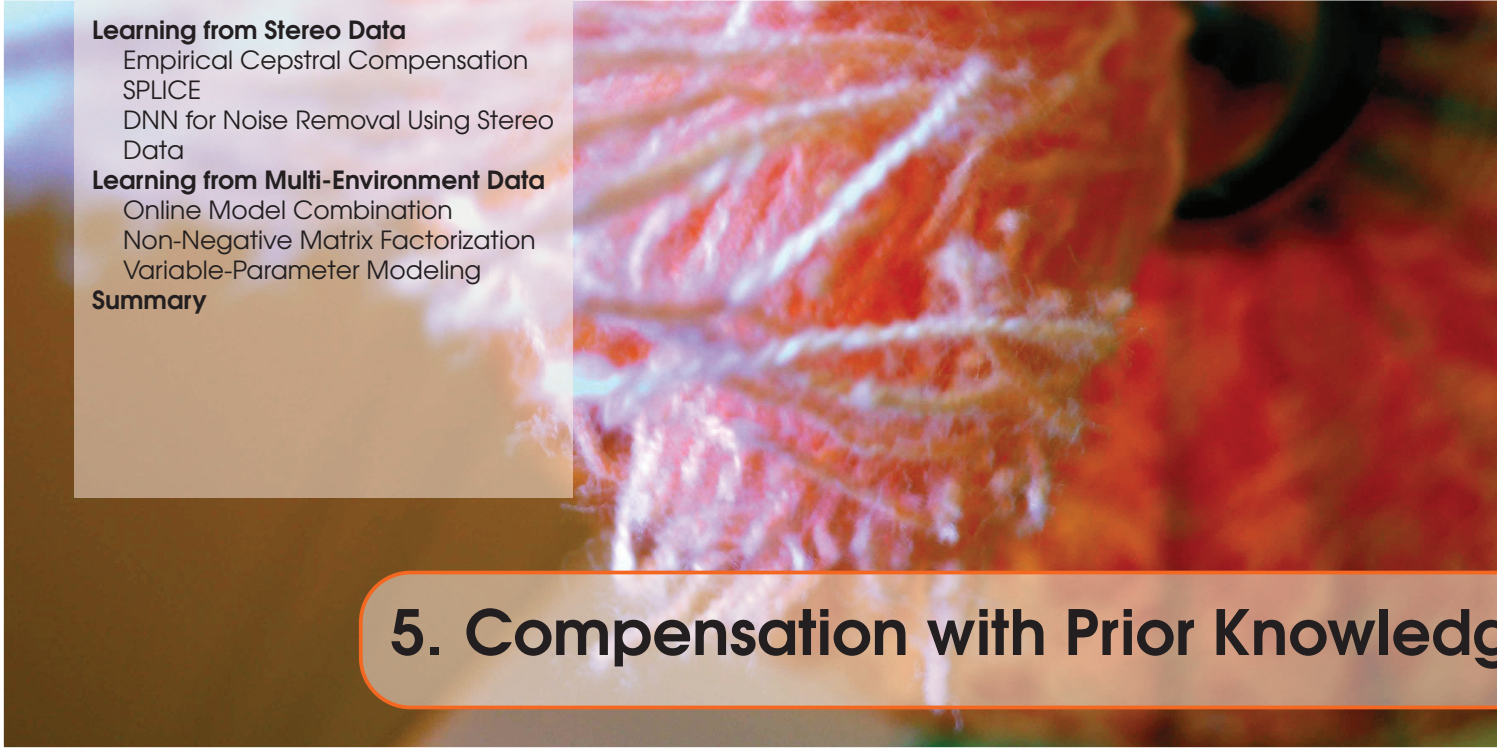

5. Compensation with Prior Knowledg

NON-PRINT ITEMS

Abstract

All methods analyzed and contrasted in this chapter have the unique attribute of exploiting prior knowledge about distortion in the training stage, in addition to training an HMM. They then use such prior knowledge as a guide to either remove noise or adapt models in the testing or deployment stage. Most methods which use prior knowledge about acoustic distortions as discussed in this chapter learn the nonlinear mapping functions between the clean and noisy speech features when they are available in the training phase as a pair of stereo data. By modeling the differences between the features or models of the stereo data, a distortion model can be learned accuratly in training and subsequently used in testing to perform feature enhancement or model compensation. Another set of methods that also exploit prior knowledge operate by collecting and learning a set of simple models first, each corresponding to one specific acoustic environment in the training. These environment-specific models are then combined in the online fashion to form a new acoustic model that is aimed to fit the test environment in an optimal matter.

Key Words

prior knowledge, stereo data, environment-specific, online model combination, variable-parameter modeling, DNN-based noise removal, SPLICE, non-negative matrix factorization

In this chapter, we explore an alternative way of categorizing and analyzing existing robust ASR techniques, where we use the attribute of whether or not they make use of prior knowledge and information about the acoustic distortion before applying formal compensation procedures. This contrasts the previous chapter when the attribute was whether the operations were applied on the feature domain or on the model domain.

Major noise-robust methods which use the prior knowledge about acoustic distortions learn the generally nonlinear mapping functions between the clean and distorted speech features when they are available in the form of stereo data in the training phase. By modeling the differences between the features or models of the stereo data, a distortion model can be learned in training and then used in testing to perform feature enhancement or model compensation. The distortion model can be a deterministic mapping function. It can also be formulated probabilistically as in $p(\mathbf{y}|\mathbf{x})$. A collection of these methods can be called stereo-data mapping methods.

In addition to stereo-based methods, another collection of methods exploiting prior knowledge are based on first establishing or sampling a set of simple models for the acoustic environments, each corresponding to one specific environment during training. These models are then combined online to form the final acoustic model of distorted speech that fits the test environment to the best extent possible.

More recently, there appeared in the literature new methods based on clean speech and noise exemplar dictionaries learned from training data for source separation. Using non-negative matrix factorization (NMF), these methods restore clean speech by constructing the noisy speech with pre-trained clean speech and noise exemplars and only keeping the clean speech exemplars. How to generalize to unseen acoustic conditions is very important to robust ASR. Variable-parameter modeling presented in this chapter will provide a decent solution by modeling the acoustic model parameters with a set of polynomial functions of the environment variable. The model parameters can be extrapolated from the learned polynomial functions if the test environments are not observed during training.

5.1 Learning from Stereo Data

Many methods use stereo data to learn the mapping from distorted speech to clean speech. The stereo data consists of time-aligned speech samples that have been simultaneously recorded in training environments and in representative test environments. Stereo data can also be obtained by digitally introducing (e.g. adding noise) distortion to the clean speech. The success of these methods usually depends on how well the representative

distorted samples during training really match test samples.

5.1.1 Empirical Cepstral Compensation

One group of methods is called empirical cepstral compensation [Stern et al., 1996], developed at CMU. Let's recap Eq-3.14 in Eq-5.1 which is the cepstral representation of the relationship between the clean speech feature and the distorted speech feature as

$$\mathbf{y} = \mathbf{x} + \mathbf{h} + \mathbf{C}\log(1 + \exp(\mathbf{C}^{-1}(\mathbf{n} - \mathbf{x} - \mathbf{h}))). \tag{5.1}$$

Then, with

$$\mathbf{v} = \mathbf{h} + \mathbf{C}\log(1 + \exp(\mathbf{C}^{-1}(\mathbf{n} - \mathbf{x} - \mathbf{h}))). \tag{5.2}$$

the distorted speech cepstrum $\mathbf{y}$ is expressed as the clean speech cepstrum $\mathbf{x}$ plus a bias $\mathbf{v}$. In empirical cepstral compensation, this bias $\mathbf{v}$ can be formulated to depend on the SNR, the location of vector quantization (VQ) cluster k, the presumed phoneme identity p, and the specific test environment e. Hence, Eq-3.14 can be re-written as

$$\mathbf{y} = \mathbf{x} + \mathbf{v}(SNR, k, p, e). \tag{5.3}$$

$\mathbf{v}(SNR, k, p, e)$ can be learned from stereo training data. During testing, the clean speech cepstrum can be recovered from the distorted speech with

$$\hat{\mathbf{x}} = \mathbf{y} - \mathbf{v}(SNR, k, p, e). \tag{5.4}$$

Depending on how $\mathbf{v}(SNR, k, p, e)$ is defined, there are different cepstral compensation methods. If SNR is the only factor for $\mathbf{v}$, it is called SNR-dependent cepstral normalization (SDCN) [Acero and Stern, 1990]. During training, frame pairs in the stereo data are allocated into different subsets according to SNR. Then, the compensation vector $\mathbf{v}(SNR)$ corresponding to a range of SNRs is estimated by averaging the difference between the cepstral vectors of the clean and distorted speech features for all frames in that range. During testing, the SNR for each frame of the input speech is first estimated, and the corresponding compensation vector is then applied to the cepstral vector for that frame with Eq-5.4.

Fixed codeword-dependent cepstral normalization (FCDCN) [Acero, 1993] is a refined version of SDCN with the compensation vector as $\mathbf{v}(SNR, k)$, which depends on both SNR and VQ cluster location. For each SNR range, there is a VQ cluster trained from the utterances representative for the testing. During training, the frame pairs in the

stereo data are allocated into different subsets according to the SNR and the VQ cluster location of the distorted feature. The compensation vector is calculated by averaging the difference between the cepstral vectors of the clean and distorted speech features for the SNR-specific VQ cluster location. During testing, both SNR and VQ cluster locations are estimated, and the corresponding compensation vector is then applied to the cepstral vector for that frame. Phone-dependent cepstral normalization (PDCN) [Liu et al., 1994] is another empirical cepstral compensation method in which the compensation vector depends on the presumed phoneme the current frame belongs to. During testing, the phoneme hypotheses can be obtained by a first pass HMM decoding. It can also be extended to include SNR as a factor, and is called SNR-dependent PDCN (SPDCN) [Liu et al., 1994]. Environment is also a factor of the compensation vector. FCDCN and PDCN can be extended to multiple FCDCN (MFCDCN) and multiple PDCN (MPDCN) when multiple environments are used in training [Liu et al., 2004]. The test utterance is first classified into one specific environment e, and then the compensation vector $\mathbf{v}(SNR,k,e)$ (in MFCDCN) or $\mathbf{v}(p,e)$ (in MPDCN) will be applied to the distorted speech cepstral vector. Another alternative is to interpolate the compensation vectors from those of multiple environments instead of making the hard decision of the specific environment. The corresponding methods are called interpolated FCDCN and interpolated PDCN [Liu et al., 1994].

5.1.2 SPLICE

Stereo-based Piecewise LInear Compensation for Environments (SPLICE) , proposed originally in [Deng et al., 2000a] and described in more detail in [Deng et al., 2001, Droppo et al., 2001b, 2002, Deng et al., 2003c], is a popular method to learn from stereo data and is more advanced than the aforementioned empirical cepstral compensation methods. In SPLICE, the noisy speech data, $\mathbf{y}$, is modeled by a mixture of Gaussians

$$p(\mathbf{y},k) = P(k)p(\mathbf{y}|k) = P(k)\mathcal{N}(\mathbf{y};\mu(k),\Sigma(k)), \tag{5.5}$$

and the *a posteriori* probability of clean speech vector $\mathbf{x}$ given the noisy speech $\mathbf{y}$ and the mixture component k is modeled using an additive correction vector $\mathbf{b}(k)$:

$$p(\mathbf{x}|\mathbf{y},k) = \mathcal{N}(\mathbf{x};\mathbf{y}+\mathbf{b}(k),\Psi(k)), \tag{5.6}$$

where $\Psi(k)$ is the covariance matrix of the mixture component dependent posterior distribution, representing the prediction error. The dependence of the additive (linear)

correction vector on the mixture component gives rise to a piecewise linear relationship between the noisy speech observation and the clean speech, hence the name of SPLICE. The feature compensation formulation can be described by

$$\hat{\mathbf{x}} = \sum_{k=1}^{K} P(k|\mathbf{y})(\mathbf{y} + \mathbf{b}(k)). \tag{5.7}$$

The prediction bias vector, $\mathbf{b}(k)$, is estimated by minimizing the mean square error (MMSE) as the weighted mean square error between the clean speech vector and the predicted clean speech vector in the mixture component k:

$$E = \sum_{t} P(k|\mathbf{y}_t)(\mathbf{x}_t - \mathbf{y}_t - \mathbf{b}(k))^2. \tag{5.8}$$

By setting $\frac{\partial E}{\partial \mathbf{b}(k)} = 0$, the estimation of the prediction bias vector, $\mathbf{b}(k)$, is obtained as

$$\mathbf{b}(k) = \frac{\sum_t P(k|\mathbf{y}_t)(\mathbf{x}_t - \mathbf{y}_t)}{\sum_t P(k|\mathbf{y}_t)}, \tag{5.9}$$

and $\Psi(k)$ can be obtained as

$$\Psi(k) = \frac{\sum_t P(k|\mathbf{y}_t)(\mathbf{x}_t - \mathbf{y}_t)(\mathbf{x}_t - \mathbf{y}_t)^T}{\sum_t P(k|\mathbf{y}_t)} - \mathbf{b}(k)\mathbf{b}^T(k). \tag{5.10}$$

To reduce the runtime cost, the following simplification can be used

$$\begin{aligned} \hat{k} &= \operatorname*{argmax}_k p(\mathbf{y}, k), \\ \hat{\mathbf{x}} &= \mathbf{y} + \mathbf{b}_{\hat{k}}. \end{aligned} \tag{5.11}$$

Note that for implementation simplicity, a fundamental assumption is made in the above SPLICE algorithm that the expected clean speech vector $\mathbf{x}$ is a shifted version of the noisy speech vector $\mathbf{y}$. In reality, when $\mathbf{x}$ and $\mathbf{y}$ are Gaussians given component k, their joint distribution can be modeled as

$$\mathcal{N}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} \mu_x(k) \\ \mu_y(k) \end{bmatrix}, \begin{bmatrix} \Sigma_x(k) & \Sigma_{xy}(k) \\ \Sigma_{yx}(k) & \Sigma_y(k) \end{bmatrix}\right). \tag{5.12}$$

and a rotation on $\mathbf{y}$ is needed for the conditional mean as

$$E(\mathbf{x}|\mathbf{y}, k) = \mu_x(k) + \Sigma_{xy}(k)\Sigma_y^{-1}(k)(\mathbf{y} - \mu_y(k)) \tag{5.13}$$

$$= \mathbf{A}(k)\mathbf{y} + \mathbf{b}(k), \tag{5.14}$$

where

$$\mathbf{A}(k) = \Sigma_{xy}(k)\Sigma_y^{-1}(k) \tag{5.15}$$

$$\mathbf{b}(k) = \mu_x(k) - \Sigma_{xy}(k)\Sigma_y^{-1}(k)\mu_y(k). \tag{5.16}$$

The feature compensation formulation in this case is

$$\hat{\mathbf{x}} = \sum_{k=1}^{K} P(k|\mathbf{y})(\mathbf{A}(k)\mathbf{y} + \mathbf{b}(k)). \tag{5.17}$$

It is interesting that feature space minimum phone error (fMPE) training [Povey et al., 2005a], a very popular feature space discriminative training method, can be linked to SPLICE to some extent [Deng et al., 2005b]. Originally derived with the MMSE criterion, SPLICE can be improved with the maximum mutual information criterion [Bahl et al., 1997] by discriminative training $\mathbf{A}(k)$ and $\mathbf{b}(k)$ [Droppo and Acero, 2005]. In [Droppo et al., 2001b], dynamic SPLICE is proposed to not only minimize the static deviation from the clean to noisy cepstral vectors, but to also minimize the deviation between the delta parameters. This is implemented by using a simple zero-phase, non-causal IIR filter to smooth the cepstral bias vectors.

In addition to SPLICE, MMSE-based stereo mapping is studied in [Cui et al., 2008a], and the MAP-based stereo mapping is formulated in [Afify et al., 2007, 2009]. Most stereo mapping methods use a GMM to construct a joint space of the clean and noisy speech feature. This is extended in [Cui et al., 2008b], where a HMM is used. The mapping methods can also be extended into a discriminatively trained feature space, such as the fMPE space [Cui et al., 2009a].

One concern for learning with stereo data is the requirement of stereo data, which may not be available in real-world application scenarios. In [Droppo et al., 2002], it is shown that a small amount of real noise synthetically mixed into a large, clean corpus is enough to achieve significant benefits for the FCDCN method. In [Du et al., 2010], the pseudo-clean features generated with a HMM-based synthesis method [Tokuda et al., 2000] are used to replace the clean features which are usually hard to get in real deployment. It is shown that this pseudo-clean feature is even more effective than the ideal clean feature [Du et al., 2010].

5.1.3 DNN for Noise Removal Using Stereo Data

Both the empirical cepstral compensation and SPLICE are piecewise linear compensation methods, in which the noisy feature $\mathbf{y}$ and the estimated clean feature $\hat{\mathbf{x}}$ have an

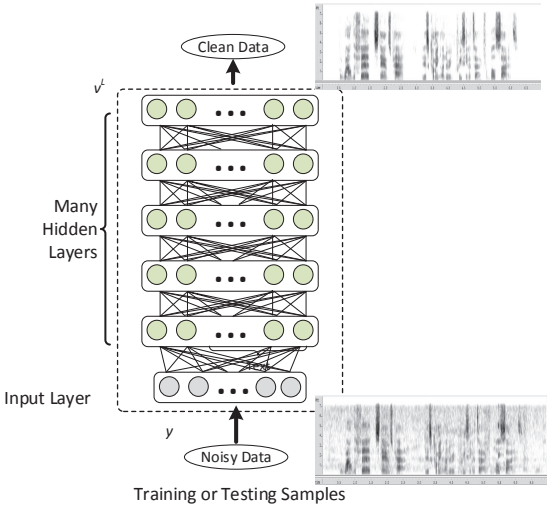

Figure 5.1: Generate clean feature from noisy feature with DNN

environment-dependent linear relationship. If putting them into the context of neural network with $\mathbf{y}$ as the input and $\hat{\mathbf{x}}$ as the output, all of these methods may be considered as a shallow neural network to learn the mapping of $\mathbf{y}$ and $\hat{\mathbf{x}}$ as $\hat{\mathbf{x}} = \mathscr{G}(\mathbf{y})$. From the success of DNNs, we learn that a deep neural network usually has more modeling power than a shallow neural network. Hence it is natural to use a DNN to better learn the mapping function $\mathscr{G}$, and this method has been recently very successful in both speech enhancement and speech recognition tasks [Maas et al., 2012b, Lu et al., 2013c, Wöllmer et al., 2013b, Narayanan and Wang, 2013a, Weninger et al., 2014a,c, Feng et al., 2014b, Du et al., 2014a,b, Narayanan and Wang, 2014b,a, Wang et al., 2014, Gao et al., 2015, Tu et al., 2015].

As shown in Figure 5.1, a DNN can be trained to generate clean feature from noisy feature $\mathbf{y}$ by minimizing the mean squared error between the DNN output $\hat{\mathbf{x}} = \mathscr{G}(\mathbf{y})$ and the reference clean features $\mathbf{x}$ [Lu et al., 2013c, Feng et al., 2014b, Du et al., 2014a,b]:

$$F_{MSE} = \sum_t \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 \tag{5.18}$$

Usually, the input noisy feature $\mathbf{y}$ is with a context window of consecutive frames, while the reference clean features $\mathbf{x}$ only corresponds to the current frame. The enhancement function $\mathscr{G}$ is realized with a DNN in Figure 5.1. This noise-removal strategy is very

effective. Evaluated in [Du et al., 2014b], when the underlying DNN model used for recognition is trained with clean data and the noisy test data is cleaned with the noise-removal DNN, huge WER reduction can be achieved. If the underlying DNN model used for recognition is trained with multi-condition data, remarkable WER reduction still can be achieved. Note that if the underlying model for recognition is a GMM, the improvement of using a noise-removal DNN is even larger. Also, the improvement is much larger than that obtained with AFE [ETSI, 2002] described in Section 4.1.3. This is due to the power of DNNs which learn the mapping between noisy and clean feature, while AFE is a traditional front-end without learning from the stereo data.

In addition to using a standard feed-forward DNN, a recurrent neural network (RNN) has also been proposed to predict the clean speech from noisy speech [Maas et al., 2012b] by modeling temporal signal dependencies in an explicit way because a RNN directly uses its time-recurrent structure to model the long-range context of speech which cannot be approximated by the feature stacking with a context window in the standard feed-forward DNN. Standard RNN has a known problem of weight decaying (or blowing up) during training. This issue can be solved by replacing the sigmoid units with long short-term memory (LSTM) units and bidirectional LSTM (BLSTM) units [Wöllmer et al., 2013b, Weninger et al., 2014a,c] which allow for a more efficient exploitation of temporal context, leading to an improved feature mapping from noisy speech to clean speech. The LSTM units have an internal memory cell whose content is modified in every time step by input, output, and forget gates so that the network memory is modeled explicitly.

While most studies [Maas et al., 2012b, Lu et al., 2013c, Wöllmer et al., 2013b, Weninger et al., 2014a, Feng et al., 2014b, Du et al., 2014a] use clean speech features as the DNN training target, there are also some works [Narayanan and Wang, 2013a, 2014b,a, Wang et al., 2014] using the time-frequency (T-F) masks such as ideal binary mask (IBM) or ideal ratio mask (IRM) as the training target. For each T-F unit, the corresponding IBM value is set to 1 if the local SNR is greater than a local criterion, otherwise it is set to 0. IRM is defined as the energy ratio of clean speech to noisy speech at each T-F unit with the assumption that noise is uncorrelated with clean speech, and can be written as a function of SNR:

$$IRM(t,k) = \left(\frac{SNR(t,k)}{1+SNR(t,k)} \right)^{\beta}, \tag{5.19}$$

where β is a tunable parameter to scale the mask. This is closely related to the frequency-domain Wiener filter in Eq-4.29. The training of IBM or IRM estimation with a DNN

is done by replacing the clean speech target in Figure 5.1 with either IBM or IRM as the target. During testing, the estimate of the T-F mask $\hat{\mathbf{m}}_t$ is obtained by forward propagating the learned DNN, and the estimated clean speech spectrum is obtained as

$$\hat{\mathbf{x}}_t = \hat{\mathbf{m}}_t .* \mathbf{y_t}, \tag{5.20}$$

where $.*$ is the element wise multiplication.

It is shown in [Narayanan and Wang, 2014a] that IRM is superior to IBM for the speech recognition task. However, it is still arguable whether using IRM is better than using clean speech feature as the target for noise removal. Suppose a clean utterance is corrupted by different types of noise with various SNRs, using clean speech feature as target directly maps the features from all the utterances with different distortion to the features from the same clean utterance. A DNN needs to learn this challenging many-to-one mapping. In contrast, by using IRM as the training target, the DNN learning is pretty simple – only the one-to-one mapping needs to be learned. Moreover, the target IRM value is between 0 and 1, which makes the learning avoid estimation of unbounded values. [Wang et al., 2014] also provides other arguments why IRM is better as the DNN training target for the task of speech separation. As a result, IRM as the training target is shown to outperform clean speech feature as the training target in speech separation tasks [Wang et al., 2014, Weninger et al., 2014c]. On the other hand, using IRM in Eq-5.19 as the training target is supposed to remove only the noise distortion. If the distorted signal $\mathbf{y}$ is also impacted by the channel distortion, an additional feature mapping function has to be provided in [Narayanan and Wang, 2014a] to remove the channel distortion in the estimated clean speech feature from the noise-removal DNN. In contrast, using clean speech feature as the training target can directly map the noise and channel distorted feature to clean speech feature with its many-to-one mapping in one step.

In addition to noise removal with DNN based on the minimum square error criterion, similar methodology can separate multiple speakers by putting the mixed feature as the input and the target speaker feature as the output in Figure 5.1. This is done in [Weng et al., 2014a]where separate DNNs are trained to predict individual sources. Another solution is proposed in [Huang et al., 2014a] where a single DNN is trained to predict all the sources as in Figure 5.2. This is optimized by minimizing the objective function

$$F_{MSE2} = \sum_t \|\hat{\mathbf{x}}_{1t} - \mathbf{x}_{1t}\|^2 + \|\hat{\mathbf{x}}_{2t} - \mathbf{x}_{2t}\|^2 \tag{5.21}$$

One improvement proposed in [Huang et al., 2014a] is to refine the final speaker sources with the constraint that they can be combined to form the original mixed feature

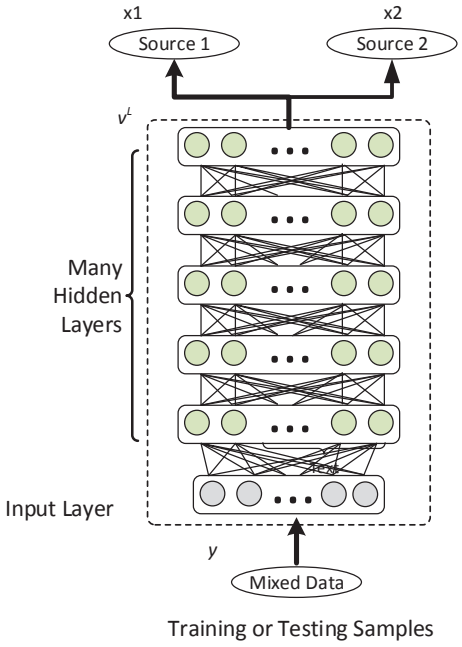

Figure 5.2: Speech separation with DNN

with

$$\tilde{\mathbf{x}}_{1t} = \frac{\|\hat{\mathbf{x}}_{1t}\|}{\|\hat{\mathbf{x}}_{1t}\| + \|\hat{\mathbf{x}}_{2t}\|} .* \mathbf{y}_t \tag{5.22}$$

$$\tilde{\mathbf{x}}_{2t} = \frac{\|\hat{\mathbf{x}}_{2t}\|}{\|\hat{\mathbf{x}}_{1t}\| + \|\hat{\mathbf{x}}_{2t}\|} .* \mathbf{y}_t \tag{5.23}$$

In this way, the reconstructed sources are more meaningful. The DNN optimization is still done with Eq-5.21 by replacing $\hat{\mathbf{x}}_{1t}$ and $\hat{\mathbf{x}}_{2t}$ with $\tilde{\mathbf{x}}_{1t}$ and $\tilde{\mathbf{x}}_{2t}$. This method should also be applicable to noise removal if we consider $\mathbf{x_1}$ and $\mathbf{x_2}$ are the clean speech and noise features, respectively.

Similar to the learning with SPLICE and empirical cepstral compensation, the supervised learning using DNN also needs stereo data which is hard to obtain in most real world scenarios. One solution is proposed in [Du et al., 2014a], where pseudo clean data is generated with HMM-based synthesis. When the test noise is also available during the learning of the enhancement function $\mathscr{G}$, high performance can always be

obtained by using DNN for noise removal. However, the challenge is the generalization to unseen conditions. This problem can be significantly alleviated by training the noise removal function on more acoustic conditions [Wang and Wang, 2013]. In [Xu et al., 2014], a set of more than 100 noise types is added when training the noise removal function in order to enrich the generalization of the DNN to unseen and non-stationary noise conditions. Although achieving satisfactory results in the area of speech separation, this noise enrichment method still degrades the recognition performance in the unseen test sets when the underlying acoustic model is a DNN [Du et al., 2014b].

5.2 Learning from Multi-Environment Data

This type of methods utilizes prior knowledge about the distortion by collecting and learning a set of models first, each corresponding to one specified environment in the training. These environment-specific models are then online combined to form a new model that fits the test environment best.

Usually, the acoustic model can be trained with a multi-condition training set to cover a wide range of application environments. However, there are two major problems with multi-style training. The first is that during training it is hard to enumerate all of the possible noise types and SNRs that may be present in future test environments. The second is that the distribution trained with multi-style training is too broad because it needs to model the data from all environments. Therefore, it is better to build environment-specific models, and use the model that best fits the test environment when doing runtime evaluation.

5.2.1 Online Model Combination

The model combination methods build a set of acoustic models, each modeling one specific environment. During testing all the models are combined to construct a target model used to recognize the current test utterance. Denote the set of environment-dependent parameters as $\{\Lambda_1, \ldots, \Lambda_K\}$, where K is the total number of environments. Then the model parameters during testing can be obtained as

$$\hat{\Lambda} = \sum_{k=1}^{K} w_k \Lambda_k, \tag{5.24}$$

where w_k is the combination weight for the k-th environment model. The model parameters can be Gaussian mean vectors or transforms when the underlying acoustic model is a GMM, and they can be weight matrices in the DNN case.

Online Model Combination for GMM

Assume that K environment-specific models share the same covariance matrix and only differ in mean parameters of GMMs. The mean parameters for each environment-specific model are concatenated together to form mean super-vectors ($\mathbf{s}_k, k = 1 \ldots K$), and the mean super-vector of the test utterance, $\hat{\mathbf{s}}$, is obtained as a linear combination of K mean super-vectors of the environment-specific models

$$\hat{\mathbf{s}} = \sum_{k=1}^{K} w_k \mathbf{s}_k, \tag{5.25}$$

where w_k is the combination weight for the k-th mean super-vector, and $\mathbf{w} = [w_1, w_2, \ldots, w_K]^T$. The combination weights $\mathbf{w}$ can be obtained with the maximum likelihood estimation (MLE) criterion as

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \log p(\mathbf{Y}|\hat{\mathbf{s}}) \tag{5.26}$$

This is solved with the expectation-maximization (EM) algorithm which finds the solution of $\mathbf{w}$ iteratively. The auxiliary function is defined as the following by ignoring standard constants and terms independent of $\mathbf{w}$

$$Q(\mathbf{w}; \mathbf{w}_0) = -\frac{1}{2} \sum_{m,t} \gamma_t(m) (\mathbf{y}_t - \mu(m))^T \Sigma^{-1}(m) (\mathbf{y}_t - \mu(m)), \tag{5.27}$$

where $\mathbf{w}_0$ is the previous weight estimate, $\gamma_t(m)$ is the posterior of Gaussian component m at time t determined using the previous model parameters, and $\mathbf{y}_t$ is the feature vector of frame t. $\mu(m)$ is the adapted mean of Gaussian component m, represented as

$$\mu(m) = \sum_{k=1}^{K} w_k \mathbf{s}_k(m) = \mathbf{S}(m)\mathbf{w}, \tag{5.28}$$

where $\mathbf{s}_k(m)$ is the subvector for Gaussian component m in super-vector $\mathbf{s}_k$ and $\mathbf{S}(m) = [\mathbf{s}_1(m), \ldots, \mathbf{s}_K(m)]$. $\Sigma(m)$ is the variance of the Gaussian component m, shared by all the environment-specific models. By maximizing the auxiliary function, the combination weight $\mathbf{w}$ can be solved as

$$\mathbf{w} = \left[\sum_{m,t} \gamma_t(m) \mathbf{S}^T(m) \Sigma^{-1}(m) \mathbf{S}(m) \right]^{-1} \sum_{m,t} \gamma_t(m) \mathbf{S}^T(m) \Sigma^{-1}(m) \mathbf{y}_t. \tag{5.29}$$

This model combination method is very similar to general speaker adaptation methods such as cluster adaptive training (CAT) [Gales, 2000b] and eigenvoice [Kuhn et al., 2000]. In the CAT approach, the speakers are clustered together and $\mathbf{s}_k$ stands for clusters instead of individual speakers. In the eigenvoice approach, a small number of eigenvectors are extracted from all the super-vectors and are used as $\mathbf{s}_k$. These eigenvectors are orthogonal to each other and guaranteed to represent the most important information. Although originally developed for speaker adaptation, both CAT and eigenvoice methods can be used for robust speech recognition. Storing K super-vectors in memory during online model combination may be too demanding. One way to reduce the cost is to use methods such as eigenMLLR [Chen et al., 2000, Wang et al., 2001] and transform-based CAT [Gales, 2000b] by adapting the mean vector with environment dependent transforms. In this way, only K transforms are stored in memory. Moreover, adaptive training can be used to find the canonical mean as in CAT [Gales, 2000b].

One potential problem of MLE model combination is that usually all combination weights are nonzero, i.e., every environment-dependent model contributes to the final model. This is obviously not optimal if the test environment is exactly the same as one of the training environments. There is also a scenario where the test environment can be approximated well by interpolating only few training environments. Including unrelated models into the construction brings unnecessary distortion to the target model. In ensemble speaker and speaking environment modeling (ESSEM) [Tsao and Lee, 2007, Tsao et al., 2009, Tsao and Lee, 2009], environment clustering is first used to cluster environments into several groups, each of which consists of environments having similar acoustic properties. During online model combination, an online cluster selection is first used to locate the most relevant cluster and then only the super-vectors in this selected cluster contribute to the model combination in Eq-5.25. In this way, most weights of the super-vectors are set to 0 and the method is shown to have better accuracy than simply combining all the super-vectors. By suitably incorporating prior knowledge, ESSEM can estimate combination weights accurately with a limited amount of adaptation data and has been shown to achieve very high accuracy on the standard Aurora 2 task [Tsao et al., 2014].

Instead of first doing the online clustering as in ESSEM, weights can also be automatically set to 0 [Xiao et al., 2012b] by using Lasso (least absolute shrinkage and selection operator) [Tibshirani, 1996] which imposes an L_1 regularization term in the weight estimation problem to shrink some weights to exactly zero. The auxiliary function in

Eq-5.27 is modified with the L_1 regularization as

$$Q(\mathbf{w};\mathbf{w}_0) = -\frac{1}{2}\sum_{m,t}\gamma_t(m)(\mathbf{y}_t-\mu(m))^T\Sigma^{-1}(m)(\mathbf{y}_t-\mu(m)) - T\alpha\sum_{k=1}^{K}|w_k|, \quad (5.30)$$

where α is a tuning parameter that controls the weight of the L_1 constraint, T is the total number of frames in the current utterance, and $|w_k|$ denotes the absolute value of w_k. This can be solved iteratively using the method proposed in [Li et al., 2011b]. In [Xiao et al., 2012b], it is shown that Lasso usually shrinks to zero the weights of those mean super-vectors not relevant to the test environment. By removing some irrelevant super-vectors, the obtained mean super-vectors are found to be more robust against noise distortions.

Note that the noisy speech feature variance changes with the introduction of noise, therefore simply adjusting the mean vector of the speech model cannot solve all of the problems. It is better to adjust the model variance as well. One way is to combine the pre-trained CMLLR matrices as in [Cui et al., 2009b]. However, this is not trivial, requiring numerical optimization methods, such as the gradient descent method or a Newton method [Cui et al., 2009b].

Online Model Combination for DNN

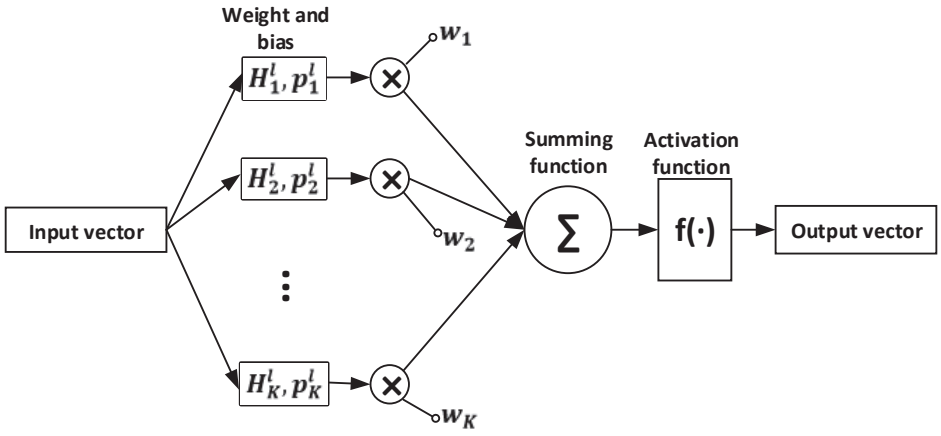

Figure 5.3: Linear model combination for DNN

The realization of Eq-5.24 in a DNN is done in the weight matrix and bias level as shown in Figure 5.3. Suppose in the l-th layer, we have trained the weight matrix set $\{\mathbf{H}_1^l,\ldots,\mathbf{H}_K^l\}$ and bias set $\{\mathbf{p}_1^l,\ldots,\mathbf{p}_K^l\}$ for all the environments. Then at test time, the weight matrix $\hat{\mathbf{A}}^l$ and bias $\hat{\mathbf{b}}^l$ for the new environment can be obtained as a linear combination of the trained counter parts as

$$\hat{\mathbf{A}}^l = \sum_{k=1}^{K} w_k \mathbf{H}_k^l, \tag{5.31}$$

$$\hat{\mathbf{b}}^l = \sum_{k=1}^{K} w_k \mathbf{p}_k^l. \tag{5.32}$$

Because the number of environments observed during training usually is much less than the number of parameters in a DNN, the combination weight $\mathbf{w} = [w_1, w_2, \ldots, w_K]^T$ can be easily estimated online with only a few utterances with the standard error back propagation training.

5.2.2 Non-Negative Matrix Factorization

In Section 5.2.1, the acoustic model for the current test utterance is obtained by combining the pre-learned acoustic models. Recently, there is increasing interest to use exemplar-based methods for general ASR [Demuynck et al., 2011, Sainath et al., 2011b] and noise-robust ASR [Gemmeke and Virtanen, 2010, Raj et al., 2010, Gemmeke et al., 2011]. Exemplar refers to an example speech segment from the training corpus. In exemplar-based noise-robust ASR [Gemmeke and Virtanen, 2010, Raj et al., 2010, Gemmeke et al., 2011], noisy speech is modeled by a linear combination of speech and noise [Gemmeke and Virtanen, 2010, Gemmeke et al., 2011] (or other interfering factors, such as music [Raj et al., 2010]) exemplars. If the reconstructed speech consists of only the exemplars of clean speech, the impact of noise is removed. This is a source separation approach, and non-negative matrix factorization (NMF) [Lee and Seung, 2000] has been shown to be a very successful method [Smaragdis and Brown, 2003, Schmidt and Olsson, 2007, Virtanen, 2007], and can directly benefit noise-robust ASR [Gemmeke and Virtanen, 2010, Raj et al., 2010, Gemmeke et al., 2011, Mohammadiha et al., 2013]. An advantage of the exemplar-based approach is that it can deal with highly non-stationary noise, such as speech recognition in the presence of background music. The source separation process with NMF is described below.

First the training corpus is used to create a dictionary $\mathbf{x}_l (1 \le l \le L)$ of clean speech exemplars and a matrix $\mathbf{X}$ is formed as $\mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_L]$. The exemplars are drawn

randomly from a collection of magnitude spectral vectors in a training set. Similarly, the noise matrix $\mathbf{N}$ is formed with noise exemplars. Then speech and noise exemplars are concatenated together to form a single matrix $\mathbf{A} = [\mathbf{XN}]$, with a total of K exemplars. The exemplars of $\mathbf{A}$ are denoted $\mathbf{a}_k, 1 \leq k \leq K$. The reconstruction feature is

$$\hat{\mathbf{y}} = \sum_{k=1}^{K} w_k \mathbf{a}_k = \mathbf{Aw}, \quad s.t. \quad w_k \geq 0 \tag{5.33}$$

with $\mathbf{w}$ as the K-dimensional activation vector. All exemplars and activation weights are required to be non-negative. The objective is to minimize the reconstruction error $d(\mathbf{y}, \mathbf{Aw})$ between the observation $\mathbf{y}$ and the reconstruction feature $\hat{\mathbf{y}}$ while constraining the matrices to be element-wise non-negative. It is also good to embed sparsity into the objective function so that the noisy speech can be represented as a combination of a small set of exemplars, similar to the concept of online GMM model combination with Lasso regularization in Section 5.2.1. This is done by penalizing the nonzero entries of $\mathbf{w}$ with the L_1 norm of the activation vector $\mathbf{w}$, weighted by element-wise multiplication (operation .*) of a non-negative vector $\boldsymbol{\lambda}$. Therefore the objective function is

$$d(\mathbf{y}, \mathbf{Aw}) + \|\boldsymbol{\lambda} .* \mathbf{w}\|_1 \quad s.t. \quad w_k \geq 0 \tag{5.34}$$

If all the elements of $\boldsymbol{\lambda}$ are zero, there is no enforced sparsity [Raj et al., 2010]. Otherwise, sparsity is enforced [Gemmeke and Virtanen, 2010, Gemmeke et al., 2011]. In [Lee and Seung, 2000], two measures are used for the reconstruction error $d(\mathbf{y}, \hat{\mathbf{y}})$, namely Euclidean distance and divergence. In most speech-related work [Gemmeke and Virtanen, 2010, Raj et al., 2010, Gemmeke et al., 2011], Kullback-Leibler (KL) divergence is used to measure the reconstruction error.

$$d(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{e=1}^{E} y_e \log\left(\frac{y_e}{\hat{y}_e}\right) - y_e + \hat{y}_e, \tag{5.35}$$

where E is the vector dimension.

To solve Eq-5.34, the entries of the vector $\mathbf{w}$ are initialized to unity. Then Eq-5.34 can be minimized by iteratively applying the multiplicative update rule [Gemmeke et al., 2011]

$$\mathbf{w} \leftarrow \mathbf{w} .* (\mathbf{A}(\mathbf{y}./(\mathbf{Aw}))) ./ (\mathbf{A}1 + \boldsymbol{\lambda}) \tag{5.36}$$

with $.*$ and $./$ denoting element-wise multiplication and division, respectively. 1 is a vector with all elements set to 1.

After getting $\mathbf{w}$, the clean speech feature can be reconstructed by simply combining all the speech exemplars with nonzero weights [Schmidt and Olsson, 2007]. Good recognition performance has been observed particularly at very low SNR (below 0 dB). Better results are reported by using the following filtering [Raj et al., 2010, Gemmeke et al., 2011, Gemmeke and Van hamme, 2012] as

$$\hat{\mathbf{x}} = \mathbf{y}. * \mathbf{A}^x\mathbf{w}^x./(\mathbf{A}^x\mathbf{w}^x + \mathbf{A}^n\mathbf{w}^n), \tag{5.37}$$

where $\mathbf{A}^x$ and $\mathbf{w}^x$ denote the exemplars and activation vector for clean speech, respectively, and $\mathbf{A}^n$ and $\mathbf{w}^n$ denote the exemplars and activation vector for noise, respectively. This procedure can be viewed as filtering the noisy speech spectrum with a time-varying filter defined by $\mathbf{A}^x\mathbf{w}^x./(\mathbf{A}^x\mathbf{w}^x + \mathbf{A}^n\mathbf{w}^n)$, similar to Wiener filtering in Eq-4.28. This is referred as feature enhancement (FE) in [Gemmeke et al., 2011, Gemmeke and Van hamme, 2012].

Instead of cleaning the noisy speech magnitude spectrum, a sparse classification (SC) method is proposed in [Gemmeke and Virtanen, 2010] to directly use the activation weights to estimate the state or word likelihood. Since each frame of each speech exemplar in the speech dictionary has state or word labels obtained from the alignment with conventional HMMs, the weights of the exemplars in the sparse representation $\mathbf{w}^x$ can be used to calculate the state or word likelihood. Then, these activation-based likelihoods are used in a Viterbi search to obtain the state sequence with the maximum likelihood criterion.

Although the root methodology of FE and SC are the same, i.e., NMF source separation, it is shown in [Weninger et al., 2012, Gemmeke and Van hamme, 2012] that they are complementary. If combined together, more gain can be achieved. There are also variations of standard NMF source separation. For example, a sliding time window approach [Gemmeke et al., 2009] that allows the exemplars to span multiple frames is used for decoding utterances of arbitrary length. Convolutive extension of NMF is proposed to handle potential dependencies across successive input columns [Smaragdis, 2007, Weninger et al., 2012]. Prior knowledge of the co-occurrence statistics of the basis functions for each source can also be employed to improve the performance of NMF [Wilson et al., 2008]. In [Grais and Erdogan, 2013], by minimizing cross-coherence between the dictionaries of all sources in the mixed signal, the bases set of one source dictionary can be prevented from representing the other source signals. This clearly gives better separation results than the traditional NMF. Superior digit recognition accuracy has been reported in [Gemmeke and Van hamme, 2012] with the exemplar-based method by increasing the number of update iterations and exemplars, designing artificial noise dictionary, doing noise sniffing, and combining SC with FE.

Although the objective of NMF is to accurately recover clean features from noisy features, most NMF approaches have not directly optimized this objective. This problem is addressed in [Weninger et al., 2014d], where discriminative training of the NMF bases are performed so that given the weight coefficients obtained on a noisy feature, the desired clean feature is optimally recovered. This is done by minimizing the distance between the recovered and reference clean feature. However, this objective becomes a bi-level optimization problem because the recovered clean feature also depends on the bases. Therefore, this involves very complicated iterative inference. In [Hershey et al., 2014] a concept of deep unfolding is proposed to address this issue by unfolding the inference iterations as layers in a DNN. Rather than optimizing the original model, the method unties the model parameters across layers to create a more powerful DNN. Then this DNN can be optimized with the back-propagation algorithm. This deep unfolding method gives superior performance to discriminative NMF than the solution in [Weninger et al., 2014d].

There are still plenty of challenges. e.g., how to deal with convolutive channel distortions [Gemmeke et al., 2013], how to most effectively deal with noise types in testing that have not been previously seen in the development of the noise dictionary [Gemmeke and Van hamme, 2012], and how to generalize to LVCSR tasks although there are recent improvements on Aurora 4 tasks [Geiger et al., 2014a]. Finally, although it is challenging to a noise-robust front-end to improve over the performance of a DNN back-end fed with raw features, it is reported in [Geiger et al., 2014a] that NMF enhancement improves the recognition accuracy substantially when the training data is clean, and it still brings improvement even with multi-condition training data.

5.2.3 Variable-Parameter Modeling

We have seen two broad classes of variables that affect the observation of speech signals: discrete (e.g. speaker and speaker classes, types of noises) and continuous (e.g. SNR, speaking rate, distance to the microphone). The variability of speech signals as a function of continuous variables can be explicitly modeled in the acoustic models. The concept of variable-parameter modeling is that speech model parameters in a specific test environment can be obtained as a function of environment variables. There are three advantages with this modeling techniques.

- When the test environment is unseen during training, the model parameters can still be extrapolated very well with the learned function. Therefore, this method generalizes very well to unseen test environments.

- With the introduction of a continuous parameterization of the model, any data sample contributes to the training of all the model parameters of the variable-parameter model. This improves training data effectiveness compared to multi-condition training where there is no design to leverage data across conditions.
- Another advantage is that the model is sharper than the model trained with standard multi-style training because it can fit the underlying individual test environments better by adjusting its parameters according to the test environment variable. This concept is first proposed to dynamically adjust GMM parameters [Cui and Gong, 2003, 2006, 2007], and then extended to DNN modeling [Zhao et al., 2014a,b].

Variable-Parameter Modeling for GMM

As shown in Cui and Gong [2007], the mean and variance of the Gaussian distribution of the observed speech acoustic feature are functions of SNR. Pooling such distributions together and training SNR-independent models, as multi-style training does, inevitably yields relatively flat distributions. Apparently, the standard GMM-HMM which employs a constant set of model parameters to describe the acoustics under all different environments is imperfect and inadequate to deal with the phenomena.

To improve the modeling accuracy and performance, it is better to make the parameters of the acoustic model change according to the environment. This is the motivation of variable-parameter HMM (VPHMM) [Cui and Gong, 2003, 2006, 2007] which models the speech Gaussian mean and variance parameters as a set of polynomial functions of an environment variable u. A popular environment variable is SNR [Cui and Gong, 2003, 2006, 2007]. Hence, the Gaussian component m is now modeled as $\mathcal{N}(\mathbf{y};\boldsymbol{\mu}(m,u),\Sigma(m,u))$. $\boldsymbol{\mu}(m,u)$ and $\Sigma(m,u)$ are polynomial functions of environment variable u. For example, $\boldsymbol{\mu}(m,u)$ can be denoted by

$$\boldsymbol{\mu}(m,u) = \sum_{j=0}^{J} \mathbf{c}_j(m) u^j, \tag{5.38}$$

where $\mathbf{c}_j(m)$ is a vector with the same dimension as the input feature vectors. The choice of polynomial function is based on its good approximation to continuous functions, its simple derivation operations, and the fact that the change of means and variances in terms of the environment is smooth and can be modeled by low order polynomials. Note that strictly speaking, variable parameter modeling using Eq-5.38 should be called variable-parameter Gaussian mixture model instead of VPHMM because only Gaussian parameters are modeled with polynomial functions, although transition probabilities can also be modeled using variable parameter techniques (e.g. speaking rate changes). Here

we still use the term VPHMM to follow the literature in which it was proposed [Cui and Gong, 2003, 2006, 2007].

Other functions can also be used for a VPHMM. For example, in [Yu et al., 2009b], piecewise spline interpolation is used to represent the dependency of the HMM parameters on the environment parameters. To reduce the total number of parameters for a VPHMM, parameter clustering can be employed [Yu et al., 2008c]. The VPHMM parameters can be trained either with the MLE criterion [Cui and Gong, 2007] or a discriminative criterion [Yu et al., 2008b]. In addition to Gaussian mean and variance parameters, other model parameters can also be modeled. In [Cheng et al., 2011, Li et al., 2013b], a more generalized form of VPHMM is investigated by modeling tied linear transforms as a function of environment variables. In addition to using the standard MFCC as the input feature for a GMM, [Xie et al., 2014] shows the effectiveness of using bottle-neck features generated from a DNN as the features of a VPHMM.

During testing, the actual set of speech model parameters can be calculated by evaluating the parametric function with the estimated environment variable. Even if the estimated environment is not seen during training, the curve fitting optimization naturally uses the information on articulation/context from neighboring environments. Therefore, VPHMM can work well in unseen environment instances modeled by the environment variable.

Variable-Component DNN

Usually multi-style data is used to train a DNN [Seltzer et al., 2013a] and good accuracies can be obtained. However, as shown in Section 3.4, speech samples from different environments cannot be well aligned even with the DNN's high-level feature extraction. Therefore, if a single DNN is used to model the multi-style speech data, it is possible to end up with "flat" distributions. So for the test speech produced in a particular environment, such a "flat" model would not be the optimal matched model. Actually, a flat model does not represent any of the training environments. It is also difficult to collect training data to cover all possible types of environments, so the performance on unseen noisy environments remains unpredictable. Therefore, it is desirable that DNN components can be modeled as a function of a continuous environment-dependent variable. At the recognition time, a set of DNN components specific to the given value of the environment variable is instantiated and used for recognition. Even if the test environment is not seen in the training, the estimated DNN components can still work well because the change of DNN components in terms of the environment variable can be predicted. Variable-component DNN (VCDNN) [Zhao et al., 2014a,b] is proposed for this purpose.

In the VCDNN method, any component in the DNN can be modeled as a set of polynomial functions of an environment variable. To that end, four types of variation can be defined for VCDNN: variable-parameter DNN (VPDNN) in which the weight matrix and bias are variable dependent, variable-output DNN (VODNN) in which the output of each hidden layer is variable dependent, variable-activation DNN (VADNN) in which the activation function is variable dependent, and variable-input DNN (VIDNN) in which the input feature is variable dependent.

Figure 5.4 shows the flow chart of one layer of a VPDNN , in which the weight matrix $\mathbf{A}$ and bias $\mathbf{b}$ of layer l is modeled as a function of the environment variable u:

$$\mathbf{A}^l = \sum_{j=0}^{J} \mathbf{H}_j^l u^j \qquad 0 < l \leq L \tag{5.39}$$

$$\mathbf{b}^l = \sum_{j=0}^{J} \mathbf{p}_j^l u^j \qquad 0 < l \leq L \tag{5.40}$$

J is the polynomial function order. $\mathbf{H}_j^l$ is a matrix with the same dimensions as $\mathbf{A}^l$ and $\mathbf{p}_j^l$ is a vector with the same dimension as $\mathbf{b}^l$.

Then the relation between the input $\mathbf{v}^l$ and the output $\mathbf{v}^{l+1}$ of the l-th layer at a VPDNN is

$$\mathbf{v}^{l+1} = \sigma(\mathbf{z}^l), \tag{5.41}$$

where

$$\mathbf{z}^l = \mathbf{A}^l \mathbf{v}^l + \mathbf{b}^l \tag{5.42}$$

and $\sigma(\cdot)$ is the sigmoid function.

Combining Eq-5.39 and 5.40 with the error back propagation algorithm introduced in Section 2.4, the update formulas for $\mathbf{H}_j^l$ and $\mathbf{p}_j^l$ can be obtained as:

$$\hat{\mathbf{H}}_j^l = \mathbf{H}_j^l + \alpha \mathbf{v}^l (\mathbf{e}^l)^T u^j \tag{5.43}$$

$$\hat{\mathbf{p}}_j^l = \mathbf{p}_j^l + \alpha \mathbf{e}^l u^j \tag{5.44}$$

where $\mathbf{v}^l$ is the input to the l-th layer, α is the learning rate, and $\mathbf{e}^l$ is the error signal at the l-th layer, defined in Eq-2.38.

In the recognition stage, the weight matrix $\mathbf{A}$ and bias $\mathbf{b}$ of each layer are instantiated according to Eq-5.39 and 5.40 with the estimated environment variable of the test data. Then the senone posterior can be calculated in the same way as in the standard DNN.

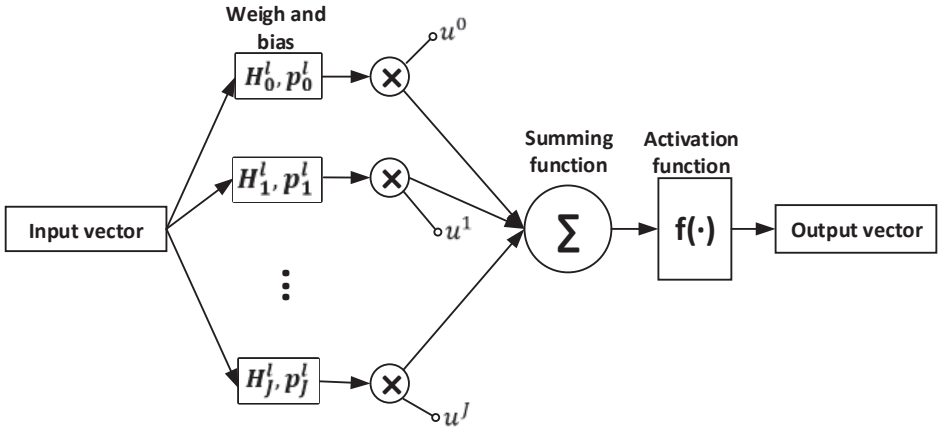

Figure 5.4: Variable-parameter DNN

Comparing Eq-5.39 and Eq-5.40 in VPDNN with Eq-5.31 and Eq-5.32 in linear DNN model combination, we can see that they are very similar – both methods linearly combine a set of basis matrix and bias at test time. However, they also have several different aspects as shown in Table 5.1. Similar comparison can also be applied to VPHMM and linear GMM model combination.

Table 5.1: Difference between VPDNN and linear DNN model combination

	VPDNN	linear DNN model combination
DNN weight matrix and bias of test utterances	a learned polynomial function of environment variables	linear combination of a set of weight matrix and bias trained from different environments
combination coefficients	directly calculated with environment variables	online estimated
environment variables	can be continuous such as SNR	discrete, each associated with a weight matrix and bias

In a VODNN , it is assumed the output of each hidden layer could be described by a polynomial function of the environment variable u:

$$\mathbf{v}^{l+1} = \sum_{j=0}^{J} \sigma(\mathbf{z}_j^l) u^j \qquad 0 < l < L \tag{5.45}$$

where

$$\mathbf{z}_j^l = (\mathbf{H}_j^l)^T \mathbf{v}^l + \mathbf{p}_j^l \tag{5.46}$$

The framework of one layer in a VODNN is shown in Figure 5.5. Similarly, the updating formulas can be obtained by combining Eq-5.45 and Eq-5.46 with the error back propagation algorithm:

$$\hat{\mathbf{H}}_j^l = \mathbf{H}_j^l + \alpha \mathbf{v}^l (\mathbf{e}_j^l)^T u^j \tag{5.47}$$

$$\hat{\mathbf{p}}_j^l = \mathbf{p}_j^l + \alpha \mathbf{e}_j^l u^j \tag{5.48}$$

The difference between the update formulas of VPDNN and VODNN parameters is that the order-independent error signal $\mathbf{e}^l$ is used in Eq-5.43 and 5.44 while the error signal $\mathbf{e}_j^l$ used in Eq-5.47 and 5.48 depends on the polynomial order j as

$$e_{i(j)}^l = \left[\sum_{n=0}^{J} \sum_{k=1}^{N_{l+1}} h_{ik(n)}^{l+1} e_{k(n)}^{l+1} \right] \sigma'(z_{ij}^l) \tag{5.49}$$

where $e_{i(j)}^l$ is the i-th element of the error signal vector $\mathbf{e}_j^l$ at the l-th layer, z_{ij}^l is the i-th element of $\mathbf{z}_j^l$, and $h_{ik(n)}^{l+1}$ is the element of matrix $\mathbf{H}_n^{l+1}$ in the i-th row and k-th column at the layer $l+1$. $\sigma'(\cdot)$ is the derivative of the sigmoid function.

In a VADNN , the activation function of hidden layers has environment-variable-dependent parameters as

$$\mathbf{v}^{l+1} = \sigma\left(\mathbf{a}^l .* \mathbf{z}^l + \mathbf{m}^l\right) \tag{5.50}$$

where $\mathbf{z}^l$ is defined in Eq-5.42 and $.*$ means the element-wise product. $\mathbf{a}^l$ and $\mathbf{m}^l$ are defined as the polynomial functions of the environment variable u

$$\mathbf{a}^l = \sum_{j=0}^{J} \mathbf{h}_j^l u^j \qquad 0 < l < L \tag{5.51}$$

$$\mathbf{m}^l = \sum_{j=0}^{J} \mathbf{p}_j^l u^j \qquad 0 < l < L \tag{5.52}$$

Figure 5.6 shows one layer of a VADNN. The additional variable-dependent parameters $\mathbf{h}_j^l$ and $\mathbf{p}_j^l$ in a VADNN for each hidden layer are vectors with dimension N_l, which

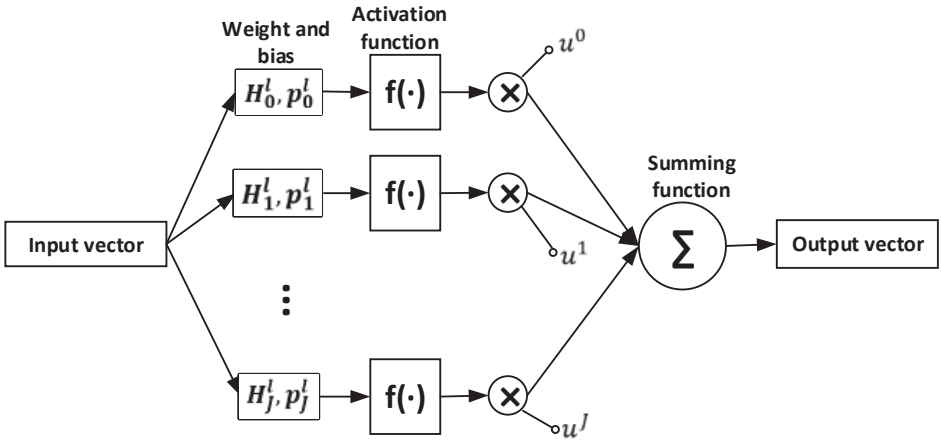

Figure 5.5: Variable-output DNN

is the number of nodes of the l-th layer. Hence its number of parameters is much smaller than that in a VPDNN or a VODNN. In the training of a VADNN, $\mathbf{h}_j^l$ and $\mathbf{p}_j^l$ as well as the DNN parameters $\mathbf{A}^l$ and $\mathbf{b}^l$ need to be updated with the error back propagation algorithm as

$$\hat{\mathbf{A}}^l = \mathbf{A}^l + \alpha \mathbf{v}^l (\mathbf{e}^l .* \mathbf{a}^l)^T \tag{5.53}$$

$$\hat{\mathbf{b}}^l = \mathbf{b}^l + \alpha (\mathbf{e}^l .* \mathbf{a}^l) \tag{5.54}$$

$$\hat{\mathbf{h}}_j^l = \mathbf{h}_j^l + \alpha (\mathbf{e}^l .* \mathbf{z}^l) u^j \tag{5.55}$$

$$\hat{\mathbf{p}}_j^l = \mathbf{p}_j^l + \alpha \mathbf{e}^l u^j \tag{5.56}$$

Finally, the simplest DNN structure to use environment variables is VIDNN , which concatenates environment variables with the original input feature. Even with the first-order polynomial, a VPDNN or a VODNN doubles the number of parameters from the standard DNN. If a large amount of training data is available, these two models may give better accuracy. In contrast, a VADNN or a VIDNN only increases negligibly the number of parameters, but still achieves satisfactory robustness.

The advantage of VCDNNs is shown in [Zhao et al., 2014a] where VCDNNs achieved better relative WER reduction from the standard DNN under unseen SNR conditions than under the seen SNR conditions. This indicates that a standard DNN has a strong power to model the various environments it has observed, but for the unseen environments, there is more room for improvement from the standard DNN. With the

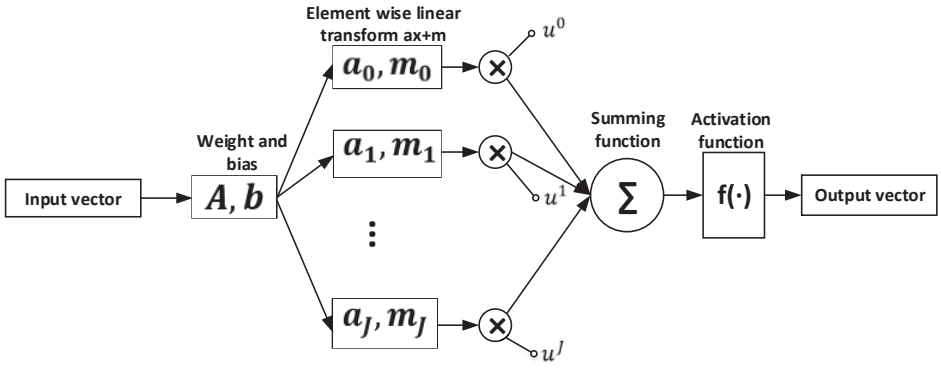

Figure 5.6: Variable-activation DNN

polynomial function, VCDNNs can very well predict the DNN components used for unseen condition by extrapolation. Therefore, VCDNNs can generalize very well to unseen environments.

5.3 Summary

Method	Proposed around	Characteristics
empirical cepstral compensation [Acero and Stern, 1990, Acero, 1993, Liu et al., 1994, Stern et al., 1996, Droppo et al., 2001a]	1990	calculates all kinds of factor-dependent (including SNR, VQ cluster, phoneme identity, etc.) bias using stereo training data, and remove that bias during testing
online GMM model combination [Kuhn et al., 2000, Gales, 2000b]	2000	online combines a set of environment-specific GMM models, representative methods are eigenvoice and cluster adaptive training
stereo piecewise linear compensation for environment (SPLICE) [Deng et al., 2000a]	2000	the additive correction vector is piecewise linear between the noisy speech observation and the clean speech of the stereo training data
variable-parameter HMM (VPHMM) [Cui and Gong, 2003]	2003	models the Gaussian mean and variance parameters as a set of polynomial functions of the environment variable

ensemble speaker and speaking environment modeling (ESSEM) [Tsao and Lee, 2007, Tsao et al., 2009]	2007	To remove unrelated models into construction, an online cluster selection is first used to locate the most relevant cluster and then only the super-vectors in this selected cluster contribute to the model combination
exemplar-based reconstruction with non-negative matrix factorization (NMF) [Gemmeke and Virtanen, 2010, Raj et al., 2010]	2010	NMF is used to reconstruct speech with only clean speech exemplars extracted from the training dictionary
Lasso model combination [Xiao et al., 2012b]	2012	imposes an L_1 regularization term in the weight estimation problem of online model combination to shrink some weights to exactly zero
discriminative NMF [Weninger et al., 2014d]	2014	discriminative training of the NMF bases is performed so that the desired clean feature is optimally recovered given the weight coefficients obtained on a noisy feature

Table 5.2: Compensation with prior knowledge methods originally proposed for GMMs in Chapter 5, arranged chronologically

Method	Proposed around	Characteristics
RNN for noise removal [Maas et al., 2012b, Wöllmer et al., 2013b, Weninger et al., 2014a,c]	2012	Uses a RNN which better models temporal sequence to learn the mapping from noisy feature to clean feature, and it is extended with advanced structure such as LSTM and BLSTM
DNN for noise removal [Lu et al., 2013c, Feng et al., 2014b, Du et al., 2014a, Narayanan and Wang, 2014a]	2013	Use a DNN to learn the mapping from noisy feature to clean feature
online DNN model combination [Wu and Gales, 2015, Tan et al., 2015]	2015	online combines a set of environment-specific DNN models
variable-component DNN [Zhao et al., 2014a,b]	2014	any component in the DNN can be modeled as a set of polynomial functions of an environment variable so that better modeling of test environments can be achieved.
deep unfolding [Hershey et al., 2014]	2014	solves the complicated bi-level optimization problem in discriminative NMF by unfolding the inference iterations as layers in a DNN

Table 5.3: Compensation with prior knowledge methods originally proposed for DNNs in Chapter 5, arranged chronologically

To provide a better view of the development trend of robustness methods from the GMM era to the DNN era, we summarize the representative methods described in this chapter for robust ASR exploiting prior knowledge originally proposed for GMM and DNN in Table 5.2 and Table 5.3, respectively, in a chronological order. Further comments and summary of these methods, as well as additional relevant work that we did not describe in detail in this chapter, are made below:

- If stereo data is available, a mapping from noisy feature to clean feature can be learned. Empirical cepstral compensation is widely used to address all kinds of factors (SNR, VQ cluster, phoneme identity, etc.) with a bias, and it is improved by SPLICE which uses piecewise linear compensation. With the layer-by-layer nonlinear modeling power of a DNN, a much better learning of the noisy-to-clean feature mapping can be obtained. This is further improved by the introducing the recurrent structure and (B)LSTM units which better model the temporal sequence of speech signals.
- Online model combination is one way to fast adapt acoustic models to environments with limited adaptation data because only combination coefficients need to be computed online. For GMM models, eigenvoice and cluster adaptive training are representative methods. The combination coefficients can be made sparse with either clustering or L_1 regularization. Similar idea can be easily extended to DNN by online combining weight matrices.
- VPHMM is another way to online constructing an adapted GMM model with a set of polynomial functions of the environment variable. It is extended to VCDNN in which any component in the DNN (parameter in VPDNN, output in VODNN, activation in VADNN, and input in VIDNN, respectively) can be modeled as a set of polynomial functions of an environment variable. With the polynomial functions, the model can be instantiated even in the unseen case by extrapolation, thus enjoying good generalization property.
- NMF is used to reconstruct the clean speech spectrum from the noisy speech spectrum using pre-constructed clean speech and noise exemplars. While no stereo data is required, examples of the corrupting noise are nevertheless required to form the noise dictionary. There are plenty of extensions of NMF, including discriminative NMF in which discriminative training of the NMF bases is per-

formed so that the desired clean feature is optimally recovered given the weight coefficients obtained on a noisy feature.

- Last, deep unfolding was proposed to solve the complicated bi-level optimization problem in discriminative NMF. It builds a bridge between DNN modeling and model-based approaches. As will be described in detail in Chapter 6, model-based approaches are very powerful because of the use of explicit distortion models between the clean and distorted speech. However, the inference is sometimes very complicated and may rely on the underlying Gaussian model assumption. On the other hand, it is straightforward to optimize parameters in DNN modeling with the back propagation algorithm. A well-known disadvantage of the DNN is that it is closer to mechanisms than problem-level formulation, and is usually considered as a "black-box". Deep unfolding may be a potential framework that allows model-based approaches to guide the exploration of the space of DNNs, which is important but missing in current literature.

References

A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition*. Cambridge University Press, 1993.

A. Acero and R. Stern. Environmental robustness in automatic speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume 2, pages 849–852, 1990.

M. Afify, X. Cui, and Y. Gao. Stereo-based stochastic mapping for robust speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume IV, pages 377–380, 2007.

M. Afify, X. Cui, and Y. Gao. Stereo-based stochastic mapping for robust speech recognition. *IEEE Trans. on Audio, Speech and Language Processing*, 17(7):1325–1334, 2009.

L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume 11, pages 49–52, 1997.

K. T. Chen, W. W. Liau, H. M. Wang, and L. S. Lee. Fast speaker adaptation using eigenspace-based maximum likelihood linear regression. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 3, pages 742–745, 2000.

N. Cheng, X. Liu, and L. Wang. Generalized variable parameter HMMs for noise robust speech recognition. In *Proc. Interspeech*, pages 481–484, 2011.

X. Cui and Y. Gong. Variable parameter Gaussian mixture hidden Markov modeling for speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume I, pages 12–15, 2003.

X. Cui and Y. Gong. Modeling variance variation in a variable parameter HMM framework for noise robust speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume I, 2006.

X. Cui and Y. Gong. A study of variable-parameter Gaussian mixture hidden Markov modeling for noisy speech recognition. *IEEE Trans. on Audio, Speech and Language Processing*, 15(4):1366–1376, 2007.

X. Cui, M. Afify, and Y. Gao. MMSE-based stereo feature stochastic mapping for noise robust speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 4077–4080, 2008a.

X. Cui, M. Afify, and Y. Gao. N-best based stochastic mapping on stereo HMM for noise robust speech recognition. In *Proc. Interspeech*, pages 1261–1264, 2008b.

X. Cui, M. Afify, and Y. Gao. Stereo-based stochastic mapping with discriminative training for noise robust speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 3933–3936, 2009a.

X. Cui, J. Xue, and B. Zhou. Improving online incremental speaker adaptation with eigen feature space MLLR. In *Proc. Workshop on Automatic Speech Recognition and Understanding*, pages 136–140, 2009b.

K. Demuynck, D. Seppi, D. Van Compernolle, P. Nguyen, and G. Zweig. Integrating meta-information into exemplar-based speech recognition with segmental conditional random fields. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 5048–5051, 2011.

L. Deng, A. Acero, M. Plumpe, and X. Huang. Large vocabulary speech recognition under adverse acoustic environment. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 3, pages 806–809, 2000a.

L. Deng, A. Acero, L. Jiang, J. Droppo, and X. D. Huang. High-performance robust speech recognition using stereo training data. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 301–304, 2001.

L. Deng, J. Droppo, and A. Acero. Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition. *IEEE Trans. on Audio, Speech and Language Processing*, 11(6):568–580, 2003c.

L. Deng, J. Wu, J. Droppo, and A. Acero. Analysis and comparison of two speech feature extraction/compensation algorithms. *IEEE Signal Processing Letters*, 12(6): 477–480, 2005b.

J. Droppo and A. Acero. Maximum mutual information SPLICE transform for seen and unseen conditions. In *Proc. Interspeech*, pages 989–992, 2005.

J. Droppo, A. Acero, and L. Deng. Efficient on-line acoustic environment estimation for FCDCN in a continuous speech recognition system. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume 1, pages 209–212, 2001a.

J. Droppo, L. Deng, and A. Acero. Evaluation of the SPLICE algorithm on the Aurora2 database. In *Proc. Eurospeech*, pages 217–220, 2001b.

J. Droppo, L. Deng, and A. Acero. Uncertainty decoding with SPLICE for noise robust speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume 1, pages 57–60, 2002.

J. Du, Y. Hu, L. R. Dai, and R. H. Wang. HMM-based pseudo-clean speech synthesis for splice algorithm. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 4570–4573, 2010.

J. Du, L. Dai, and Q. Huo. Synthesized stereo mapping via deep neural networks for noisy speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 1764–1768, 2014a.

J. Du, Q. Wang, T. Gao, Y. Xu, L. Dai, and C.-H. Lee. Robust speech recognition with speech enhanced deep neural networks. In *Proc. Interspeech*, 2014b.

ETSI. *Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms*. ETSI, 2002.

X. Feng, Y. Zhang, and J. Glass. Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2014b.

M. J. F. Gales. Cluster adaptive training of hidden Markov models. *IEEE Trans. Speech and Audio Processing*, 8(4):417–428, 2000b.

T. Gao, J. Du, L.-R. Dai, and C.-H. Lee. Joint training of front-end and back-end deep neural networks for robust speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2015.

J. Geiger, J. Gemmeke, B. Schuller, and G. Rigoll. Investigating NMF speech enhancement for neural network based acoustic models. *Proc. Interspeech*, 2014a.

J. F. Gemmeke and H. Van hamme. Advances in noise robust digit recognition using hybrid exemplar-based techniques. In *Proc. Interspeech*, pages 2134–2137, 2012.

J. F. Gemmeke and T. Virtanen. Noise robust exemplar-based connected digit recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 4546–4549, 2010.

J. F. Gemmeke, L. ten Bosch, L. Boves, and B. Cranen. Using sparse representations for exemplar based continuous digit recognition. In *Proc. EUSIPCO*, pages 1755–1759, 2009.

J. F. Gemmeke, T. Virtanen, and A. Hurmalainen. Exemplar-based sparse representations for noise robust automatic speech recognition. *IEEE Trans. on Audio, Speech and Language Processing*, 19(7):2067–2080, 2011.

J. F. Gemmeke, T. Virtanen, and K. Demuynck. Exemplar-based joint channel and noise compensation. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 868–872, 2013.

E. M. Grais and H. Erdogan. Discriminative nonnegative dictionary learning using cross-coherence penalties for single channel source separation. In *Proc. Interspeech*, pages 808–812, 2013.

J. Hershey, J. Le Roux, and F. Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv preprint arXiv:1409.2574*, 2014.

P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Deep learning for monaural speech separation. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2014a.

R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski. Rapid speaker adaptation in eigenvoice space. *IEEE Trans. Speech and Audio Processing*, 8(6):695–707, 2000.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. Neural Information Processing Systems*, pages 556–562, 2000.

J. Li, M. Yuan, and C. H. Lee. Lasso model adaptation for automatic speech recognition. In *ICML Workshop on Learning Architectures, Representations, and Optimization for Speech and Visual Information Processing*, 2011b.

Y. Li, X. Liu, and L. Wang. Feature space generalized variable parameter HMMs for noise robust recognition. In *Proc. Interspeech*, pages 2968–2972, 2013b.

B. Liu, L. Dai, J. Li, and R. H. Wang. Double Gaussian based feature normalization for robust speech. In *International Symposium on Chinese Spoken Language Processing*, pages 705–708, 2004.

F.-H. Liu, R. M. Stern, A. Acero, and P. Moreno. Environment normalization for robust speech recognition using direct cepstral comparison. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume I, pages 61–64, 1994.

X. Lu, Y. Tsao, S. Matsuda, and C. Hori. Speech enhancement based on deep denoising autoencoder. In *Proc. Interspeech*, pages 436–440, 2013c.

A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng. Recurrent neural networks for noise reduction in robust ASR. In *Proc. Interspeech*, pages 22–25, 2012b.

N. Mohammadiha, P. Smaragdis, and A. Leijon. Supervised and unsupervised speech enhancement using nonnegative matrix factorization. *IEEE Trans. on Audio, Speech and Language Processing*, 21(10):2140–2151, 2013.

A. Narayanan and D. Wang. Investigation of speech separation as a front-end for noise robust speech recognition. *IEEE/ACM Trans. on Audio, Speech and Language Processing*, 22(4):826–835, 2014a.

A. Narayanan and D.L. Wang. Ideal ratio mask estimation using deep neural networks. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 7092–7096, 2013a.

A. Narayanan and D.L. Wang. Joint noise adaptive training for robust automatic speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2014b.

D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig. fMPE: Discriminatively trained features for speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume 1, pages 961–964, 2005a.

B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh. Non-negative matrix factorization based compensation of music for automatic speech recognition. In *Proc. Interspeech*, pages 717–720, 2010.

T. N. Sainath, B. Ramabhadran, M. Picheny, D. Nahamoo, and D. Kanevsky. Exemplar-based sparse representation features: from TIMIT to LVCSR. *IEEE Trans. on Audio, Speech and Language Processing*, 19(8):2598–2613, 2011b.

M. N. Schmidt and R. K. Olsson. Linear regression on sparse features for single-channel speech separation. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 26–29, 2007.

M. L. Seltzer, D. Yu, and Y. Wang. An investigation of deep neural networks for noise robust speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 7398–7402, 2013a.

P. Smaragdis. Convolutive speech bases and their application to supervised speech separation. *IEEE Trans. on Audio, Speech and Language Processing*, 15(1):1–12, 2007.

P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, 2003.

R. Stern, A. Acero, F. H. Liu, and Y. Ohshima. Signal processing for robust speech recognition. In C.-H. Lee, F. K. Soong, and K. K. Paliwal, editors, *Automatic speech and speaker recognition : advanced topics*, chapter 15, pages 357–384. Kluwer Academic Publishers, 1996.

T. Tan, Y. Qian, M. Yin, Y. Zhuang, and K. Yu. Cluster adaptive training for deep neural network. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2015.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58(1):267–288, 1996.

K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume 3, pages 1315–1318, 2000.

Y. Tsao and C. H. Lee. An ensemble modeling approach to joint characterization of speaker and speaking environments. In *Proc. Interspeech*, pages 1050–1053, 2007.

Y. Tsao and C. H. Lee. An ensemble speaker and speaking environment modeling approach to robust speech recognition. *IEEE Trans. Speech and Audio Processing*, 17 (5):1025–1037, 2009.

Y. Tsao, J. Li, and C. H. Lee. Ensemble speaker and speaking environment modeling approach with advanced online estimation process. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 3833–3836, 2009.

Y. Tsao, S. Matsuda, C. Hori, H. Kashioka, and C.-H. Lee. A map-based online estimation approach to ensemble speaker and speaking environment. *IEEE/ACM Trans. on Audio, Speech and Language Processing*, 2014.

Y. Tu, J. Du, L.-R. Dai, and C.-H. Lee. Speech separation based on signal-noise-dependent deep neural networks for robust speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2015.

T. Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans. on Audio, Speech and Language Processing*, 15(3):1066–1074, 2007.

N.J.-C. Wang, S.S.-M. Lee, F. Seide, and L. S. Lee. Rapid speaker adaptation using a priori knowledge by eigenspace analysis of MLLR parameters. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, volume 1, pages 345–348, 2001.

Y. Wang and D. Wang. Towards scaling up classification-based speech separation. *IEEE Trans. on Audio, Speech and Language Processing*, 21(7):1381–1390, 2013.

Y. Wang, A. Narayanan, and D. Wang. On training targets for supervised speech separation. *IEEE/ACM Trans. on Audio, Speech and Language Processing*, 22(12): 1849–1858, 2014.

C. Weng, D. Yu, M. Seltzer, and J. Droppo. Single-channel mixed speech recognition using deep neural networks. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2014a.

F. Weninger, M. Wöllmer, J. Geiger, B. Schuller, J. Gemmeke, A. Hurmalainen, T. Virtanen, and G. Rigoll. Non-negative matrix factorization for highly noise-robust ASR: to enhance or to recognize? In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 4681–4684, 2012.

F. Weninger, F. Eyben, and B. Schuller. Single-channel speech separation with memory-enhanced recurrent neural networks. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2014a.

F. Weninger, J. Hershey, J. Le Roux, and B. Schuller. Discriminatively trained recurrent neural networks for single-channel speech separation. In *Proc. IEEE GlobalSIP Symposium on Machine Learning Applications in Speech Processing*, 2014c.

F. Weninger, J. Le Roux, J. Hershey, and S. Watanabe. Discriminative NMF and its application to single-channel source separation. *Proc. Interspeech*, 2014d.

K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran. speech denoising using nonnegative matrix factorization with priors. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 4029–4032, 2008.

M. Wöllmer, Z. Zhang, F. Weninger, B. Schuller, and G. Rigoll. Feature enhancement by bidirectional lstm networks for conversational speech recognition in highly non-stationary noise. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 6822–6826, 2013b.

C. Wu and M. J. F. Gales. Multi-basis adaptive neural network for rapid adaptation in speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, 2015.

X. Xiao, J. Li, E. S. Chng, and H. Li. Lasso environment model combination for robust speech recognition. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 4305–4308, 2012b.

X. Xie, R. Su, X. Liu, and L. Wang. Deep neural network bottleneck features for generalized variable parameter HMMs. In *Proc. Interspeech*, 2014.

Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee. Dynamic noise aware training for speech enhancement based on deep neural networks. In *Proc. Interspeech*, pages 2670 – 2674, 2014.

D. Yu, L. Deng, Y. Gong, and A. Acero. Discriminative training of variable-parameter HMMs for noise robust speech recognition. In *Proc. Interspeech*, pages 285–288, 2008b.

D. Yu, L. Deng, Y. Gong, and A. Acero. Parameter clustering and sharing in variable-parameter HMMs for noise robust speech recognition. In *Proc. Interspeech*, pages 1253–1256, 2008c.

D. Yu, L. Deng, Y. Gong, and A. Acero. A novel framework and training algorithm for variable-parameter hidden Markov models. *IEEE Trans. on Audio, Speech and Language Processing*, 17(7):1348–1360, 2009b.

R. Zhao, J. Li, and Y. Gong. Variable-component deep neural network for robust speech recognition. In *Proc. Interspeech*, 2014a.

R. Zhao, J. Li, and Y. Gong. Variable-activation and variable-input deep neural network for robust speech recognition. In *Proc. IEEE Spoken Language Technology Workshop*, 2014b.