as $argmax_y\big[p_\lambda(\boldsymbol{y}(s)|\mathcal{X}(s))\big]$ is given by the mean of the posterior distribution.

For re-estimating the $\boldsymbol{V}$ matrix, the maximization of the expected value of the log-likelihood function (EM algorithm), gives the following relation [26],

$$\sum_{s=1}^{S} \boldsymbol{N}(s)\ \boldsymbol{V}\ \mathrm{E}\big[\boldsymbol{y}(s)\boldsymbol{y}^*(s)\big] = \sum_{s=1}^{S} \boldsymbol{S}_X(s)\mathrm{E}\big[\boldsymbol{y}^*(s)\big] \tag{4}$$



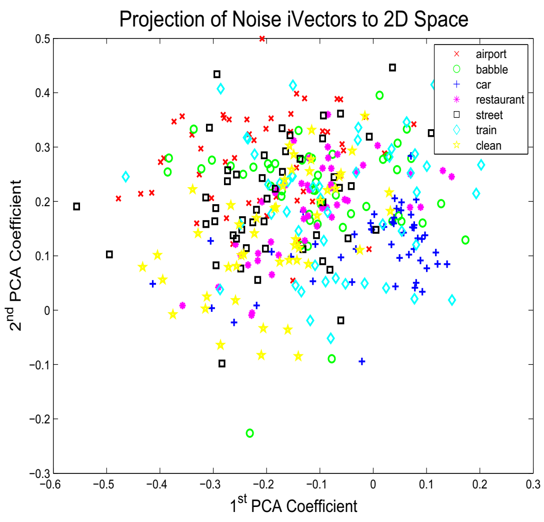


Figure 1: 2D Projection of the 25D i-vector Space. PCA Projection is trained on Aurora-4 Train Set. (left) Projection of Noisy i-vectors, (right) Projection of Noise i-vectors.

where E[.] denotes the posterior expectation operator. The solution for Eq. (4) can be computed for each row of $\boldsymbol{V}$. Thus, the i-vector estimation is performed by iterating between the estimation of posterior distribution and the update of the total variability matrix (Eq. (4)).

4.2. Noise i-vector Estimation

The MMSE-based denoising algorithm described in Sec. 2 is used to separate the "noise" components from the "clean" speech power spectrum. The noise power spectral components derived from training recordings of Aurora-4 are used as features to train a noise UBM. The zeroth and first order statistics of this UBM are derived from the noise features according to Eq. 2. These statistics are used to derive 25 dimensional i-vectors. We refer to these i-vectors as *noise i-vectors* as these i-vectors contain information purely from the noise component of the noisy speech signal.

4.3. Noisy i-vector Estimation

In a manner similar to the noise i-vector estimation, we also estimate i-vectors directly from the noisy speech signal (without denoising). These i-vectors contain the information about the broad interaction between the speech and noise signal. We refer to these i-vectors as *noisy i-vectors*.

We applied the PCA training only on the noisy training data (no clean speech) using only the "wv1" instances. The training set was standardized before estimating the PCA loadings. Then, we kept the first two principal components (those with the largest variance). During the training process, the clean speech and the "wv2" channel noises remain unseen. In Fig. 1, we plot the first two principal components of the noise and noisy i-vectors. The noisy i-vectors derived from the noisy speech have more structured information corresponding to the various types of noise corrupting the input data. This is reflected in our experiments, where the noisy i-vectors also contribute to higher gains in the ASR performance. In Fig. 2, we show the PCA projection for 2 types of additive noise, i.e. "restaurant" and "street" noise, under different channel conditions, i.e. "wv1" vs. "wv2". In this figure, the "noisy" i-vectors from similar noise types appear to cluster together, despite the channel mismatch, thus demonstrating a level of channel invariance. This is particularly useful for the DNN since it learns the additive noise conditions independent of any channel mismatches.

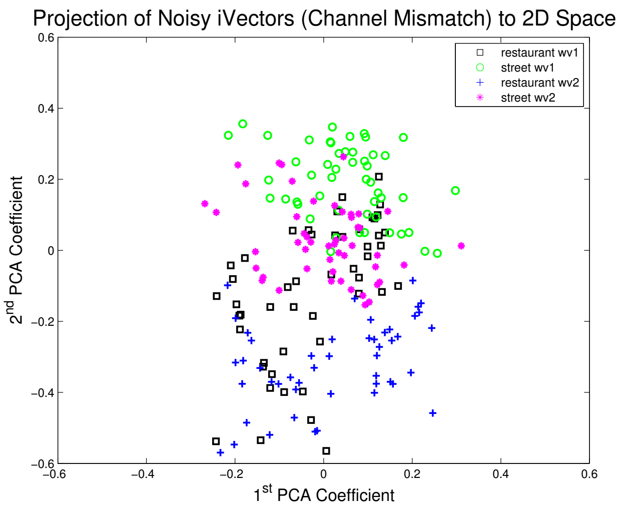


Figure 2: 2D Projection of the 25D i-vector Space: i-vectors of Restaurant and Street Noise Under 2 Channel Conditions (wv1 vs. wv2).

5. Experiments

The proposed techniques are evaluated on the Aurora 4 - a medium vocabulary task, based on the Wall Street Journal corpus [16]. We use the IBM Recognizer, Attila [29]. The AM DNNs are trained on the task's multi-condition training set with 7137 utterances sampled at 16kHz from 83 speakers, and then tested on a set of 330 utterances from 8 speakers. Half of the training utterances is recorded with a primary Sennheiser microphone, while the second half is collected using one of 18 other secondary microphones. The noisy utterances are corrupted with one of six different noise types (airport, babble, car, restaurant, street traffic and train station) at 10-20 db SNR.

Similarly to the train set, the test sets are also recorded over multiple microphones - a primary microphone and a secondary microphone. In addition to the clean test data collected over each of these microphones, the same six noise types used in train are employed to create noisy test sets at 5-15dB SNR, resulting in a total of 14 test sets. These test sets are commonly grouped into 4 subsets - clean (test set A), noisy (test set B), clean with channel distortion (test set C) and noisy with channel distortion (test set D).

An initial set of HMM-GMM models are trained to produce alignments for the multi-condition training utterances. Unlike the baseline systems, these models are built on the corresponding clean training (7137 utterances) set of the Aurora 4 task in a speaker dependent fashion. Starting with 39-dimentional VTL-warped PLP features and speaker based cepstral mean/variance normalization, an Maximum Likelihood system with fMLLR based speaker adaptation and 2000 context-dependent HMM states is trained. The alignments produced by this system, are further refined using a DNN system also trained on the clean training set with fMLLR based features.

Two different DNN architectures using sigmoid and leaky ReLU (*LReLU*) non-linearities are examined. In contrast to the ReLUs, in which the negative part is totally dropped, LReLUs assign a non-zero slope to it. The leaky rectifier allows for a small, non-zero gradient when the unit is saturated and not active [30]

$$h^{(i)} = \max(w^{(i)T}x, 0) = \begin{cases} w^{(i)T}x, & w^{(i)T}x > 0 \\ 0.1w^{(i)T}x, & \text{else} \end{cases} \quad (5)$$

All the systems are trained on 40 dimensional *logmel* and *TESC* spectra augmented with Δ and $\Delta\Delta$s. Each frame of speech is also appended with a context of 11 frames after applying a speaker independent global mean and variance normalization.

The DNN systems estimate posterior probabilities of 2000 targets using a network with either 6 or 7 hidden layers, each having 1024/2048 units per layer. For the DNN systems using LReLUs, a fixed dropout of 50% is applied only on the third and fourth hidden layers, only when the pre-training of the networks is finished. Similarly, we have also applied a fixed dropout rate of 20% to the input features [31]. Finally, rescaling of the weights is performed after every mini-batch iteration. All DNNs are discriminatively pre-trained before being fully trained to convergence. After training, the DNN models are decoded with the task-standard WSJ0 bigram Language Model.

We first investigated the optimal DNN architecture for the two different nonlinearities. We experimentally verified that the LReLU-based DNN generalize better, due to sparser activations, requiring a smaller number of hidden nodes and layers. In addition to that, the use of dropouts reducing the overfitting to the data [31]. The 6 hidden layer LReLU DNN (6×1024) provides a 6% rel. better performance in terms of average WER compared to the corresponding 7×2048 DNN. On the other hand, the sigmoid-based DNN needs more layers in order to generalize. The best ASR results for this baseline architecture are obtained when using 7×2048 layers, outperforming the 6×1024 architecture by 10% relative. All subsequently experiments with "sigmoids" have 7×2048 layers and the "advanced" DNNs have 6 hidden layers with 1024 nodes each.

The following observations can be drawn from Tables 1 and 2: (a) in both experiments the recognition systems trained on two different kinds of acoustic features benefit from utterance level side information available in the i-vectors, (b) the gains from the i-vector systems are much less pronounced in the LReLU based systems than in the sigmoid based system. We hypothesize this could because of the inherent robustness of the system coming from the nonlinearity being used, and (c) the noisy i-vectors in general provide more gains than the noise i-vectors in line with earlier visual observations. However both representations contain complimentary information as most gains are observed when they used in combination.

Table 1: DNN architecture is 7×2048 with Sigmoids. The noise and/or noisy i-vectors are concatenated to the $11 \times 3 \times 40$ noisy logmel or TESC features.

Multi-condition Training: Sigmoids					
	A	B	C	D	Aver.
logmel	5.85	10.33	11.13	22.64	15.34
TESC	6.07	10.24	11.34	21.82	14.98
logmel+Noise i-vectors	5.34	9.81	11.54	21.13	14.47
logmel+Noisy i-vectors	5.51	10.43	11.41	22.44	15.29
logmel+ Noise+Noisy i-vectors	5.38	9.79	11.58	21.74	14.72
TESC+Noise i-vectors	6.00	10.43	11.86	22.38	15.34
TESC+Noisy i-vectors	5.70	9.70	12.27	21.38	14.61
TESC+ Noise+Noisy i-vectors	5.62	9.83	12.54	21.22	14.60

Table 2: DNN architecture is 6×1024 with LReLUs. The noise and/or noisy i-vectors are concatenated to the $11 \times 3 \times 40$ noisy logmel or TESC features.

Multi-condition Training: ReLU					
	A	B	C	D	Aver.
logmel	4.13	7.46	7.34	16.19	10.96
TESC	4.46	7.58	7.90	16.20	11.07
logmel+Noise i-vectors	4.28	7.32	7.75	16.78	11.19
logmel+Noisy i-vectors	4.35	7.35	8.14	16.73	11.21
logmel+ Noise+Noisy i-vectors	4.00	7.17	7.70	16.41	10.94
TESC+Noise i-vectors	4.24	7.37	7.62	16.26	10.98
TESC+Noisy i-vectors	4.75	7.11	7.81	15.41	10.55
TESC+ Noise+Noisy i-vectors	4.26	7.12	7.64	15.44	10.51

6. Conclusions

One of the earlier conclusions in robust ASR is that noise suppression is hardly helpful, especially when multi-condition training is involved. However, we herein propose using the noise suppression approach indirectly for estimating only the noise signal residuals. Then, we estimate i-vectors based on these residuals, providing information about the noise conditions. The proposed algorithm can be compared with the NAT coefficients [11], but the i-vectors are now estimated over the entire signal, instead of the first (and last) few frames. The experimental results in Tables 1 and 2 show that incorporating such information about noise is helpful in most of the scenarios. These improvements are consistent with the noise invariance of the i-vectors (especially in the case of channel noise) shown in Figs. 1 and 2. The proposed system is comparable with previously published systems [11], outperforming them by more than 15% (relative).

7. References

[1] O. Kalinli, N. L. Seltzer, and A. Acero, "Noise adaptive training using a vector taylor series approach for noise robust automatic speech recognition," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3825–3828.

[2] V. Mitra, H. Franco, M. Graciarena, and A. Mandal, "Normalized amplitude modulation features for large vocabulary noise-robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4117–4120.

[3] S. Ganapathy, S. Thomas, and H. Hermansky, "Robust spectro-temporal features based on autoregressive models of hilbert envelopes," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4286–4289.

[4] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (pncc) for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4101–4104.

[5] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 366–369.

[6] A. Narayanan and D. Wang, "Joint noise adaptive training for robust automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2504–2508.

[7] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 24–29.

[8] S. J. Rennie, V. Goel, and S. Thomas, "Annealed dropout training of deep networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 159–164.

[9] R. Hsiao, J. Ma, W. Hartmann, M. Karafiat, F. Grézl, L. Burget, I. Szoke, J. Cernocky, S. Watanabe, Z. Chen *et al.*, "Robust speech recognition in unknown reverberant and noisy conditions," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 2015.

[10] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proceedings of INTERSPEECH*, 2015.

[11] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *ICASSP*, 2013.

[12] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7942–7946.

[13] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors." in *ASRU*, 2013, pp. 55–59.

[14] A. W. Senior and I. Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs." in *ICASSP*, 2014, pp. 225–229.

[15] S. Ganapathy, S. Thomas, D. Dimitriadis, and S. Rennie, "Investigating factor analysis features for deep neural networks in noisy speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] N. Parihar and J. Picone, "Aurora Working Group: DSP Front-end and LVCSR Evaluation AU/384/02," Inst. for Signal and Information Processing, Mississippi State University, Tech. Rep., 2002.

[17] J. S. Erkelens and R. Heusdens, "Tracking of nonstationary noise based on data-driven recursive noise power estimation," *IEEE Trans. on Audio, Speech and Language Process.*, vol. 16, no. 6, pp. 1112–1123, Aug. 2008.

[18] Y. Ephraim and D. Malah, "Speech enhancement using a minimum min-square error log-spectral amplitude estimator," *IEEE Trans. on Acoust., Speech and Signal Process.*, vol. 33, no. 2, pp. 443–445, 1985.

[19] J. S. Erkelens, J. Jensen, and R. Heusdens, "A data-driven approach to optimizing spectral speech enhancement methods for various error criteria," *Speech Communication*, vol. 49, pp. 530–541, Aug. 2007.

[20] T. Irino and R. D. Patterson, "A Time-Domain, Level-Dependent Auditory Filter: The Gammachirp," *Journ. Acoustical Society of America*, 1997.

[21] O. Ghitza, "Auditory Models and Human Performance in Tasks Related to Speech Coding and Speech Recognition," *IEEE Trans. Speech and Audio Processing*, 1994.

[22] B. R. Glasberg and B. C. J. Moore, "Derivation of Auditory Filter Shapes from Notched-Noise Data," *"Hear. Res."*, 1990.

[23] D. Dimitriadis, P. Maragos, and A. Potamianos, "On the effects of filterbank design and energy computation on robust speech recognition," *IEEE Trans. on Audio, Speech and Language Process.*, vol. 19, no. 6, pp. 1504–1516, Aug. 2011.

[24] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, 1980.

[25] D. Dimitriadis, P. Maragos, and A. Potamianos, "Auditory Teager Energy Cepstrum Coefficients for Robust Speech Recognition," in *Eurospeech*, 2005.

[26] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 345–354, 2005.

[27] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, 2005.

[28] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[29] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila Speech Recognition Toolkit," in *IEEE SLT*, 2010.

[30] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML Workshop on Deep Learning for Audio, Speech, and Language Processing (WDLASL)*, 2013.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.