An Overview of Noise-Robust Automatic Speech Recognition

Jinyu Li, *Member, IEEE*, Li Deng, *Fellow, IEEE*, Yifan Gong, *Senior Member, IEEE*, and Reinhold Haeb-Umbach, *Senior Member, IEEE*

***Abstract*—New waves of consumer-centric applications, such as voice search and voice interaction with mobile devices and home entertainment systems, increasingly require automatic speech recognition (ASR) to be robust to the full range of real-world noise and other acoustic distorting conditions. Despite its practical importance, however, the inherent links between and distinctions among the myriad of methods for noise-robust ASR have yet to be carefully studied in order to advance the field further. To this end, it is critical to establish a solid, consistent, and common mathematical foundation for noise-robust ASR, which is lacking at present. This article is intended to fill this gap and to provide a thorough overview of modern noise-robust techniques for ASR developed over the past 30 years. We emphasize methods that are proven to be successful and that are likely to sustain or expand their future applicability. We distill key insights from our comprehensive overview in this field and take a fresh look at a few old problems, which nevertheless are still highly relevant today. Specifically, we have analyzed and categorized a wide range of noise-robust techniques using five different criteria: 1) feature-domain vs. model-domain processing, 2) the use of prior knowledge about the acoustic environment distortion, 3) the use of explicit environment-distortion models, 4) deterministic vs. uncertainty processing, and 5) the use of acoustic models trained jointly with the same feature enhancement or model adaptation process used in the testing stage. With this taxonomy-oriented review, we equip the reader with the insight to choose among techniques and with the awareness of the performance-complexity tradeoffs. The pros and cons of using different noise-robust ASR techniques in practical application scenarios are provided as a guide to interested practitioners. The current challenges and future research directions in this field is also carefully analyzed.**

***Index Terms*—Speech recognition, noise, robustness, distortion modeling, compensation, uncertainty processing, joint model training .**

I. Introduction

AUTOMATIC speech recognition (ASR) is the process and the related technology for converting the speech signal into its corresponding sequence of words or other linguistic entities by means of algorithms implemented in a device, a computer, or computer clusters [1], [2]. Historically, ASR applications have included voice dialing, call routing, interactive voice response, data entry and dictation, voice command and control, structured document creation (e.g., medical and legal transcriptions), appliance control by voice, computer-aided language learning, content-based spoken audio search, and robotics. More recently, with the exponential growth of big data and computing power, ASR technology has advanced to the stage where more challenging applications are becoming a reality. Examples are voice search and interactions with mobile devices (e.g., Siri on iPhone, Bing voice search on winPhone, and Google Now on Android), voice control in home entertainment systems (e.g., Kinect on xBox), and various speech-centric information processing applications capitalizing on downstream processing of ASR outputs [3]. For such large-scale, real-world applications, noise robustness is becoming an increasingly important core technology since ASR needs to work in much more difficult acoustic environments than in the past [4].

A large number of noise-robust ASR methods, in the order of hundreds, have been proposed and published over the past 30 years or so, and many of them have created significant impact on either research or commercial use. Such accumulated knowledge deserves thorough examination not only to define the state of the art in this field from a fresh and unifying perspective but also to point to fruitful future directions in the field. Nevertheless, a well-organized framework for relating and analyzing these methods is conspicuously missing. The existing survey papers [5]–[13] in noise-robust ASR either do not cover all recent advances in the field or focus only on a specific sub-area. Although there are also few recent books [14], [15], they are collections of topics with each chapter written by different authors and it is hard to provide a unified view across all topics. Given the importance of noise-robust ASR, the time is ripe to analyze and unify the solutions. In this paper, we elaborate on the basic concepts in noise-robust ASR and develop categorization criteria and unifying themes. Specifically, we hierarchically classify the major and significant noise-robust ASR methods using a consistent and unifying mathematical language. We establish their interrelations and differentiate among important techniques, and discuss current technical challenges and future research directions. This paper also identifies relatively promising, short-term new research areas based on a careful analysis of successful methods, which can serve as a reference for future algorithm development in the field. Furthermore, in the literature spanning over 30 years on noise-robust ASR, there is inconsistent use of basic concepts

Manuscript received May 21, 2013; revised October 09, 2013; accepted January 23, 2014. Date of publication February 05, 2014; date of current version February 19, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. DeLiang Wang.

J. Li, L. Deng, and Y. Gong are with Microsoft Corporation, Redmond, WA 98052-6399 USA (e-mail: jinyli@microsoft.com; deng@microsoft.com; ygong@microsoft.com).

R. Haeb-Umbach is with the Department of Communications Engineering, University of Paderborn, 33098 Paderborn, Germany (e-mail: haeb@nt.uni-paderborn.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2014.2304637

TABLE I
DEFINITIONS OF A SUBSET OF COMMONLY USED SYMBOLS AND NOTIONS IN THIS ARTICLE

Symbol	Meaning
T	number of frames in a speech sequence
$\mathbf{X}$	sequence of clean speech vectors $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$
$\mathbf{Y}$	sequence of distorted speech vectors $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T)$
θ	sequence of speech states $(\theta_1, \theta_2, ..., \theta_T)$
$\mathbf{\Lambda}$	acoustic model parameter
$\mathbf{\Gamma}$	language model parameter
$\mathbf{C}$	discrete cosine transform (DCT) matrix
$\mathbf{x}$	clean speech in the cepstral domain
$\mathbf{y}$	distorted speech in the cepstral domain
$\mathbf{n}$	noise in the cepstral domain
$\mathbf{h}$	channel in the cepstral domain
$\mu_{\mathbf{x}}$	clean speech mean in the cepstral domain
$\mu_{\mathbf{y}}$	distorted speech mean in the cepstral domain
$\mu_{\mathbf{n}}$	noise mean in the cepstral domain
$\mu_{\mathbf{h}}$	channel mean in the cepstral domain
$\mathcal{N}$	Gaussian distribution

and terminology as adopted by different researchers in the field. This kind of inconsistency sometimes brings confusion to the field, especially for new researchers and students. It is therefore important to examine discrepancies in the current literature and re-define consistent terminology, another goal of this overview paper.

This paper is organized as follows. In Section II, we discuss the fundamentals of noise-robust ASR. The impact of noise and channel distortions on clean speech is examined. Then, we build a general framework for noise-robust ASR and define five ways of categorizing noise-robust ASR techniques. (This expands the previous taxonomy-oriented review from the use of two criteria to five [10].) Section III is devoted to the first category—feature-domain vs. model-domain techniques. The second category, detailed in Section IV, comprises methods that exploit prior knowledge about the signal distortion. Methods that incorporate an explicit distortion model to predict the distorted speech from a clean one define the third category, covered in Section V. The use of uncertainty constitutes the fourth way to categorize a wide range of noise-robust ASR algorithms, and is covered in Section VI. Uncertainty in either the model space or feature space may be incorporated within the Bayesian framework to promote noise-robust ASR. The final, fifth way to categorize and analyze noise-robust ASR techniques utilizes joint model training, described in Section VII. With joint model training, environmental variability in the training data is removed in order to generate canonical models. We conclude this overview paper in Section VIII, with discussions on future directions for noise-robust ASR.

II. THE BACKGROUND AND SCOPE

In this section, we establish some fundamental concepts that are most relevant to the discussions in the remainder of this paper. Since mathematical language is an essential tool in our exposition, we first introduce our notation in Table I. Throughout this paper, vectors are in bold type and matrices are capitalized.

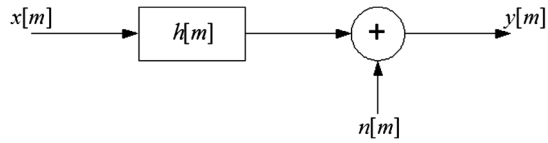


Fig. 1. A model of acoustic environment distortion in the discrete-time domain relating clean speech samples $x[m]$ to distorted speech samples $y[m]$.

A. Modeling Distortions of Speech in Acoustic Environments

Mel-frequency cepstral coefficients (MFCCs) [16] are the most widely used acoustic features. The short-time discrete Fourier transform (STDFT) is applied to the speech signal, and the power or magnitude spectrum is generated. A set of Mel scale filters is applied to obtain Mel-filter-bank output. Then the log operator is used to get the log-filter-bank output. Finally, the discrete cosine transform (DCT) is used to generate MFCCs. In the following, we use MFCCs as the acoustic feature to elaborate on the relation between clean and distorted speech.

Figure 1 shows a time domain model for speech degraded by both additive noise and convolutive channel distortions [5]. The observed distorted speech signal $y[m]$, where m denotes the discrete time index, is generated from the clean speech signal $x[m]$ with additive noise $n[m]$ and convolutive channel distortions $h[m]$ according to

$$y[m] = x[m] * h[m] + n[m], \tag{1}$$

where ' $*$' denotes the convolution operator.

After applying the STDFT, the following equivalent relation can be established in the spectral domain:

$$\dot{y}[k] = \dot{x}[k]\dot{h}[k] + \dot{n}[k]. \tag{2}$$

Here, k is the frequency bin index. Note that we left out the frame index for ease of notation. To arrive at Eq. (2) we assumed that the impulse response $h[m]$ is much shorter than the DFT analysis window. Then we can make use of the multiplicative transfer function approximation by which a convolution in the time domain corresponds to a multiplication in the STDFT domain [17]. This approximation does not hold in the presence of reverberated speech, because the acoustic impulse response characterizing the reverberation is typically much longer than the STDFT window size. Thus Eq. (2) is not adequate to describe reverberated speech in the STDFT domain.

The power spectrum of the distorted speech can then be obtained as:

$$\begin{aligned} |\dot{y}[k]|^2 = {} & |\dot{x}[k]|^2|\dot{h}[k]|^2 \\ & + |\dot{n}[k]|^2 + 2|\dot{x}[k]||\dot{h}[k]||\dot{n}[k]|\cos\beta_k, \end{aligned} \tag{3}$$

where β_k denotes the (random) angle between the two complex variables $\dot{n}[k]$ and $\dot{x}[k]\dot{h}[k]$. If $\cos\beta_k$ is set as 0, Eq. (3) will become:

$$|\dot{y}[k]|^2 = |\dot{x}[k]|^2|\dot{h}[k]|^2 + |\dot{n}[k]|^2. \tag{4}$$

Removing this "phase" term is a common practice in the formulation of speech distortion in the power spectral domain; e.g. in the spectral subtraction technique. So is approximating

the phase term in the log-spectral domain; e.g. [18]. While achieving simplicity in developing speech enhancement algorithms, removing this term is a partial cause of the degradation of enhancement performance at low SNRs (around 0 dB) [19].

By applying a set of Mel-scale filters (L in total) to the power spectrum in Eq. (3), we have the l-th Mel-filter-bank energies for distorted speech, clean speech, noise and channel:

$$|\breve{y}[l]|^2 = \sum_k W_k[l]|\dot{y}[k]|^2 \tag{5}$$

$$|\breve{x}[l]|^2 = \sum_k W_k[l]|\dot{x}[k]|^2 \tag{6}$$

$$|\breve{n}[l]|^2 = \sum_k W_k[l]|\dot{n}[k]|^2 \tag{7}$$

$$|\breve{h}[l]|^2 = \frac{\sum_k W_k[l]|\dot{x}[k]|^2|\dot{h}[k]|^2}{|\breve{x}[l]|^2} \tag{8}$$

where the l-th filter is characterized by the transfer function $W_k[l] \geq 0$ with $\sum_k W_k[l] = 1$.

The phase factor $\alpha[l]$ of the l-th Mel-filter-bank is [19]

$$\alpha[l] = \frac{\sum_k W_k[l]|\dot{x}[k]||\dot{h}[k]||\dot{n}[k]|\cos\beta_k}{|\breve{x}[l]||\breve{h}[l]||\breve{n}[l]|} \tag{9}$$

Then, the following relation is obtained in the Mel-filter-bank domain for the l-th Mel-filter-bank output

$$\begin{aligned}|\breve{y}[l]|^2 = {} & |\breve{x}[l]|^2|\breve{h}[l]|^2 + |\breve{n}[l]|^2 \\ & + 2\alpha[l]|\breve{x}[l]||\breve{h}[l]||\breve{n}[l]|,\end{aligned} \tag{10}$$

By taking the log operation in both sides of Eq. (10), we have the following in the log-Mel-filter-bank domain, using vector notation

$$\begin{aligned}\tilde{\mathbf{y}} = \tilde{\mathbf{x}} + \tilde{\mathbf{h}} + \log\Big(1 + \exp(\tilde{\mathbf{n}} - \tilde{\mathbf{x}} - \tilde{\mathbf{h}}) \\ + 2\boldsymbol{\alpha}.*\exp(\frac{\tilde{\mathbf{n}} - \tilde{\mathbf{x}} - \tilde{\mathbf{h}}}{2})\Big).\end{aligned} \tag{11}$$

The $.*$ operation for two vectors denotes element-wise product, and taking the logarithm and exponentiation of a vector above are also element-wise operations.

By applying the DCT transform to Eq. (11), we can get the distortion formulation in the cepstral domain as

$$\begin{aligned}\mathbf{y} = \mathbf{x} + \mathbf{h} + \mathbf{C}\log\big(1 + \exp(\mathbf{C}^{-1}(\mathbf{n} - \mathbf{x} - \mathbf{h})) \\ + 2\boldsymbol{\alpha}.*\exp(\mathbf{C}^{-1}\frac{\mathbf{n} - \mathbf{x} - \mathbf{h}}{2})\big),\end{aligned} \tag{12}$$

where $\mathbf{C}$ denotes the DCT matrix.

In [19], it was shown that the phase factor $\alpha[l]$ for each Mel-filter l can be approximated by a weighted sum of a number of independent zero-mean random variables distributed over $[-1, 1]$, where the total number of terms equals the number of DFT bins. When the number of terms becomes large, the central limit theorem postulates that $\alpha[l]$ will be approximately Gaussian. A more precise statistical description has been developed in [20], where it is shown that all moments of $\alpha[l]$ of odd order are zero.

If we ignore the phase factor, Eq. (11) and Eq. (12) can be simplified to

$$\tilde{\mathbf{y}} = \tilde{\mathbf{x}} + \tilde{\mathbf{h}} + \log(1 + \exp(\tilde{\mathbf{n}} - \tilde{\mathbf{x}} - \tilde{\mathbf{h}})), \tag{13}$$

$$\mathbf{y} = \mathbf{x} + \mathbf{h} + \mathbf{C}\log(1 + \exp(\mathbf{C}^{-1}(\mathbf{n} - \mathbf{x} - \mathbf{h}))), \tag{14}$$

which are the log-Mel-filter-bank and cepstral representations, respectively, corresponding to Eq. (4) in the power spectral domain. Eq. 13 and Eq. 14 are widely used in noise-robust ASR technology as the basic formulation that characterizes the relationship between clean and distorted speech in the logarithmic domain. The effect of the phase factor is small if noise estimates are poor. However, with an increase in the quality of the noise estimates, the effect of the phase factor is shown experimentally to be stronger [19].

In ASR, Gaussian mixture models (GMMs) are widely used to characterize the distribution of speech in the log-Mel-filter-bank or cepstral domain. It is important to understand the impact of noise, which is additive in the spectral domain, on the distribution of noisy speech in the log-Mel-filter-bank and cepstral domains. Using Eq. 13 while setting $\tilde{\mathbf{h}} = 0$ for simplicity, we can simulate noisy speech in the log-filter-bank domain. In Figure 2, we show the impact of noise on the clean speech signal in the log-filter-bank domain with increasing noise mean values, i.e., decreasing SNRs. The clean speech shown with solid lines is Gaussian distributed, with a mean value of 25 and a standard deviation of 10. The noise $\tilde{\mathbf{n}}$ is also Gaussian distributed, with different mean values and a standard deviation of 2. The noisy speech shown with dashed lines deviates from the Gaussian distribution to a varying degree. We can use a Gaussian distribution, shown with dotted lines, to make an approximation. The approximation error is large in the low SNR cases. When the noise mean is raised to 20 and 25, as in Figure 2(c) and 2(d), the distribution of noisy speech is skewed far away from a Gaussian distribution.

A natural way to deal with noise in the acoustic environment is to use multi-style training [21], which trains the acoustic model with all available noisy speech data. The hope is that one of the noise types in the training set will appear in the deployment scenario. However, there are two major problems with multi-style training. The first is that during training it is hard to enumerate all noise types and SNRs encountered in test environments. The second is that the model trained with multi-style training has a very broad distribution because it needs to model all the environments. Given the unsatisfactory behavior of multi-style training, it is necessary to work on technologies that directly deal with the noise and channel impacts. In the next section, we lay down a general mathematical framework for noise-robust speech recognition.

B. A General Framework for Noise-Robust Speech Recognition

The goal of ASR is to obtain the optimal word sequence $\mathbf{W}$, given the spoken speech signal $\mathbf{X}$, which can be formulated as the well-known maximum *a posteriori* (MAP) problem:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{X}), \tag{15}$$

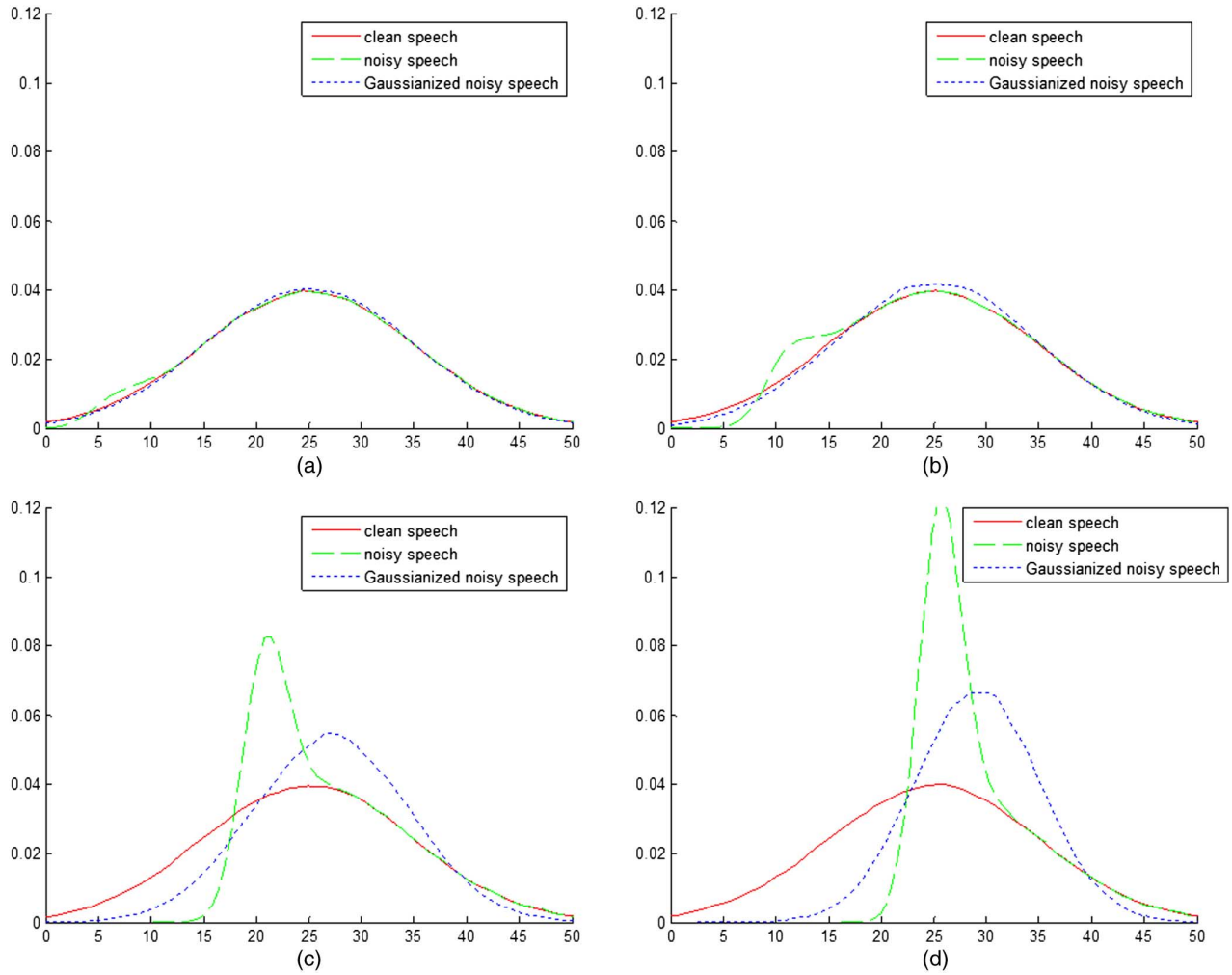


Fig. 2. The impact of noise, with varying mean values from 5 in (a) to 25 in (d), in the log-filter-bank domain. The clean speech has a mean value of 25 and a standard deviation of 10. The noise has a standard deviation of 2.

where Λ and Γ are the acoustic model (AM) and language model (LM) parameters. Using Bayes' rule

$$P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{X}) = \frac{p_\Lambda(\mathbf{X}|\mathbf{W})P_\Gamma(\mathbf{W})}{p(\mathbf{X})}, \tag{16}$$

Eq. (15) can be re-written as:

$$\hat{W} = \arg\max_{\mathbf{W}} p_\Lambda(\mathbf{X}|\mathbf{W})P_\Gamma(\mathbf{W}), \tag{17}$$

where $p_\Lambda(\mathbf{X}|\mathbf{W})$ is the AM likelihood and $P_\Gamma(\mathbf{W})$ is the LM probability. When the time sequence is expanded and the observations $\mathbf{x}_t$ are assumed to be generated by hidden Markov models (HMMs) with hidden states θ_t, we have

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P_\Gamma(\mathbf{W}) \sum_\theta \prod_{t=1}^{T} p_\Lambda(\mathbf{x}_t|\theta_t)P_\Lambda(\theta_t|\theta_{t-1}), \tag{18}$$

where θ belongs to the set of all possible state sequences for the transcription W.

When the noisy speech Y is presented, the decision rule becomes

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{Y}). \tag{19}$$

Introducing clean speech as a hidden variable, we have

$$\begin{aligned} P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{Y}) &= \int P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{X},\mathbf{Y})p(\mathbf{X}|\mathbf{Y})d\mathbf{X} \\ &= \int P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{X})p(\mathbf{X}|\mathbf{Y})d\mathbf{X}. \end{aligned} \tag{20}$$

In Eq. (20) we exploited the fact that the distorted speech signal $\mathbf{Y}$ doesn't deliver additional information if the clean speech signal $\mathbf{X}$ is given. With Eq. (16), $P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{Y})$ can be re-written as

$$P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{Y}) = P_\Gamma(\mathbf{W}) \int \frac{p_\Lambda(\mathbf{X}|\mathbf{W})}{p(\mathbf{X})} p(\mathbf{X}|\mathbf{Y})d\mathbf{X}. \tag{21}$$

Note that

$$p_\Lambda(\mathbf{X}|\mathbf{W}) = \sum_\theta p_\Lambda(\mathbf{X}|\theta,\mathbf{W})P_\Lambda(\theta|\mathbf{W}) \tag{22}$$

and then Eq. (21) becomes

$$P_\Gamma(\mathbf{W}) \int \frac{\sum_\theta p_\Lambda(\mathbf{X}|\theta,\mathbf{W})P_\Lambda(\theta|\mathbf{W})}{p(\mathbf{X})} p(\mathbf{X}|\mathbf{Y})d\mathbf{X}. \tag{23}$$

Under some mild assumptions this can be simplified to [11], [22]

$$P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{Y}) = P_{\Gamma}(\mathbf{W}) \sum_{\theta} \prod_{t=1}^{T} \int \frac{p_{\Lambda}(\mathbf{x}_t|\theta_t) p(\mathbf{x}_t|\mathbf{Y})}{p(\mathbf{x}_t)} d\mathbf{x}_t \; P_{\Lambda}(\theta_t|\theta_{t-1}). \quad (24)$$

One key component in Eq. (24) is $p(\mathbf{x}_t|\mathbf{Y})$, the clean speech's posterior given noisy speech $\mathbf{Y}$. In principle, it is computed via

$$p(\mathbf{x}_t|\mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{x}_t) p(\mathbf{x}_t), \quad (25)$$

i.e., employing an *a priori* model $p(\mathbf{x}_t)$ of clean speech and an observation model $p(\mathbf{Y}|\mathbf{x}_t)$, which relates the clean to the noisy speech features. Noise robustness techniques may be categorized by the kind of observation models used, also according to whether an explicit or an implicit distortion model is used, and according to whether or not prior knowledge about distortion is employed to learn the relationship between $\mathbf{x}_t$ and $\mathbf{y}_t$, as we will further develop in later sections of the article.

For simplicity, many noise-robust ASR techniques use a point estimate. That is, the back-end recognizer considers the cleaned or denoised signal $\hat{\mathbf{x}}_t(\mathbf{Y})$ as an estimate without uncertainty:

$$p(\mathbf{x}_t|\mathbf{Y}) = \delta(\mathbf{x}_t - \hat{\mathbf{x}}_t(\mathbf{Y})), \quad (26)$$

where $\delta(.)$ is a Kronecker delta function. Then, Eq. (24) is reduced to

$$P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{Y}) = P_{\Gamma}(\mathbf{W}) \sum_{\theta} \prod_{t=1}^{T} \frac{p_{\Lambda}(\hat{\mathbf{x}}_t(\mathbf{Y})|\theta_t)}{p(\hat{\mathbf{x}}_t(\mathbf{Y}))} P_{\Lambda}(\theta_t|\theta_{t-1}). \quad (27)$$

Because the denominator, $p(\hat{\mathbf{x}}_t(\mathbf{Y}))$, is independent of the underlying word sequence, the decision is further reduced to

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P_{\Gamma}(\mathbf{W}) \sum_{\theta} \prod_{t=1}^{T} p_{\Lambda}(\hat{\mathbf{x}}_t(\mathbf{Y})|\theta_t) P_{\Lambda}(\theta_t|\theta_{t-1}). \quad (28)$$

Eq. (28) is the formulation most commonly used. Comparing with Eq. (18), the only difference is that. $\hat{\mathbf{x}}_t(\mathbf{Y})$. is used to replace $\mathbf{x}_t$. In feature processing methods, only the distorted feature $\mathbf{y}_t$ is enhanced with $\hat{\mathbf{x}}_t(\mathbf{Y})$, without changing the acoustic model parameter, Λ.

In contrast, there is another major category of model-domain processing methods, which adapt model parameters to fit the distorted speech signal

$$\hat{\Lambda} = \mathcal{F}(\Lambda, \mathbf{Y}) \quad (29)$$

and in this case the posterior used in the MAP decision rule is computed using

$$P_{\hat{\Lambda},\Gamma}(\mathbf{W}|\mathbf{Y}) = P_{\Gamma}(\mathbf{W}) \sum_{\theta} \prod_{t=1}^{T} p_{\hat{\Lambda}}(\mathbf{y}_t|\theta_t) P_{\hat{\Lambda}}(\theta_t|\theta_{t-1}). \quad (30)$$

C. Five Ways of Categorizing and Analyzing Noise-Robust ASR Techniques: An Overview

The main theme of this article is to provide insights from multiple perspectives in organizing a multitude of noise-robust ASR techniques. Based on the general framework in this section, we provide a comprehensive overview, in a mathematically rigorous and unified manner, of noise-robust ASR using five different ways of categorizing, analyzing, and characterizing major existing techniques. The categorization is based on the following key attributes of the algorithms in our review:

1) Feature-Domain vs. Model-Domain Compensation: The acoustic mismatch between training and testing conditions can be viewed from either the feature domain or the model domain, and noise or distortion can be compensated for in either space. Some methods are formulated in both the feature and model domains, and can thus be categorized as "hybrid". Feature-space approaches usually do not change the parameters of acoustic models. Most feature-space methods use Eq. (28) to compute the posterior $P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{Y})$ after "plugging in" the enhanced signal $\hat{\mathbf{x}}_t(\mathbf{Y})$. On the other hand, model-domain methods modify the acoustic model parameters with Eq. (29) to incorporate the effects of the distortion, as in Eq. (30). In contrast with feature-space methods, the model-domain methods are closely linked with the objective function of acoustic modeling. While typically achieving higher accuracy than feature-domain methods, they usually incur significantly larger computational costs. We will discuss both the feature- and model-domain methods in detail in Sections III-A and III-B. Specifically, noise-resistant features, feature moment normalization, and feature compensation methods are presented in III-A1, III-A2, and III-A3, respectively.

2) Compensation Using Prior Knowledge about Acoustic Distortion: This axis for categorizing and analyzing noise-robust ASR techniques examines whether the method exploits prior knowledge about the distortion. Details follow in Section IV. Some of these methods, discussed in Section IV-A, learn the mapping between clean and noisy speech features when they are available as a pair of "stereo" data. During decoding, with the pre-learned mapping, the clean speech feature $\hat{\mathbf{x}}_t(\mathbf{Y})$ can be estimated and plugged into Eq. (28) to decode the word sequence. Another method presented in Section IV-B builds multiple models or dictionaries of speech and noise from multi-environment data. Some examples discussed in IV-B1 collect and learn a set of models first, each corresponding to one specified environment in training. These pre-trained models are then combined online to form a new model $\hat{\Lambda}$ that fits the test environment best. The methods described in section IV-B2 are usually based on source separation—they build clean speech and noise exemplars from training data, and then reconstruct the speech signal $\hat{\mathbf{x}}_t(\mathbf{Y})$ only from the exemplars of clean speech. With variable-parameter HMM methods, examined in IV-B3, the acoustic model parameters or transforms are polynomial functions of an environment variable.

3) Compensation with Explicit vs. Implicit Distortion Modeling: To adapt the model parameters in Eq. (29), general technologies make use of a set of linear transformations to compensate for the mismatch between training and testing conditions. This involves many parameters and thus typically requires a large amount of data for estimation. This difficulty can be overcome when exploiting an explicit distortion model which takes into account the way in which distorted speech features are produced. That is, the distorted speech features are

represented using a nonlinear function of clean speech features, additive noise, and convolutive distortion. This type of physical model enables structured transformations to be used, which are generally nonlinear and involve only a parsimonious set of free parameters to be estimated. We refer to a noise-robust method as an explicit distortion modeling one when a physical model for the generation of distorted speech features is employed. If no physical model is explicitly used, the method will be referred to as an implicit distortion modeling method. Since physical constraints are modeled, the explicit distortion modeling methods exhibit high performance and require a relatively small number of distortion parameters to be estimated. Explicit distortion models can also be applied to feature processing. With the guide of explicit modeling, the enhancement of speech often becomes more effective. Noise-robust ASR techniques with explicit distortion modeling will be explored in Section V. In particular, parallel model combination is briefly described in Section V-A, and vector Taylor series (VTS) is presented in Section V-B, along with the details of VTS model adaption, distortion estimation, VTS feature enhancement, and recent improvements. Finally, sampling-based methods, such as data-driven PMC and the unscented transform, are examined in Section V-C.

4) Compensation with Deterministic vs. Uncertainty Processing: Most noise-robust ASR methods use a deterministic strategy; i.e., the compensated feature is a point estimate from the corrupted speech feature with Eq. (26), or the compensated model is a point estimate as adapted from the clean speech model with Eq. (29). We refer to methods in this category as deterministic processing methods. However, strong noise and unreliably decoded transcriptions necessarily create inherent uncertainty in either the feature or the model space, which should be accounted for in MAP decoding. When a noise-robust method takes that uncertainty into consideration, we call it an uncertainty processing method. In the feature space, the presence of noise brings uncertainty to the enhanced speech signal, which is modeled as a distribution instead of a deterministic value. In the general case, $p(\mathbf{x}_t|\mathbf{Y})$ in Eq. (24) is not a Kronecker delta function and there is uncertainty in the estimate of $\hat{\mathbf{x}}_t$ given $\mathbf{Y}$. Uncertainty can also be introduced in the model space when assuming the true model parameters are in a neighborhood of the trained model parameters Λ, or compensated model parameters $\hat{\Lambda}$. We will study uncertainty methods in feature and model spaces in Sections VI-B and VI-A, respectively. Then joint uncertainty decoding is described in Section VI-C and missing feature approaches are discussed in Section VI-D.

5) Disjoint vs. Joint Model Training: Finally, we can categorize most existing noise-robust techniques in the literature into two broad classes depending on whether or not the acoustic model, Λ, is trained jointly with the same process of feature enhancement or model adaptation used in the test stage. Among the joint model training methods, the most prominent set of techniques are based on a paradigm called noise adaptive training (NAT) which applies consistent processing during the training and testing phases while eliminating any residual mismatch in an otherwise disjoint training paradigm. Further developments of NAT include joint training of a canonical acoustic model and a set of transforms under maximum likelihood estimation or a discriminative training criterion. In Section VII, these methods will be examined in detail.

Note that the chosen categories discussed above are by no means orthogonal. While it may be ambiguous under which category a particular noise-robustness approach would fit the best, we have used our best judgement with a balanced view.

D. Standard Evaluation Database

In the early years of developing noise-robust ASR technologies, it was very hard to conclude which technology was better since different groups used different databases for evaluation. The introduction of a standard evaluation database and training recipes finally allowed noise-robustness methods developed by different groups to be compared fairly using the same task, thereby fast-tracking development of these methods. Among the standard evaluation databases, the most famous tasks are the Aurora series developed by the European Telecommunications Standards Institute (ETSI), although there are some other tasks such as Noisex-92 [23], SPINE (SPeech In Noisy Environments) [24], and the recently developed CHiME (Computational Hearing in Multisource Environments) task [25].

The first Aurora database is Aurora 2 [26], a task of recognizing digit strings in noise and channel distorted environments. The evaluation data is artificially corrupted. The Aurora 3 task consists of noisy speech data recorded inside cars as part of the SpeechDatCar project [27]. Although still a digit recognition task, the utterances in Aurora 3 are collected in real noisy environments. The Aurora 4 task [28] is a standard large vocabulary continuous speech recognition (LVCSR) task which is constructed by artificially corrupting the clean data from the Wall Street Journal (WSJ) corpus [29]. Aurora 5 [30] was mainly developed to investigate the influence of hands-free speech input on the performance of digit recognition in noisy room environments and over a cellular telephone network. The evaluation data is artificially simulated. The progression of the Aurora tasks after Aurora 2 show a clear trend: from real noisy environments (Aurora 3), to a LVCSR task (Aurora 4), to working in the popular cellular scenario (Aurora 5). This is consistent with the need to develop noise-robust ASR technologies for real-world deployment.

E. The Scope of This Overview

Noise robustness in ASR is a vast topic, spanning research literature over 30 years. In developing this overview, we necessarily have to limit its scope. In particular,

- we only consider single-channel input, thus leaving out the topics of acoustic beamforming, multi-channel speech enhancement and source separation;
- we assume that the noise can be considered more stationary than speech, thus disregarding for the most part the recognition of speech in the presence of music or other competing speakers;
- we assume that the channel impulse response is much shorter than the frame size; i.e., we do not consider the case of reverberation, but rather convolutional channel distortions caused by, e.g., different microphone characteristics.

Readers interested in the topic of multi-channel speech processing are referred to recent books in this field, which provide overview articles on acoustic beamforming, multi-channel speech enhancement, source separation and speech dereverberation [15], [31]–[33]. Recognition of reverberant speech is covered by the book of Woelfel and McDonough [34] and tutorial articles of more recent developments are [35], [36]. Further, [37] provides an overview of speech separation in the presence of non-stationary distortions. A source of many good tutorial articles on recent developments in automatic speech recognition is [38]. Some specific techniques pertaining to the above topics not covered in this article can be found in [39] for multi-sensory speech detection, in [40], [41] for nonstationary noise estimation and ASR robustness against nonstationary noise, in [42] for a new multichannel framework for speech source separation and noise reduction, in [43], [44] for robustness against reverberation, and in [45] for speech-music separation.

What remains is still a huge field of research. We hope that despite the limited scope the reader will find this overview useful.

III. Feature-Domain and Model-Domain Methods

Feature-space approaches usually do not change the parameters of the acoustic model (e.g., HMMs). They either rely on auditory features that are inherently robust to noise or modify the test features to match the training features. Because they are not related to the back-end, usually the computational cost of these methods is low. In contrast, model-domain methods modify the acoustic model parameters to incorporate the effects of noise. While typically achieving higher accuracy than feature-domain methods, they usually incur significantly larger computational cost.

A. Feature-Space Approaches

Feature-space methods can be classified further into three sub-categories:

- noise-resistant features, where robust signal processing is employed to reduce the sensitivity of the speech features to environment conditions that don't match those used to train the acoustic model;
- feature normalization, where the statistical moments of speech features are normalized; and
- feature compensation, where the effects of noise embedded in the observed speech features are removed.

1) Noise-Resistant Features: Noise-resistant feature methods focus on the effect of noise rather than on the removal of noise. One of the advantages of these techniques is that they make only weak or no assumptions about the noise. In general, no explicit estimation of the noise statistics is required. On the other hand, this can be a shortcoming since it is impossible to make full use of the characteristics specific to a particular noise type.

a) Auditory-Based Feature: Perceptually based linear prediction (PLP) [46], [47] filters the speech signal with a Bark-scale filter-bank. The output is converted into an equal-loudness representation. The resulting auditory spectrum is then modeled by an all-pole model. A cepstral analysis can also be performed.

Many types of additive noise as well as most channel distortions vary slowly compared to the variations in speech signals. Filters that remove variations in the signal that are uncharacteristic of speech (including components with both slow and fast modulation frequencies) improve the recognition accuracy significantly [48]. Relative spectral processing (RASTA) [49], [50] consists of suppressing constant additive offsets in each log spectral component of the short-term auditory-like spectrum. This analysis method can be applied to PLP parameters, resulting in RASTA-PLP [49], [50]. Each frequency band is filtered by a noncausal infinite impulse response (IIR) filter that combines both high- and low-pass filtering. Assuming a frame rate of 100 Hz, the transfer function

$$H(z) = 0.1z^4 \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}} \tag{31}$$

yields a spectral zero at zero modulation frequency and a pass-band approximately from 1 to 12 Hz. While the design of the IIR-format RASTA filter in Eq. (31) is based on auditory knowledge, the RASTA filter can also be designed as a linear finite impulse response (FIR) filter in a data-driven way using technology such as linear discriminant analysis (LDA) [51].

There are plenty of other auditory-based feature extraction methods, such as zero crossing peak amplitude (ZCPA) [52], average localized synchrony detection (ALSD) [53], perceptual minimum variance distortionless response (PMVDR) [54], power-normalized cepstral coefficients (PNCC) [55], invariant-integration features (IIF) [56], amplitude modulation spectrogram [57], Gammatone frequency cepstral coefficients [58], sparse auditory reproducing kernel (SPARK) [59], and Gabor filter bank features [60], to name a few. [61] provides a relatively complete review on auditory-based features. All these methods are designed by utilizing some auditory knowledge. However, there is no universally-accepted theory about which kind of auditory information is most important to robust speech recognition. Therefore, it is hard to argue which one in theory is better than another.

Since there is no universally-accepted auditory theory for robust speech recognition, it is sometimes very hard to set the right parameter values in auditory methods. Some parameters can be learned from data [62], but this may not always be the case. Although the auditory-based features can usually achieve better performance than MFCC, they have a much more complicated generation process which sometimes prevents them from being widely used together with some noise-robustness technologies. For example, in Section II-A, the relation between clean and noisy speech for MFCC features can be derived as Eq. (12). However, it is very hard to derive such a relation for auditory-based features. As a result, MFCC is widely used as the acoustic feature for methods with explicit distortion modeling.

b) Neural Network Approaches: Artificial neural network (ANN) based methods have a long history of providing effective features for ASR. For example, ANN-HMM hybrid systems [63] replace the GMM acoustic model with an ANN when evaluating the likelihood score. The ANNs used before 2009 usually had the multi-layer perceptron (MLP) structure with one hidden layer. Hybrid systems have been shown to have comparable performance to GMM-based systems.

The TANDEM system was later proposed in [64] to combine ANN discriminative feature processing with a GMM, and it demonstrated strong performance on the Aurora 2 noisy contin-

uous digit recognition task. Instead of using the posterior vector for decoding as in the hybrid system, the TANDEM system omits the final nonlinearity in the ANN output layer and applies a global decorrelation to generate a new set of features used to train a GMM system. One reason the TANDEM system has very good performance on noise-robust tasks is that the ANN has the modeling power in small regions of feature space that lie on phone boundaries [64]. Another reason is due to the nonlinear modeling power of the ANN, which can normalize data from different sources well.

Another way to obtain probabilistic features is TempoRAL Pattern (TRAP) processing [65], which captures the appropriate temporal pattern with a long temporal vector of log-spectral energies from a single frequency band. One main reason for the noise-robustness of TRAP is the ability to handle band-specific processing [66]. Even if one band of speech is polluted by noise, the phoneme classifier in another band can still work very well. TRAP processing works on temporal patterns, differing from the conventional spectral feature vector. Hence, TRAP can be combined very well with MFCC or PLP features to further boost performance [67].

Building upon the above-mentioned methods, bottle-neck (BN) features [68] were developed as a new method to use ANNs for feature extraction. A five-layer MLP with a narrow layer in the middle (bottle-neck) is used to extract BN features. The fundamental difference between TANDEM and BN features is that the latter are not derived from the posterior vector. Instead, they are obtained as linear outputs of the bottle-neck layer. Principal component analysis (PCA) or heteroscedastic linear discriminant analysis (HLDA) [69] is used to decorrelate the BN features which then become inputs to the GMM-HMM system. Although current research of BN features is not focused on noise robustness, it has been shown that BN features outperform TANDEM features on some LVCSR tasks [68], [70]. Therefore, it is also possible that BN features can perform well on noise-robust tasks.

More recently, a new acoustic model, referred to as the context-dependent deep neural network hidden Markov model (CD-DNN-HMM), has been developed. It has been shown, by many groups [71]–[75], to outperform the conventional GMM-HMMs in many ASR tasks. The CD-DNN-HMM is also a hybrid system. There are three key components in the CD-DNN-HMM: modeling senones (tied states) directly even though there might be thousands or even tens of thousands of senones; using deep instead of shallow multi-layer perceptrons; and using a long context window of frames as the input. These components are critical for achieving the huge accuracy improvements reported in [71], [73], [76]. Although the conventional ANN in TANDEM also takes a long context window as the input, the key to success of the CD-DNN-HMM is due to the combination of these components. With the excellent modeling power of the DNN, in [77] it is shown that DNN-based acoustic models can easily match state-of-the-art performance on the Aurora 4 task [28], which is a standard noise-robustness LVCSR task, without any explicit noise compensation. The CD-DNN-HMM is expected to make further progress on noise-robust ASR due to the DNN's ability to handle heterogeneous data [77], [78]. Although the CD-DNN-HMM is a modeling technology, its layer-by-layer setup provides a feature extraction strategy that automatically derives powerful noise-resistant features from primitive raw data for senone classification. In addition to using the standard feed-forward structure of DNNs, recurrent neural networks (RNN) that model temporal signal dependence in an explicit way have also been exploited for noise-robust ASR, either in modeling the posterior [79] or predicting clean speech from noisy speech [80].

2) Feature Moment Normalization: Feature moment normalization methods normalize the statistical moments of speech features. Cepstral mean normalization (CMN) [81] and cepstral mean and variance normalization (CMVN) [82] normalize the first and second order statistical moments, respectively, while histogram equalization (HEQ) [83] normalizes the higher order statistical moments through the feature histogram.

a) Cepstral Mean Normalization: Cepstral mean normalization (CMN) [81] is the simplest feature moment normalization technique. Given a sequence of cepstral vectors $[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{T-1}]$, CMN subtracts the mean value $\boldsymbol{\mu}_x$ from each cepstral vector $\mathbf{x}_t$ to obtain the normalized cepstral vector $\hat{\mathbf{x}}_t$. After normalization, the mean of the cepstral sequence is 0. It is easy to show that, in absence of noise, the convolutive channel distortion in the time domain has an additive effect in the log-Mel-filter-bank domain and the cepstral domain. Therefore, CMN is good at removing the channel distortion. It is also shown in [9] that CMN can help to improve recognition in noisy environments even if there is no channel distortion.

Instead of using a single mean for the whole utterance, CMN can be extended to use multi-class normalization. Better performance is obtained with augmented CMN [84], where speech and silence frames are normalized to their own reference means rather than a global mean.

For real-time applications, CMN is unacceptable because the mean value is calculated using the features in the whole utterance. Hence, it needs to be modified for deployment in a real-time system. CMN can be considered as a high-pass filter with a cutoff frequency that is arbitrarily close to 0 Hz [1]. Following this interpretation, it is reasonable to use other types of high-pass filters to approximate CMN. A widely used one is a first-order recursive filter, in which the cepstral mean is a function of time according to

$$\boldsymbol{\mu}_{x_t} = \alpha \mathbf{x}_t + (1-\alpha)\boldsymbol{\mu}_{x_{t-1}}, \tag{32}$$

$$\hat{\mathbf{x}}_t = \mathbf{x}_t - \boldsymbol{\mu}_{x_t}, \tag{33}$$

where α is chosen in the way that the filter has a time constant of at least 5 seconds of speech [1]. Other types of filters can also be used. For example, the band-pass IIR filter of RASTA shown in Eq. (31) performs similarly to CMN [85]. Its high-pass portion of the filter is used to compensate for channel convolution effects as with CMN, while its low-pass portion helps to smooth some of the fast frame-to-frame spectral changes which should not exist in speech.

b) Cepstral Mean and Variance Normalization: Cepstral mean and variance normalization (CMVN) [82] normalizes the mean and covariance together. After normalization, the sample

mean and variance of the cepstral sequence are 0 and 1, respectively. CMVN has been shown to outperform CMN in noisy test conditions. [9] gives a detailed comparison of CMN and CMVN, and discusses different strategies to apply them. [86] proposes a method is to combine mean subtraction, variance normalization, and ARMA filtering (MVA) post-processing together. An analysis showing why MVA works well is also presented in [86].

Although mean normalization is directly related to removing channel distortion, variance normalization cannot be easily associated with removing any distortion explicitly. Instead, CMN and CMVN can be considered as ways to reduce the first- and second-order moment mismatches between the training and testing conditions. In this way, the distortion brought by additive noise and a convolutive channel can be reduced to some extent. As an extension, third-order [87] or even higher-order [88] moment normalization can be used to further improve noise robustness. Multi-class extensions can also be applied to CMVN to further improve robustness [89].

c) Histogram Equalization: A natural extension to the moments normalization techniques is to normalize the distribution between training and testing data. This normalizes all of the moments in the speech-feature distribution. This approach is called histogram equalization method (HEQ) [83], [90]–[92]. HEQ postulates that the transformed speech-feature distributions of the training and test data are the same. Each feature vector dimension is normalized independently. HEQ can be applied either in the Mel-filter-bank domain [92], [93] or the cepstral domain [91], [94].

The following transformation function $f(.)$ is applied to the test feature y:

$$f(y) = C_x^{-1}(C_y(y)), \tag{34}$$

where $C_y(.)$ is the cumulative distribution function (CDF) of the test data, and $C_x^{-1}(.)$ is the inverse CDF of the training data. Afterwards, the transformed test feature will have the distribution of the training data. In this way, HEQ reduces the systematic statistical mismatch between test and training data.

While the underlying principle is rather straightforward, the problem is how to reliably estimate CDFs. When a large amount of data is available, the CDFs can be accurately approximated by the cumulative histogram. Such approximations become unreliable for short test utterances. Order-statistics based methods tend to be more accurate and reliable when there is an insufficient amount of data [91], [94].

There are several implementation methods for HEQ. Table-based HEQ (THEQ) is a popular method [90] that uses a cumulative histogram to estimate the corresponding CDF value of the feature vector elements. In THEQ, a look-up table is used as the implementation C_x^{-1} in Eq. (34). This requires that all of the look-up tables in every feature dimension are kept in memory, causing a large deployment cost that applications with limited resources may find unaffordable. Also, the testing CDF is not as reliable as the training CDF because limited data is available for the estimation. Therefore, several methods are proposed to work with only limited test data, such as quantile-based HEQ (QHEQ) [95] and polynomial-fit HEQ (PHEQ) [96]. Instead of fully matching the training and test CDF, QHEQ calibrates the test CDF to the training CDF in a quantile-corrective manner. To achieve this goal, it uses a transformation function which is estimated by minimizing the mismatch between the quantiles of the test and training data. In PHEQ, a polynomial function is used to fit C_x^{-1}. The polynomial coefficients are learned by minimizing the squared error between the input feature and the approximated feature for all the training data.

One HEQ assumption is that the distributions of acoustic classes (e.g., phones) should be identical or similar for both training and test data. However, a test utterance is usually too short for the acoustic class distribution to be similar enough to the training distribution. To remedy this problem, two-class [97], [98] or multi-class HEQ [99]–[101] can be used. All of these methods equalize different acoustic classes separately according to their corresponding class-specific distribution.

Conventional HEQ always equalizes the test utterance after visiting the whole utterance. This is not a problem for offline processing. However, for commercial systems with real-time requirements this is not acceptable. Initially proposed to address the time-varying noise issue, progressive HEQ [102] is a good candidate to meet the real-time processing requirement by equalizing with respect to a short interval around the current frame. The processing delay can be reduced from the length of the whole utterance to just half of the reference interval.

3) Feature Compensation: Feature compensation aims to remove the effect of noise from the observed speech features. In this section, we will introduce several methods in this class, but leave some to be discussed in later sections.

a) Spectral Subtraction: The spectral subtraction [103] method assumes that noise and clean speech are uncorrelated and additive in the time domain. Assuming the absence of channel distortions in Eq. (4), the power spectrum of the noisy signal is the sum of the noise and the clean speech power spectrum:

$$|\dot{\mathbf{y}}[k]|^2 = |\dot{\mathbf{x}}[k]|^2 + |\dot{\mathbf{n}}[k]|^2, \tag{35}$$

The method assumes that the noise characteristics change slowly relative to those of the speech signal. Therefore, the noise spectrum estimated during a non-speech period can be used for suppressing the noise contaminating the speech. The simplest way to get the estimated noise power spectrum, $|\hat{\dot{\mathbf{n}}}[k]|^2$, is to average the noise power spectrum in N non-speech frames:

$$|\hat{\dot{\mathbf{n}}}[k]|^2 = \frac{1}{N}\sum_{i=0}^{N-1} |\dot{\mathbf{y}}_i[k]|^2, \tag{36}$$

where $|\dot{\mathbf{y}}_i[k]|^2$ denotes the k th bin of the speech power spectrum in the i th frame.

Then the clean speech power spectrum can be estimated by subtracting $|\hat{\dot{\mathbf{n}}}[k]|^2$ from the noisy speech power spectrum:

$$|\hat{\dot{\mathbf{x}}}[k]|^2 = |\dot{\mathbf{y}}[k]|^2 - |\hat{\dot{\mathbf{n}}}[k]|^2 \tag{37}$$

$$= |\dot{\mathbf{y}}[k]|^2 G^2[k] \tag{38}$$

where

$$G[k] = \sqrt{\frac{\mathrm{SNR}(k)}{1+\mathrm{SNR}(k)}} \tag{39}$$

is a real-valued gain function, and

$$\mathrm{SNR}(k) = \frac{|\dot{\mathbf{y}}[k]|^2 - |\hat{\mathbf{n}}[k]|^2}{|\hat{\mathbf{n}}[k]|^2} \tag{40}$$

is the frequency-dependent signal-to-noise ratio estimate.

While the method is simple and efficient for stationary or slowly varying additive noise, it comes with several problems:

- The estimation of the noise spectrum from noisy speech is not an easy task. The simple scheme outlined in Eq. (36) relies on a voice activity detector (VAD). However, voice activity detection in low SNR is known to be error-prone. Alternatives have therefore been developed to estimate the noise spectrum without the need of a VAD. A comprehensive performance comparison of state of the art noise trackers can be found in [104].
- The instantaneous noise power spectral density will fluctuate around its temporally and spectrally smoothed estimate, resulting in amplification of random time frequency bins, a phenomenon known under the name musical noise [105], which is not only annoying to a human listener but also leads to word errors in a machine recognizer.
- The subtraction in Eq. (37) may result in negative power spectrum values because $|\hat{\mathbf{n}}[k]|^2$ is an estimated value and may be greater than $|\dot{\mathbf{y}}[k]|^2$. If this happens the numerator of Eq. (40) should be replaced by a small positive constant.

Many very sophisticated gain functions have been proposed which are derived from statistical optimization criteria.

b) Wiener Filtering: Wiener filtering is similar to spectral subtraction in that a real-valued gain function is applied in order to suppress the noise.

Wiener filtering aims at finding a linear filter $g[m]$ such that the sequence

$$\hat{\mathbf{x}}[m] = \mathbf{y}[m] * g[m] = \sum_{i=-\infty}^{\infty} g[i]\mathbf{y}[m-i] \tag{41}$$

has the minimum expected squared error from $\mathbf{x}[m]$. This results in the frequency domain filter

$$G[k] = \frac{S_{xy}[k]}{S_{yy}[k]}. \tag{42}$$

Here, S_{xy} and S_{yy} are the cross power spectral density between clean and noisy speech and the power spectral density of noisy speech, respectively.

With the assumption that the clean speech signal and the noise signal are independent, Eq. (42) becomes

$$G[k] = \frac{S_{xx}[k]}{S_{xx}[k] + S_{nn}[k]}, \tag{43}$$

which is referred to as the Wiener filter [106], [107], and can be realized only if $S_{xx}(f)$ and $S_{nn}(f)$ are known.

In practice the power spectra have to be estimated, e.g., via the periodograms $|\hat{\mathbf{x}}[k]|^2$ and $|\hat{\mathbf{n}}[k]|^2$. Plugging them in Eq. (43) we obtain

$$\begin{aligned} G[k] &= \frac{|\dot{\mathbf{y}}[k]|^2 - |\hat{\mathbf{n}}[k]|^2}{|\dot{\mathbf{y}}[k]|^2} \\ &= \frac{\mathrm{SNR}(k)}{1 + \mathrm{SNR}(k)}, \end{aligned} \tag{44}$$

which shows that Wiener filtering and spectral subtraction are closely related.

From Eq. (44), it is easy to see that the Wiener filter attenuates low SNR regions more than high SNR regions. If the speech signal is very clean with very large SNR approaching to ∞, $G[k]$ is close to 1, resulting in no attenuation. In contrast, if the speech is buried in the noise with very low SNR approaching 0, $G[k]$ is close to 0, resulting in total attenuation. Similar reasoning also applies to spectral subtraction.

c) Advanced Front-End: In 2002, the advanced front-end (AFE) for distributed speech recognition (DSR) was standardized by ETSI [108]. It obtained 53% relative word error rate reduction from the MFCC baseline on the Aurora 2 task [109]. The AFE is one of the most popular methods for comparison in the noise robustness literature. It integrates several noise robustness methods to remove additive noise with two-stage Mel-warped Wiener filtering [109] and SNR-dependent waveform processing [110], and mitigates the channel effect with blind equalization [111].

The two-stage Mel-warped Wiener filtering algorithm is the main body of the noise reduction module and accounts for the major gain of noise reduction. It is a combination of the two-stage Wiener filter scheme from [112] and the time domain noise reduction proposed in [113]. The algorithm has two stages of Mel Wiener filtering. The denoised signal in the first stage is passed to the second stage, which is used to further reduce the residual noise. Although having outstanding performance, the two-stage Mel-warped Wiener filtering algorithm has a high computational load which is significantly reduced in [114] by constructing and applying Wiener filters in the Mel-warped filter-bank domain.

The basic idea behind SNR-dependent waveform processing [110] is that the speech waveform exhibits periodic maxima and minima in the voiced speech segments due to the glottal excitation while the additive noise energy is relatively constant. Therefore, the overall SNR of the voiced speech segments can be boosted if one can locate the high (or low) SNR period portions and increase (or decrease) their energy.

Blind equalization [111] reduces convolutional distortion by minimizing the mean square error between the current and target cepstrum. The target cepstrum corresponds to the cepstrum of a flat spectrum. Blind equalization is an online method to remove convolutional distortion without the need to first visit the whole utterance as the standard CMN does. As shown in [111], it can obtain almost the same performance as the conventional offline cepstral subtraction approach. Therefore, it is preferred in real-time applications.

B. Model-Space Approaches

Model-domain methods modify the acoustic model parameters to incorporate the effects of noise. While typically achieving higher accuracy than feature-domain methods, they usually incur significantly higher computational cost. The model-domain approaches can be further classified into two sub-categories: general adaptation and noise-specific compensation. General adaptation methods compensate for the mismatch between training and testing conditions by using

generic transformations to convert the acoustic model parameters. These methods are general, applicable not only to noise compensation but also to other types of acoustic variations.

As in Eq. (29), model-domain methods only adapt the model parameters to fit the distorted speech signal. The model adaptation can operate in either supervised or unsupervised mode. In supervised mode, the correct transcription of the adapting utterance is available. It is used to guide model adaptation to obtain the adapted model, $\hat{\Lambda}$, used to decode the incoming utterances. In unsupervised mode, the correct transcription is not available, and usually two-pass decoding is used. In the first pass, the initial model Λ is used to decode the utterance to generate a hypothesis. Usually one hypothesis is good enough. The gain from using a lattice or N-best list to represent multiple hypotheses is limited [115]. Then, $\hat{\Lambda}$ is obtained with the model adaptation process and used to generate the final decoding result.

Popular speaker adaptation methods such as maximum a posteriori (MAP) and its extensions such as structural MAP (SMAP) [116], MAPLR [117], and SMAPLR [118], may not be a good fit for most noise-robust speech recognition scenarios where only a very limited amount of adaptation data is available, e.g., when only the utterance itself is used for unsupervised adaptation. Most popular methods use the maximum likelihood estimation (MLE) criterion [119], [120]. Discriminative adaptation is also investigated in some studies [121]–[123]. Unlike MLE adaptation, discriminative adaption is very sensitive to hypothesis errors [124]. As a result, most discriminative adaptation methods only work in supervised mode [121], [122]. Special processing needs to be used for unsupervised discriminative adaptation. In [123], a speaker-independent discriminative mapping transformation (DMT) is estimated during training. During testing, a speaker-specific transform is estimated with unsupervised ML, and the speaker-independent DMT is then applied. In this way, discriminative adaptation is implicitly applied without the strict dependency on a correct transcription.

In the following, popular MLE adaptation methods will be reviewed. Since they are general adaptation methods not specific to the problem of noise robustness, we will not address them in detail. Maximum likelihood linear regression (MLLR) is proposed in [119] to adapt model mean parameters with a class-dependent linear transform

$$\boldsymbol{\mu}_y(m) = \mathbf{A}(r_m)\boldsymbol{\mu}_x(m) + \mathbf{b}(r_m), \tag{45}$$

where $\boldsymbol{\mu}_y(m)$ and $\boldsymbol{\mu}_x(m)$ are the clean and distorted mean vectors for Gaussian component m, and r_m is the corresponding regression class. $\mathbf{A}(r_m)$ and $\mathbf{b}(r_m)$ are the regression-class-dependent transform and bias to be estimated, which can be put together as $\mathbf{W}(r_m) = [\mathbf{A}(r_m)\mathbf{b}(r_m)]$.

The expectation-maximization (EM) algorithm [125] is used to get the maximum likelihood solution of $\mathbf{W}(r_m)$. First, an auxiliary Q function for an utterance is defined

$$Q = \sum_{t,m} \gamma_t(m) \log p_{\hat{\Lambda}}(\mathbf{y}_t|m), \tag{46}$$

where $\hat{\Lambda}$ denotes the adapted model, and $\gamma_t(m)$ is the posterior probability for Gaussian component m at time t. $\mathbf{W}(r_m)$ can be obtained by setting the derivative of Q w.r.t. $\mathbf{W}(r_m)$ to 0. A special case of MLLR is the signal bias removal algorithm [126], where the only single transform is simply a bias. The MLE criterion is used to estimate this bias, and it is shown that signal bias removal is better than CMN [126].

The variance of the noisy speech signal also changes with the introduction of noise. Hence, in addition to transforming mean parameters with Eq. (45), it is better to also transform covariance parameters [120], [127] as

$$\boldsymbol{\Sigma}_y(m) = \mathbf{H}(r_m)\boldsymbol{\Sigma}_x(m)\mathbf{H}^T(r_m). \tag{47}$$

A two-stage optimization is usually used. First, the mean transform $\mathbf{W}(r_m)$ is obtained, given the current variance. Then, the variance transform $\mathbf{H}(r_m)$ is computed, given the current mean. The whole process can be done iteratively. The EM method is used to obtain the solution, which is done in a row-by-row iterative format.

Constrained MLLR (CMLLR) [120] is a very popular model adaption method in which the transforms of the mean and covariance, $\mathbf{A}(r_m)$ and $\mathbf{H}(r_m)$, are constrained to be the same:

$$\boldsymbol{\mu}_y(m) = \mathbf{H}(r_m)(\boldsymbol{\mu}_x(m) - \mathbf{g}(r_m)), \tag{48}$$

$$\boldsymbol{\Sigma}_y(m) = \mathbf{H}(r_m)\boldsymbol{\Sigma}_x(m)\mathbf{H}^T(r_m). \tag{49}$$

Rather than adapting all model parameters, CMLLR can be efficiently implemented in the feature space with the following relation

$$\mathbf{y} = \mathbf{H}(r_m)(\mathbf{x} - \mathbf{g}(r_m)), \tag{50}$$

or

$$\mathbf{x} = \mathbf{A}(r_m)\mathbf{y} + \mathbf{b}(r_m), \tag{51}$$

with $\mathbf{A}(r_m) = \mathbf{H}(r_m)^{-1}$ and $\mathbf{b}(r_m) = \mathbf{g}(r_m)$. The likelihood of the distorted speech $\mathbf{y}$ can now be expressed as

$$p(\mathbf{y}|m) = |\mathbf{A}(r_m)|\mathcal{N}(\mathbf{A}(r_m)y + \mathbf{b}(r_m); \boldsymbol{\mu}_x(m), \boldsymbol{\Sigma}_x(m)) \tag{52}$$

As a result, CMLLR is also referred to as feature space MLLR (fMLLR) in the literature. Note that signal bias removal is a special form of CMLLR with a unit scaling matrix.

In [128], fMLLR and its projection variant (fMLLR-P) [129] are used to adapt the acoustic features in noisy environments. Adaptation needs to accumulate sufficient statistics for the test data of each speaker, which requires a relatively large number of adaptation utterances.

As reported in [130] and [128], general adaptation methods such as MLLR and fMLLR in noisy environments yield moderate improvement, but with a large gap to the performance of noise-specific methods [131], [132] on the same task. Noise-specific compensation methods usually modify model parameters by explicitly addressing the nature of the distortions caused by the presence of noise. Therefore, they can address the noise-robustness issue better. The representative methods in this sub-category are parallel model combination (PMC) [133] and model-domain vector Taylor series [134]. Some representative noise-specific compensation methods will be discussed in detail in Section V.

IV. Compensation Using Prior Knowledge about Distortion

In addition to training an HMM, all methods analyzed in this section have the unique attribute of exploiting prior knowledge about distortion in the training stage. They then use such prior knowledge as a guide to either remove noise or adapt models in the testing stage.

A. Learning from Stereo Data

There are many methods that use stereo data to learn the mapping from noisy 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. The success of this kind of method usually depends on how well the representative training samples for the test environments really match test scenarios.

1) Empirical Cepstral Compensation: One group of methods is called empirical cepstral compensation [135], developed at CMU. In Eq. (14), the distorted speech cepstrum $\mathbf{y}$ is expressed as the clean speech signal $\mathbf{x}$ plus a bias $\mathbf{v}$. In empirical cepstral compensation, this bias $\mathbf{v}$ can be dependent on the SNR, the location of vector quantization (VQ) cluster k, the presumed phoneme identity p, and the specific testing environment e. Hence, Eq. (14) can be re-written as

$$\mathbf{y} = \mathbf{x} + \mathbf{v}(SNR, k, p, e). \tag{53}$$

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

$$\hat{\mathbf{x}} = \mathbf{y} - \mathbf{v}(SNR, k, p, e). \tag{54}$$

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) [136]. 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 signals 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. (54).

Fixed codeword-dependent cepstral normalization (FCDCN) [5] is a refined version of SDCN with the compensation vector as $\mathbf{v}(SNR, k)$, which depends on both SNR and VQ cluster location. Phone-dependent cepstral normalization (PDCN) [137] is another empirical cepstral compensation method in which the compensation vector depends on the presumed phoneme the current frame belongs to. It can also be extended to include SNR as a factor, and is called SNR-dependent PDCN (SPDCN) [137]. 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 [97].

2) SPLICE: Stereo-based Piecewise LInear Compensation for Environments (SPLICE), proposed originally in [138] and described in more detail in [40], [139], is a popular method to learn from stereo data and is more advanced than the above-mentioned empirical cepstral compensation methods. In SPLICE, the noisy speech data, $\mathbf{y}$, is modeled by a mixture of Gaussians

$$p(\mathbf{y}, k) = P(k)p(\mathbf{y}|k) = P(k)\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}(k), \boldsymbol{\Sigma}(k)), \tag{55}$$

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 $\mathbf{b}(k)$:

$$p(\mathbf{x}|\mathbf{y}, k) = \mathcal{N}(\mathbf{x}; \mathbf{y} + \mathbf{b}(k), \boldsymbol{\Psi}(k)), \tag{56}$$

where $\boldsymbol{\Psi}(k)$ 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 is

$$\hat{\mathbf{x}} = \sum_{k=1}^{K} P(k|\mathbf{y})(\mathbf{y} + \mathbf{b}(k)). \tag{57}$$

The prediction bias vector, $\mathbf{b}(k)$, is estimated by minimizing the mean square error (MMSE) as

$$\mathbf{b}(k) = \frac{\sum_t P(k|\mathbf{y}_t)(\mathbf{x}_t - \mathbf{y}_t)}{\sum_t P(k|\mathbf{y}_t)}, \tag{58}$$

and $\boldsymbol{\Psi}(k)$ can be obtained as

$$\boldsymbol{\Psi}(k) = \frac{\sum_t P(k|\mathbf{y}_t)(\mathbf{x}_t - \mathbf{y}_t)(\mathbf{x}_t - \mathbf{y}_t)^T}{\sum_t P(k|\mathbf{y}_t)} - \mathbf{b}(k)\mathbf{b}^T(k). \tag{59}$$

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

$$\begin{aligned} \hat{k} &= \arg\max_k p(\mathbf{y}, k), \\ \hat{\mathbf{x}} &= \mathbf{y} + \mathbf{b}_{\hat{k}}. \end{aligned} \tag{60}$$

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

$$\mathcal{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu}_x(k)\\ \boldsymbol{\mu}_y(k)\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_x(k) & \boldsymbol{\Sigma}_{xy}(k)\\ \boldsymbol{\Sigma}_{yx}(k) & \boldsymbol{\Sigma}_y(k)\end{bmatrix}\right). \tag{61}$$

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

$$E(\mathbf{x}|\mathbf{y}, k) = \mathbf{A}(k)\mathbf{y} + \mathbf{b}(k), \tag{62}$$

where

$$\mathbf{A}(k) = \boldsymbol{\Sigma}_{xy}(k)\boldsymbol{\Sigma}_y^{-1}(k) \tag{63}$$

$$\mathbf{b}(k) = \boldsymbol{\mu}_x(k) - \boldsymbol{\Sigma}_{xy}(k)\boldsymbol{\Sigma}_y^{-1}(k)\boldsymbol{\mu}_y(k). \tag{64}$$

The feature compensation formulation in this case is

$$\hat{\mathbf{x}} = \sum_{k=1}^{K} P(k|\mathbf{y})(\mathbf{A}(k)\mathbf{y} + \mathbf{b}(k)). \quad (65)$$

It is interesting that feature space minimum phone error (fMPE) training [140], a very popular feature space discriminative training method, can be linked to SPLICE to some extent [141]. Originally derived with the MMSE criterion, SPLICE can be improved with the maximum mutual information criterion [142] by discriminatively training $\mathbf{A}(k)$ and $\mathbf{b}(k)$[143]. In [144], 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, the MMSE-based stereo mapping is studied in [145], and the MAP-based stereo mapping is formulated in [146], [147]. Most stereo mapping methods use a GMM to construct a joint space of the clean and noisy speech signal. This is extended in [148], where a HMM is used. The mapping methods can also be extended into a discriminatively trained feature space, such as the fMPE space [149].

One concern for learning with stereo data is the requirement of stereo data, which may not be available in real-world application scenarios. In [150], the pseudo-clean features generated with a HMM-based synthesis method [151] are used to replace the clean features which are usually hard to get in a real deployment. It is shown that this pseudo-clean feature is even more effective than the ideal clean feature [150].

In addition to the above-mentioned methods, a recurrent neural network (RNN) has also been proposed to predict the clean speech from noisy speech [80] by modeling temporal signal dependencies in an explicit way. With its nonlinear modeling power, the RNN has been shown to be a very effective noise-cleaning method [80]. This is further improved with a bidirectional long short-term memory (BLSTM) structure [152] which allows for a more efficient exploitation of temporal context, leading to an improved feature mapping from noisy speech to clean speech.

B. Learning from Multi-Environment Data

Usually, the speech 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.

1) Linear Model Combination: The model combination methods build a set of acoustic models, each modeling one specific environment. During testing all the models are combined, usually with the MLE criterion, to construct a target model used to recognize the current test utterance. Assume that K environment-specific models share the same covariance matrix and only differ in mean parameters. The mean parameters for each environment-specific model are concatenated together to form mean supervectors ($\mathbf{s}_k, k = 1 \ldots K$), and the mean supervector of the testing utterance, $\mathbf{s}$, is obtained as a linear combination of K mean supervectors of the environment-specific models

$$\mathbf{s} = \sum_{k=1}^{K} w_k \mathbf{s}_k, \quad (66)$$

where w_k is the combination weight for the k-th mean supervector, and $\mathbf{w} = [w_1, w_2, \ldots, w_k]^T$.

The EM algorithm is used to find the solution of $\mathbf{w}$ iteratively. The auxiliary function is defined as the following by ignoring standard constants and terms independent of $\mathbf{w}$

$$Q(\mathbf{w}; \mathbf{w}_0) = -\frac{1}{2} \sum_{m,t} \gamma_t(m)(\mathbf{y}_t - \boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1} (\mathbf{y}_t - \boldsymbol{\mu}_m), \quad (67)$$

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

$$\boldsymbol{\mu}_m = \sum_{k=1}^{K} w_k \mathbf{s}_{k,m} = \mathbf{S}_m \mathbf{w}, \quad (68)$$

where $\mathbf{s}_{k,m}$ is the subvector for Gaussian component m in supervector $\mathbf{s}_k$ and $\mathbf{S}_m = [\mathbf{s}_{1,m}, \ldots, \mathbf{s}_{K,m}]$. Σ_m is the variance of Gaussian component m, shared by all the environment-specific models. By maximizing the auxiliary function, the combination weight $\mathbf{w}$ can be solved as

$$\mathbf{w} = \left[\sum_{m,t} \gamma_t(m) \mathbf{S}_m^T \boldsymbol{\Sigma}_m^{-1} \mathbf{S}_m \right]^{-1} \sum_{m,t} \gamma_t(m) \mathbf{S}_m^T \boldsymbol{\Sigma}_m^{-1} \mathbf{y}_t. \quad (69)$$

This model combination method is very similar to general speaker adaptation methods such as cluster adaptive training (CAT) [153] and eigenvoice [154]. In the CAT approach, the speakers are clustered together and $\mathbf{s}_i$ stands for clusters instead of individual speakers. In the eigenvoice approach, a small number of eigenvectors are extracted from all the supervectors and are used as $\mathbf{s}_i$. 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 noise-robust speech recognition. Storing K supervectors in memory during online model combination may be too demanding. One way to reduce the cost is to use methods such as eigenMLLR [155], [156] and transform-based CAT [153] by adapting a canonical mean 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 [153].

One potential problem of ML model combination is that usually all combination weights are not zero, 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. This can be solved by ensemble speaker and speaking environment modeling [157], in which an online

cluster selection is first used to locate the most relevant cluster and then only the supervectors in this selected cluster contribute to the model combination. Another way is to use Lasso (least absolute shrinkage and selection operator) [158] to impose an L_1 regularization term in the weight estimation problem. In [159], it is shown that Lasso usually shrinks the weights of the mean supervectors not relevant to the test environment to zero. By removing some irrelevant supervectors, the resulting mean supervectors are found to be more robust to noise distortions.

Note that the noisy speech signal 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 [160]. However, this is not trivial, requiring numerical optimization methods, such as the gradient descent method or a Newton method as in [160].

2) Source Separation: In Section IV-B1, the acoustic model for the current test utterance is obtained by combining the pre-trained acoustic models. Recently, there is increasing interest to use exemplar-based methods for general ASR [161], [162] and noise-robust ASR [163]–[165]. Exemplar refers to an example speech segment from the training corpus. In exemplar-based noise-robust ASR [163]–[165], noisy speech is modeled by a linear combination of speech and noise [163], [165] (or other interfering factors, such as music [164]) 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) [166] has been shown to be a very successful method [167], [168], and can directly benefit noise-robust ASR [163]–[165], [169]. The source separation process with NMF is described below.

First the training corpus is used to create a dictionary $\mathbf{x}_l (1 \leq l \leq L)$ 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 $\mathbf{A} = [\mathbf{X}\mathbf{N}]$, with a total of K exemplars. The exemplars of $\mathbf{A}$ are denoted $\mathbf{A}_k, 1 \leq k \leq K$. The reconstruction signal is

$$\hat{\mathbf{y}} = \sum_{k=1}^{K} w_k \mathbf{A}_k = \mathbf{A}\mathbf{w}, \quad s.t. \quad w_k \geq 0 \tag{70}$$

with $\mathbf{w}$ as the K-dimension activation vector. All exemplars and activation weights are non-negative. The objective is to minimize the reconstruction error $d(\mathbf{y}, \mathbf{A}\mathbf{w})$ between the observation $\mathbf{y}$ and the reconstruction signal $\hat{\mathbf{y}}$ 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. 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

$$d(\mathbf{y}, \mathbf{A}\mathbf{w}) + \|\boldsymbol{\lambda} .* \mathbf{w}\|_1 \quad s.t. \quad w_k \geq 0 \tag{71}$$

If all the elements of $\boldsymbol{\lambda}$ are zero, there is no enforced sparsity [164]. Otherwise, sparsity is enforced [163], [165]. In [166], two measures are used for the reconstruction error, namely Euclidean distance and divergence. In most speech-related work [163]–[165], Kullback-Leibler (KL) divergence is used to measure the reconstruction error.

$$d(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{e=1}^{E} y_e \log\left(\frac{y_e}{\hat{y}_e}\right) - y_e + \hat{y}_e, \tag{72}$$

where E is the vector dimension.

To solve Eq. (71), the entries of the vector $\mathbf{w}$ are initialized to unity. Then Eq. (71) can be minimized by iteratively applying the update rule [165]

$$\mathbf{w} \leftarrow \mathbf{w} .* (\mathbf{A}(\mathbf{y}./(\mathbf{A}\mathbf{w}))) ./ (\mathbf{A}\mathbf{1} + \boldsymbol{\lambda}) \tag{73}$$

with $.*$ and $./$ denoting element-wise multiplication and division, respectively. $\mathbf{1}$ 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 [167]. Good recognition performance has been observed particularly at very low SNR (below 0 dB). Better results are reported by using the following filtering [164], [165], [170] as

$$\mathbf{x} = \mathbf{y} .* \mathbf{A}^x \mathbf{w}^x ./ (\mathbf{A}^x \mathbf{w}^x + \mathbf{A}^n \mathbf{w}^n), \tag{74}$$

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 is referred as feature enhancement (FE) in [165], [170].

Instead of cleaning the noisy speech magnitude spectrum, a sparse classification (SC) method is proposed in [163] 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 maximum likelihood.

Although the root methodology of FE and SC are the same, i.e., NMF source separation, it is shown in [170], [171] 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 [172], 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 [171], [173]. Prior knowledge of the co-occurrence statistics of the basis functions for each source can also be employed to improve the performance of NMF [174]. In [175], 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 [170] 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. 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. However, there are still plenty of challenges. e.g., how to deal with convolutive channel distortions, how to most effectively deal with noise types in testing that have not been previously seen in the development of the noise dictionary, and how to generalize to LVCSR tasks.

3) Variable-Parameter HMM: Variable-parameter HMM (VPHMM) [176] models the speech Gaussian mean and variance parameters as a set of polynomial functions of the environment variable v, which is SNR in [176]. Hence, the Gaussian component m is now modeled as $\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}(m,v), \Sigma(m,v))$. $\boldsymbol{\mu}(m,v)$ and $\Sigma(m,v)$ are polynomial functions of environment variable v. For example, $\boldsymbol{\mu}(m,v)$ can be denoted by

$$\boldsymbol{\mu}(m,v) = \sum_{p=0}^{P} \mathbf{c}_p(m) v^p, \tag{75}$$

where $\mathbf{c}_p(m)$ 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. Other functions can also be used. For example, in [177], piecewise spline interpolation is used to represent the dependency of the HMM parameters on the conditioning parameters. To reduce the total number of parameters for VPHMM, parameter clustering can be employed [178]. The VPHMM parameters can be trained either with the MLE criterion [176] or a discriminative criterion [179]. In addition to Gaussian mean and variance parameters, other model parameters can also be modeled. In [180], [181], a more generalized form of VPHMM is investigated by modeling tied linear transforms as a function of environment variables.

During testing, speech model parameters can be calculated 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 environments.

V. Explicit Distortion Modeling

An explicit distortion modeling method is a noise-robustness method that employs a physical model of how distorted speech features are generated. Because the physical constraints are explicitly modeled, the explicit distortion modeling methods require only a relatively small number of distortion parameters to be estimated. They also exhibit high noise-robustness performance due to the explicit modeling of the distorted speech generation process.

A. Parallel Model Combination

Parallel model combination (PMC) uses the explicit distortion model to adapt the clean speech model. The model parameters of clean speech and noise in the cepstral domain are first transformed to the log-Mel-filter-bank domain and further to the Mel-filter-bank domain. Then, the model parameters of distorted speech in the Mel-filter-bank domain can be calculated by using the explicit distortion model which assumes that noise and clean speech are independent and additive in the Mel-filter-bank domain. With either the log-normal approximation [133] or the log-add approximation [133], the model parameters of distorted speech in the log-Mel-filter-bank domain can be obtained and finally are transformed back to the cepstral domain with the DCT transform.

The basic PMC method can also be extended for situations where there is channel distortion as well as additive noise [133]. A simple technique presented in [133] uses a one state single Gaussian speech model to calculate the convolutive component. An approximate solution of the convolutive component by steepest descent methods has also been reported [182], which relies on the Viterbi approximation and does not handle mixture of Gaussian distributions. The method in [183] uses an additional universal speech Gaussian mixture model and incorporates an existing bias estimation procedure [184] for channel estimation.

As shown in [185], the vector Taylor series (VTS) approximation appears to be more accurate than the log-normal approximation in PMC. Therefore, many studies of explicit distortion modeling have switched to the VTS direction over the last decade.

B. Vector Taylor Series

In recent years, a model-domain approach that jointly compensates for additive and convolutive (JAC) distortions (e.g., [131], [132], [134], [185]–[189]) has yielded promising results. The various methods proposed so far use a parsimonious nonlinear physical model to describe the environmental distortion and use the VTS approximation technique to find closed-form HMM adaptation and noise/channel parameter estimation formulas. Although some methods are referred to with different names, such as Jacobian adaptation [186] and JAC [131], [132], [188], they are in essence the VTS methods since VTS is used to linearize the involved nonlinearity, from which the solutions are derived.

Eq. (14) is a popular nonlinear distortion model between clean and distorted speech in the cepstral domain. It can be re-written as

$$\mathbf{y} = \mathbf{x} + \mathbf{h} + \mathbf{g}(\mathbf{x}, \mathbf{h}, \mathbf{n}), \tag{76}$$

where

$$\mathbf{g}(\mathbf{x}, \mathbf{h}, \mathbf{n}) = \mathbf{C} \log(1 + \exp(\mathbf{C}^{-1}(\mathbf{n} - \mathbf{x} - \mathbf{h})))). \tag{77}$$

In standard VTS adaptation [134], the nonlinear function in Eq. (77) is approximated using a first order VTS expansion at point $(\boldsymbol{\mu}_x(m), \boldsymbol{\mu}_h, \boldsymbol{\mu}_n)$. $\boldsymbol{\mu}_x(\mathrm{m})$, $\boldsymbol{\mu}_n$, and $\boldsymbol{\mu}_h$ are the static cepstral means of the clean speech Gaussian component m, noise, and channel, respectively.

Denoting

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}|_{\boldsymbol{\mu}_x(m), \boldsymbol{\mu}_n, \boldsymbol{\mu}_h} = \frac{\partial \mathbf{y}}{\partial \mathbf{h}}|_{\boldsymbol{\mu}_x(m), \boldsymbol{\mu}_n, \boldsymbol{\mu}_h} = \mathbf{G}(m), \tag{78}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{n}}|_{\boldsymbol{\mu}_x(m), \boldsymbol{\mu}_n, \boldsymbol{\mu}_h} = \mathbf{I} - \mathbf{G}(m) = \mathbf{F}(m), \tag{79}$$

where $\mathbf{G}(m)$ is the Jacobian matrix for Gaussian component m, defined as

$$\mathbf{G}(m) = \mathbf{C}diag\left(\frac{1}{1+\exp(\mathbf{C}^{-1}(\boldsymbol{\mu}_n - \boldsymbol{\mu}_x(m) - \mu_h))}\right)\mathbf{C}^{-1}, \tag{80}$$

Eq. (76) can then be approximated by a vector Taylor series expansion, truncated after the linear term:

$$\begin{aligned}\mathbf{y}|_{\boldsymbol{\mu}_x(m),\boldsymbol{\mu}_h,\boldsymbol{\mu}_n} \approx \boldsymbol{\mu}_x(m) + \boldsymbol{\mu}_h + \mathbf{g}(\boldsymbol{\mu}_x(m),\boldsymbol{\mu}_h,\boldsymbol{\mu}_n) \\ + \mathbf{G}(m)(\mathbf{x}-\boldsymbol{\mu}_x(m)) + \mathbf{G}(m)(\mathbf{h}-\boldsymbol{\mu}_h) + \mathbf{F}(m)(\mathbf{n}-\boldsymbol{\mu}_n),\end{aligned} \tag{81}$$

With Eq. (81), the distorted speech $\mathbf{y}$ is now a linear function of the clean speech $\mathbf{x}$, the noise $\mathbf{n}$, and the channel $\mathbf{h}$, in the cepstral domain. This linearity facilitates the HMM model adaptation and distortion parameter estimation by providing possible closed-form solutions because a Gaussian distribution, dominantly used in ASR, with linear operation is still a Gaussian distribution.

1) Vector Taylor Series Model Adaptation: By taking the expectation on both sides of Eq. (81), the static mean of the distorted speech signal $\boldsymbol{\mu}_y$ for Gaussian component m can be written as

$$\boldsymbol{\mu}_y(m) \approx \boldsymbol{\mu}_x(m) + \boldsymbol{\mu}_h + \mathbf{g}(\boldsymbol{\mu}_x(m),\boldsymbol{\mu}_h,\boldsymbol{\mu}_n), \tag{82}$$

and the static variance of the distorted speech signal $\boldsymbol{\mu}_y$ for Gaussian component m can be obtained by taking the variance operation on both sides of Eq. (81):

$$\boldsymbol{\Sigma}_y(m) \approx diag(\mathbf{G}(m)\boldsymbol{\Sigma}_x(m)\mathbf{G}(m)^T + \mathbf{F}(m)\boldsymbol{\Sigma}_n\mathbf{F}(m)^T), \tag{83}$$

The delta parameters can be updated [185] with the continuous time approximation [190]:

$$\boldsymbol{\mu}_{\Delta y}(m) \approx \mathbf{G}(m)\boldsymbol{\mu}_{\Delta x}(m), \tag{84}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{\Delta y}(m) \approx diag\Big(&\mathbf{G}(m)\boldsymbol{\Sigma}_{\Delta x}(m)\mathbf{G}(m)^T \\ &+\mathbf{F}(m)\boldsymbol{\Sigma}_{\Delta n}\mathbf{F}(m)^T\Big),\end{aligned} \tag{85}$$

The delta-delta model parameters are updated similarly.

Note that although the cepstral distortion formulation, i.e., Eq. (14), is widely used in VTS studies, the log spectral distortion formulation, i.e., Eq. (13), can also be used [188]. Some studies [191] even work in the linear frequency domain. However, this brings a large computational cost and the diagonal covariance assumption used in [191] may not be valid for linear frequency. Therefore, there are only a small number of VTS methods working in the linear frequency domain.

2) Distortion Estimation in VTS: Although proposed in 1996 [134], VTS model adaptation has shown great accuracy advantages over other noise-robustness methods only recently [131], [132] when the distortion model parameters are re-estimated based on the first-pass decoding result with the expectation-maximization (EM) algorithm [131], [132]. First, an auxiliary Q function for an utterance is

$$Q = \sum_{t,m}\gamma_t(m)\log p_{\hat{\Lambda}}(\mathbf{y}_t|m), \tag{86}$$

where $\hat{\Lambda}$ denotes the adapted model, and $\gamma_t(m)$ is the posterior probability for the Gaussian component m of the HMM, i.e.,

$$\gamma_t(m) = p_{\hat{\Lambda}_0}(m|\mathbf{Y}), \tag{87}$$

where $\hat{\Lambda}_0$ denotes the previous model.

To maximize the auxiliary function in the M-step of the EM algorithm, the derivatives of Q are taken with respect to $\boldsymbol{\mu}_n$ and $\boldsymbol{\mu}_h$, and are set to 0. Then, the mean of the noise $\boldsymbol{\mu}_n$ can be updated according to [132]:

$$\boldsymbol{\mu}_n = \boldsymbol{\mu}_{n,0} + \mathbf{A}_n^{-1}\mathbf{b}_n, \tag{88}$$

with

$$\mathbf{A}_n = \sum_{t,m}\gamma_t(m)\mathbf{F}(m)^T\boldsymbol{\Sigma}_y^{-1}(m)\mathbf{F}(m), \tag{89}$$

$$\mathbf{b}_n = \sum_{t,m}\gamma_t(m)\mathbf{F}(m)^T\boldsymbol{\Sigma}_y^{-1}(m)(\mathbf{y}_t - \boldsymbol{\mu}_{y,0}(m)), \tag{90}$$

$$\boldsymbol{\mu}_{y,0}(m) = \boldsymbol{\mu}_x(m) - \boldsymbol{\mu}_{h,0} - \mathbf{g}(\boldsymbol{\mu}_x(m),\boldsymbol{\mu}_{h,0},\boldsymbol{\mu}_{n,0}). \tag{91}$$

where $\boldsymbol{\mu}_{n,0}$ and $\boldsymbol{\mu}_{h,0}$ are the VTS expansion points for $\boldsymbol{\mu}_n$ and $\boldsymbol{\mu}_h$, respectively. The channel mean $\boldsymbol{\mu}_h$ can be updated similarly as in [132]. Newton's method, an iterative second order approach, is used to estimate the noise variance [132].

Distortion parameter estimation can also be done in other ways. In [192], a gradient-descent method is used to obtain the noise variance estimate. Since there is no guarantee that the auxiliary function will increase, a back-off step is needed. In [193], a Gauss-Newton method is used by discarding the second derivative of the residual with respect to the distortion parameters when calculating the Hessian. In [187], both the static mean and variance parameters in the cepstral domain are adjusted using the VTS approximation technique. In that work, however, noise was estimated on a frame-by-frame basis, which is complex and computationally costly. It is shown in [194] that the estimation method used in this section [131], [132] is clearly better than the estimation method in [187].

3) VTS Feature Enhancement: As shown in [132], VTS model adaptation achieves much better accuracy than several popular model adaptation technologies. Although VTS model adaptation can achieve high accuracy, the computational cost is very high as all the Gaussian parameters in the recognizer need to be updated every time the environmental parameters change. This time-consuming requirement hinders VTS model adaptation from being widely used, especially in LVCSR tasks where the number of model parameters is large.

On the other hand, VTS feature enhancement has been proposed as a lower-cost alternative to VTS model adaptation. For example, a number of techniques have been proposed that can be categorized as model-based feature enhancement schemes [134], [195]–[197]. These methods use a small GMM in the front-end and the same methodology used in VTS model adaptation to derive a minimum-mean-square-error (MMSE) estimate of the clean speech features given the noisy observations. In addition to the advantage of a low runtime cost, VTS feature enhancement can be easily combined with other popular

feature-based technologies, such as CMN, HLDA, fMPE, etc., which are challenging to VTS model adaptation.

In general, the MMSE method can be used to get the estimate of clean speech

$$\hat{\mathbf{x}} = E(\mathbf{x}|\mathbf{y}) = \int \mathbf{x} p(\mathbf{x}|\mathbf{y}) d\mathbf{x}. \tag{92}$$

Denote the clean-trained GMM as

$$p_\Lambda(x) = \sum_{k=1}^{K} \mathbf{c}(k) \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_x(k), \boldsymbol{\Sigma}_x(k)), \tag{93}$$

along with Eq. (14), the MMSE estimate of clean speech becomes

$$\hat{\mathbf{x}} = \mathbf{y} - \mathbf{h} - \sum_{k=1}^{K} P(k|\mathbf{y}) \int \mathbf{C} \log(1 + \exp(\mathbf{C}^{-1}(\mathbf{n} - \mathbf{x} - \mathbf{h}))) p(\mathbf{x}|\mathbf{y}, k) d\mathbf{x}, \tag{94}$$

where $P(k|\mathbf{y})$ is the Gaussian posterior probability, calculated as

$$P(k|\mathbf{y}) = \frac{\mathbf{c}(k)\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_y(k), \boldsymbol{\Sigma}_y(k))}{\sum_{k=1}^{K} \mathbf{c}(k)\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_y(k), \boldsymbol{\Sigma}_y(k))}. \tag{95}$$

If the 0th-order VTS approximation is used for the nonlinear term in Eq. (94), the MMSE estimate of cleaned speech $\mathbf{x}$ is obtained as

$$\hat{\mathbf{x}} = \mathbf{y} - \mathbf{h} - \sum_{k=1}^{K} P(k|\mathbf{y}) \mathbf{C} \log(1 + \exp(\mathbf{C}^{-1}(\boldsymbol{\mu}_n - \boldsymbol{\mu}_x(k) - \boldsymbol{\mu}_h))). \tag{96}$$

This formulation was first proposed in [134]. In [195], another solution was proposed when expanding Eq. (14) with the 1st-order VTS. For the k th GMM component, the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ is modeled as Eq. (61).

The following can be derived [195]

$$E(\mathbf{x}|\mathbf{y}, k) = \boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}}(k) = \boldsymbol{\mu}_x(k) + \boldsymbol{\Sigma}_{xy}(k)\boldsymbol{\Sigma}_y^{-1}(k)(\mathbf{y} - \boldsymbol{\mu}_y(k)). \tag{97}$$

Then the MMSE estimate of clean speech is [195]

$$\hat{\mathbf{x}} = \sum_{k=1}^{K} P(k|\mathbf{y}) \left(\boldsymbol{\mu}_x(k) + \boldsymbol{\Sigma}_x(k)\mathbf{G}(k)^T \boldsymbol{\Sigma}_y^{-1}(k)(\mathbf{y} - \boldsymbol{\mu}_y(k)) \right). \tag{98}$$

Two key aspects of VTS feature enhancement are how to obtain reliable estimates of the noise and channel distortion parameters and how to accurately calculate the Gaussian occupancy probability. In contrast to using static features alone to calculate the Gaussian occupancy probability [189], both static and dynamic features are used to obtain more reliable Gaussian occupancy probabilities. Then, these probabilities are plugged into Eq. (96) or Eq. (98). In [198], it is shown that recent improvements in VTS model adaptation can be incorporated into VTS feature enhancement to improve the algorithm performance: Updating all of the environment distortion parameters [131] and subsequently carrying out noise adaptive training [199].

A common concern of feature enhancement is that after the enhancement, the clean speech signal is distorted and the accuracy on clean testing will drop. As shown in [200], VTS feature enhancement enjoys the nice property that it significantly improves accuracy in noisy test conditions without degrading accuracy in clean test conditions.

By incorporating the recent advances in VTS model adaptation, VTS feature enhancement can obtain very high accuracy on some noisy tasks [198]. However, it is shown that there is still a small accuracy gap between VTS feature enhancement and VTS model adaptation [198]. Regarding the runtime cost, VTS model adaptation needs to adapt HMM parameters twice, while VTS feature enhancement needs to adapt GMM parameters twice. Usually, the number of parameters in a front-end GMM is much smaller than that in the back-end HMM. Furthermore, two rounds of decoding are needed in VTS model adaptation while only one round of decoding is performed in VTS feature enhancement. As a consequence, VTS feature enhancement has a much lower computational cost than VTS model adaptation. Therefore, the tradeoff between accuracy and computational cost will determine which technology is more suitable in a real world deployment scenario.

4) Improvements over VTS: Recently, there has been a series of studies focusing on how to improve the performance of VTS. A natural extension to the VTS methods described in the previous section is to use high-order VTS expansion instead of first-order VTS expansion. That way, the nonlinear relation found in Eq. (14) can be well modeled. There are several studies [201]–[203] along this line. As shown in [203], the 2nd-order VTS is shown to achieve a noticeable performance gain over the 1st-order VTS, although the accuracy gap between the 3rd-order VTS and 2nd-order VTS is small. Another way to address the inaccuracy problem of the first-order Taylor series expansion in the VTS is to use piecewise functions to model the nonlinearity, such as a piecewise linear approximation [204] or linear spline interpolation [205].

Eq. (14) is simplified from Eq. (12) of the phase-sensitive model, by approximating $\boldsymbol{\alpha}[l]$ as 0 [19]. There is some work [202] that uses the phase-sensitive model for better modeling of the distortion. $\boldsymbol{\alpha}$ can be estimated from the training set [202]. Due to its physical interpretation, the value of elements in $\boldsymbol{\alpha}$ ranges from -1 to 1. In [206], this value constraint is broken by assigning a constant value α to the elements of $\boldsymbol{\alpha}$. If α is set to 0 and 1, VTS can be considered to work with MFCCs extracted from the power spectrum and magnitude spectrum, respectively [132]. It is shown in [206] that the best accuracy is obtained on the Aurora 2 task when the value is set to around 2.5, which is larger than 1, the theoretical maximum value. Similar observations are also reported in later work [207], [208]. Therefore, the phase-sensitive model with a constant α value can also be considered to be a generalization function of the distortion model. The phase-sensitive model with a large α value may be considered as a way to compensate the loss brought by the inaccurate approximation in VTS. Another way to handle the phase term

is to use the ALGONQUIN algorithm [209], which models the phase term as the modeling error with Eq. (76).

In standard VTS, the delta and delta-delta model parameters are updated [185] with the continuous time approximation [190], which makes the assumption that the dynamic cepstral coefficients are the time derivatives of the static cepstral coefficients. In [210], extended VTS is proposed to provide a more accurate form to adapt dynamic model parameters. Extended VTS has been shown to outperform standard VTS with the cost of more expensive computation [210].

As mentioned before, high computational cost is a concern for VTS model adaptation. Feature VTS enhancement uses a small GMM on the front-end and the same methodology used in VTS model adaptation to derive a MMSE estimate of the clean speech features given the noisy observations. However, even after employing the recent advanced methods in VTS model adaptation, feature VTS enhancement still has an obvious accuracy gap between it and VTS model adaptation [198]. In [200], VTS model adaptation with a diagonal Jacobian approximation method is proposed to have a relatively small accuracy loss and to offer a drastic savings in computational cost over all three major components in standard VTS model adaptation. The computational cost reduction is on a scale of $\mathcal{O}(D)$ for the Jacobian calculation and most parts of parameter adaptation, and $\mathcal{O}(D^2)$ for online distortion estimation, where D is the dimension of static cepstral feature. There is also a family of joint uncertainty decoding (JUD) methods [211]–[213] that can reduce the computational cost of VTS by changing the Jacobian in Eq. (80) from being Gaussian-dependent to regression-class-dependent. We will discuss these methods in Section VI-C2.

C. Sampling-Based Methods

The PMC methods in Section V-A rely on either the log-normal or the log-add approximation while the VTS methods in Section V-B rely on the first-order or higher-order VTS approximation. These approximations inevitably cause loss in model adaptation or feature enhancement. To improve the implementation accuracy of explicit distortion modeling, sampling-based methods can be used.

1) Data-Driven PMC: Data-driven parallel model combination (DPMC) [133] can be used to improve the modeling accuracy of PMC. This method is based on Monte-Carlo (MC) sampling by drawing random samples from the clean speech and noise distributions. In a non-iterative DPMC, the frame/state component alignment within a state does not change, and the clean speech samples are drawn from each Gaussian of the clean speech distributions.

$$\begin{aligned}\mathbf{x}(m) &\sim \mathcal{N}(\boldsymbol{\mu}_x(m), \boldsymbol{\Sigma}_x(m)) \\ \mathbf{n} &\sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)\end{aligned} \tag{99}$$

Then, the distorted speech samples can be obtained with Eq. (14), and the static mean and variance of the distorted speech for Gaussian component m are estimated as the sample mean and variance of N distorted speech samples.

As $N \rightarrow \infty$, the sample mean and covariance are approaching the true values. However, due to the nature of random sampling, N needs to be very large to guarantee the approximation accuracy. Hence, the biggest disadvantage of DPMC is the computational cost. As a solution, a model adaptation method based on the unscented transform [214] is proposed in [215], [216].

2) Unscented Transform: Originally developed to improve the extended Kalman filter and introduced to the field of robust ASR in [215], [216], the unscented transform (UT) [214] gives an accurate estimate of the mean and variance parameters of a Gaussian distribution under a nonlinear transformation by drawing only a limited number of samples. This is achieved by systematically drawing samples jointly from the clean speech and noise distributions, described in the following.

An augmented signal $\mathbf{s} = [\mathbf{x}^T, \mathbf{n}^T]^T$ is formed with a D-dimensional clean speech cepstral vector $\mathbf{x}$ and a noise cepstral vector $\mathbf{n}$, with dimensionality $D_s = D_x + D_n = 2D$. The UT algorithm samples the Gaussian-dependent augmented signal with $4D + 1$ sigma points $\mathbf{s}_i(m)$.

In the feature space, the transformed sample $\mathbf{y}_i(m)$ from the sigma point $\mathbf{s}_i(m) = [\mathbf{x}_i(m)^T, \mathbf{n}^T]^T$ is obtained with the mapping function of Eq. (14). Then, the static mean and variance of the distorted speech are estimated as the sample mean and variance of these $4D + 1$ transformed samples.

It is shown in [214] that the UT accurately matches the mean and covariance of the true distribution. Due to the special sampling strategy of the UT, the number of samples to be computed, $4D + 1$, is much smaller than N. Therefore model adaptation with the UT is more affordable than the MC method.

In [216], the static mean and variance of nonlinearly distorted speech signals are estimated using the UT, but the static noise mean and variance are estimated from a simple average of the beginning and ending frames of the current utterance. This technique was improved in [217], where the static noise parameters were estimated online with MLE using the VTS approximation and the estimates were subsequently plugged into the UT formulation to obtain the estimate of the mean and variance of the static distorted speech features. In [218], a robust feature extraction technique is proposed to estimate the parameters of the conditional noise and channel distribution using the UT and embed the estimated parameters into the EM framework. In all of these approaches [216]–[218], sufficient statistics of only the static features or model parameters are estimated using the UT although adaptation of the dynamic model parameters with reliable noise and channel estimations has shown to be important [131], [132]. As a solution, an approach is proposed in [219] to unify static and dynamic model parameter adaptation with online estimation of noise and channel parameters in the UT framework.

3) Methods Beyond the Gaussian Assumption: As shown in Fig. 2, with the introduction of noise, the distorted speech is no longer Gaussian distributed. Therefore, the popular one-to-one Gaussian mapping used in the above-mentioned methods has a theoretic flaw. The iterative DPMC method [133] solves this problem by sampling from GMMs instead of Gaussians and then uses the Baum-Welch algorithm to re-estimate the distorted speech parameters. This is extended in [220], where a variational method is used to remove the constraint that the samples must be used to model the Gaussians they are originally drawn from. This is also extended to variational

PCMLLR [220], which is shown to be better than PCMLLR [211] and has a much lower computational cost than variational DPMC. In [221] the Gaussian at the input of the nonlinearity is approximated by a GMM whose individual components have a smaller variance than the original Gaussian. A VTS linearization of the individual GMM components then incurs fewer errors than the linearization of the original Gaussian. Thus the overall modeling accuracy could be improved.

The explicit distortion modeling methods discussed in this section separate the clean speech feature from the environment (noise and channel) factors. This can be further extended to an acoustic factorization [222] problem: separate the clean speech feature/model from the multiple speaker and environmental factors irrelevant to the phonetic classification. There are plenty of recent studies addressing acoustic factorization [223]–[228].

From the work on explicit distortion modeling and acoustic factorization, we can see the trend of building increasingly sophisticated models to characterize the impact of different distortion sources, such as noise, channel, and speaker, on clean speech. Importantly, these better and better explicit distortion models and the related techniques are already providing outstanding performance which is superior to other methods exploiting less powerful distortion models. A greater performance gap is expected in the future as more advanced explicit distortion models are being developed and incorporated into noise-robust ASR methods.

VI. Compensation with Uncertainty Processing

The effects of strong noise necessarily create inherent uncertainty, either in the feature or model space, which can be beneficially integrated into the popular plug-in MAP decoding in the ASR process. When a noise-robust method takes into consideration that uncertainty, we call it an uncertainty processing method.

A. Model-Domain Uncertainty

Uncertainty in the HMM parameters has been represented by their statistical distribution [229]. In order to take advantage of the model parameter uncertainty, the decision rule for recognition can be improved from the conventional MAP decision rule in Eq. (100)

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} p_{\Lambda}(\mathbf{Y}|\mathbf{W}) P_{\Gamma}(W). \tag{100}$$

to the minimax decision rule [230][1]

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \left(P_{\Gamma}(\mathbf{W}) \max_{\Lambda \in \Omega} p_{\Lambda}(\mathbf{Y}|\mathbf{W}) \right), \tag{101}$$

or to the Bayesian prediction classification (BPC) rule [231], [232]

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \left(\int_{\Lambda \in \Omega} p_{\Lambda}(\mathbf{Y}|\mathbf{W}) p(\Lambda|\phi, \mathbf{W}) d\Lambda \right) P_{\Gamma}(\mathbf{W}), \tag{102}$$

[1]This is derived from minimizing the upper bound of the worse-case probability of classification error.

where ϕ is the hyper-parameter characterizing the distribution of acoustic model parameter Λ, and Ω denotes the space that Λ lies in. Both minimax classification and BPC consider the uncertainty of the estimated model, reflected by Ω. They change the decision rule to address this uncertainty using two steps. In the first step, either the maximum value of $p_{\Lambda}(\mathbf{Y}|\mathbf{W})$ within the parameter neighborhood (as in minimax classification) or the integration of $p_{\Lambda}(\mathbf{Y}|\mathbf{W})$ in this parameter neighborhood (as in BPC) for word W is obtained. In the second step, the value obtained in the first step is plugged into the MAP decision rule.

It is usually difficult to define the parameter neighborhood Ω. Moreover, with two-stage processing the computational cost of model space uncertainty using the modified decision rule is very large. It usually involves a very complicated implementation, which prevents this type of method from being widely used although there was some research into minimax classification [230], [233] and BPC [231], [232] until around 10 years ago. An alternative treatment of uncertainty is by integrating over the feature space instead of over the model space. This will offer a much simpler system implementation and lower computational cost. Therefore, research has switched to feature space uncertainty or joint uncertainty decoding as described in the following sections.

B. Feature-Domain Uncertainty

1) Observation Uncertainty: Although it has been shown that more gain can be obtained with a context window (e.g., [64], [234]), it is still a very popular assumption in most noise-robust ASR methods that the clean feature is only dependent on the distorted feature of current frame. In this way, $p(\mathbf{x}_t|\mathbf{Y})$ is replaced by $p(\mathbf{x}_t|\mathbf{y}_t)$ and Eq. (24) becomes

$$P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{Y}) = P_{\Gamma}(\mathbf{W}) \sum_{\theta} \prod_{t=1}^{T} \int \frac{p_{\Lambda}(\mathbf{x}_t|\theta_t) p(\mathbf{x}_t|\mathbf{y}_t)}{p(\mathbf{x}_t)} d\mathbf{x}_t \, P_{\Lambda}(\theta_t|\theta_{t-1}). \tag{103}$$

As in Eq. (26), the most popular noise robustness techniques use a point estimate which means that the back-end recognizer considers the cleaned signal $\hat{\mathbf{x}}_t(\mathbf{Y})$ to be noise free. However, the de-noising process is not perfect and there may exist some residual uncertainty. Hence, in the observation uncertainty work [235], instead of using a point estimate of clean speech, a posterior is passed to the back-end recognizer. The prior $p(\mathbf{x}_t)$ always has a larger variance than the posterior $p(\mathbf{x}_t|\mathbf{y}_t)$. If it is much larger, the denominator $p(\mathbf{x}_t)$ in Eq. (103) can be considered constant in the range of values around x_t. As a consequence, the denominator $p(\mathbf{x}_t)$ is neglected in [236]–[238], and Eq. (103) becomes

$$P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{Y}) = P_{\Gamma}(\mathbf{W}) \sum\nolimits_{\theta} \prod\nolimits_{t=1}^{T} \int p_{\Lambda}(\mathbf{x}_t|\theta_t) p(\mathbf{x}_t|\mathbf{y}_t) d\mathbf{x}_t \, P_{\Lambda}(\theta_t|\theta_{t-1}). \tag{104}$$

If the de-noised speech is $\hat{\mathbf{x}}_t(\mathbf{Y})$, the clean speech estimate can be considered as a Gaussian distribution

$\mathbf{x}_t \sim \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_t, \mathbf{\Sigma}_{\hat{x}})$. Then, the integration in Eq. (104) reduces to

$$\int p_\Lambda(\mathbf{x}_t|\theta_t)p(\mathbf{x}_t|\mathbf{y}_t)d\mathbf{x}_t = \sum_m c(m)\mathcal{N}(\hat{\mathbf{x}}_t; \boldsymbol{\mu}_x(m), \mathbf{\Sigma}_x(m) + \mathbf{\Sigma}_{\hat{x}}). \quad (105)$$

It is clear that during recognition, every Gaussian component in the acoustic model has a variance bias $\mathbf{\Sigma}_{\hat{x}}$ in observation uncertainty methods. The key is to have this frame-dependent variance $\mathbf{\Sigma}_{\hat{x}}$, which depends on which noise-robustness method is used to clean the noise. If SPLICE is used as in Section IV-A2, the bias variance is given in Eq. (59). In [235], it is a polynomial function of SNR.

Eq. (103) is reduced to Eq. (104) by omitting $p(\mathbf{x}_t)$ with the belief that it has a larger variance than the posterior $p(\mathbf{x}_t|\mathbf{y}_t)$. However, this assumption may not be always true. Another better variation of Eq. (103) is to multiply both the numerator and denominator by $p(\mathbf{y}_t)$. By applying Bayes' rule, we get

$$\int \frac{p_\Lambda(\mathbf{x}_t|\theta_t)p(\mathbf{x}_t|\mathbf{y}_t)}{p(\mathbf{x}_t)}d\mathbf{x}_t = \int p_\Lambda(\mathbf{x}_t|\theta_t)p(\mathbf{y}_t|\mathbf{x}_t)d\mathbf{x}_t \frac{1}{p(\mathbf{y}_t)}. \quad (106)$$

Then, Eq. (106) is plugged into Eq. (103) and $p(\mathbf{y}_t)$ is omitted since it does not affect the MAP decision rule, and we obtain

$$P_{\Lambda,\Gamma}(\mathbf{W}|\mathbf{Y}) = P_\Gamma(\mathbf{W}) \sum_\theta \prod_{t=1}^{T} \int p_\Lambda(\mathbf{x}_t|\theta_t)p(\mathbf{y}_t|\mathbf{x}_t)d\mathbf{x}_t \, P_\Lambda(\theta_t|\theta_{t-1}). \quad (107)$$

We can denote

$$p_\Lambda(\mathbf{y}_t|\theta_t) = \int p_\Lambda(\mathbf{x}_t|\theta_t)p(\mathbf{y}_t|\mathbf{x}_t)d\mathbf{x}_t. \quad (108)$$

The key to calculating $p_\Lambda(\mathbf{y}_t|\theta_t)$ is to estimate the conditional distribution $p(\mathbf{y}_t|\mathbf{x}_t)$ because $p_\Lambda(\mathbf{x}_t|\theta_t)$ has already been trained. It is better to denote $p(\mathbf{y}_t|\mathbf{x}_t)$ as a Gaussian or GMM so that the integration of Eq. (108) is still a Gaussian or GMM. Eq. (107) is used in uncertainty decoding with SPLICE [239] and joint uncertainty decoding work [192], [212], [240], [241] which we will discuss in detail in the next section. An interesting alternative supervised approach to estimate uncertainties was proposed in [242]. A more recent study on propagating uncertainties from short-time Fourier transform into the nonlinear feature domain appeared in [243] for noise-robust ASR.

C. Joint Uncertainty Decoding

Joint uncertainty decoding (JUD) uses a feature transform derived from the joint distribution between the clean and noisy speech and an uncertainty variance bias to modify the decoder. While the joint distribution can be estimated from stereo data as in SPLICE [139], the most popular way to obtain it is to use the physical distortion model as in Section V. JUD has two implementation forms: front-end JUD and model JUD. In front-end JUD, a front-end GMM is built and one of its components is selected to pass one single transform and bias variance to the decoder. In contrast, model JUD is connected with the acoustic model and generates transform and uncertainty variance bias based on the regression class that the individual acoustic model component belongs to.

1) Front-end JUD: In front-end JUD, $p(\mathbf{y}_t|\mathbf{x}_t)$ is represented by a GMM

$$p(\mathbf{y}_t|\mathbf{x}_t) \approx \sum_k P(k|\mathbf{x}_t)\mathcal{N}\left(\mathbf{y}_t; f_\mu(\mathbf{x}_t, k), f_\Sigma(\mathbf{x}_t, k)\right), \quad (109)$$

where $f_\mu(\mathbf{x}_t, k)$ and $f_\Sigma(\mathbf{x}_t, k)$ are functions used to calculate the mean vector and covariances matrix. The joint distribution of clean and noisy speech can be modeled the same as in Eq. (61). It can be either obtained from stereo training data or derived with physical distortion modeling as in Section V. In most JUD methods, the latter option is used given the difficulty to obtain stereo data.

The Gaussian conditional distribution in Eq. (109) can be derived front the joint distribution as [212]

$$\mathcal{N}\left(\mathbf{y}_t; f_\mu(\mathbf{x}_t, k), f_\Sigma(\mathbf{x}_t, k)\right) = |\mathbf{A}(k)|\mathcal{N}\left(\mathbf{A}(k)\mathbf{y}_t + \mathbf{b}(k); \mathbf{x}_t, \Sigma_b(k)\right), \quad (110)$$

with

$$\mathbf{A}(k) = \mathbf{\Sigma}_x(k)\mathbf{\Sigma}_{yx}^{-1}(k) \quad (111)$$

$$\mathbf{b}(k) = \boldsymbol{\mu}_x(k) - \mathbf{A}(k)\boldsymbol{\mu}_y(k) \quad (112)$$

$$\mathbf{\Sigma}_b(k) = \mathbf{A}(k)\mathbf{\Sigma}_y(k)\mathbf{A}^T(k) - \mathbf{\Sigma}_x(k) \quad (113)$$

These transforms can be obtained using VTS related schemes.

Using Eq. (110) and a rough assumption that $P(k|\mathbf{x}_t) \approx P(k|\mathbf{y}_t)$, Eq. (108) can be re-written as [212]

$$p_\Lambda(\mathbf{y}_t|\theta_t) = \sum_k \sum_m c(m)P(k|\mathbf{y}_t)|\mathbf{A}(k)| \mathcal{N}\left(\mathbf{A}(k)\mathbf{y}_t + \mathbf{b}(k); \boldsymbol{\mu}_x(m), \mathbf{\Sigma}_x(m) + \mathbf{\Sigma}_b(k)\right) \quad (114)$$

The summation in Eq. (114) is time consuming, involving the clean speech GMM component m and the front-end GMM component k. One popular approach is to select the most dominating front-end component k^*

$$k^* = \arg\max_k P(k|\mathbf{y}_t) \quad (115)$$

and Eq. (114) can be simplified as

$$p_\Lambda(\mathbf{y}_t|\theta_t) \approx \sum_m c(m)|\mathbf{A}(k^*)| \mathcal{N}\left(\mathbf{A}(k^*)\mathbf{y}_t + \mathbf{b}(k^*); \boldsymbol{\mu}_x(m), \mathbf{\Sigma}_x(m) + \mathbf{\Sigma}_b(k^*)\right) \quad (116)$$

Comparing Eq. (116) with Eq. (105), we can see that front-end uncertainty decoding transfers the distorted feature $\mathbf{y}_t$ in addition to adding a variance bias. SPLICE with uncertainty decoding [239] is similar to front-end JUD, but with a different format of $\mathbf{A}(k)$, $\mathbf{b}(k)$, and $\mathbf{\Sigma}_b(k)$.

As discussed in [212], [244], in low SNR conditions where noise dominates the speech signal, the conditional distribution $p(\mathbf{y}_t|\mathbf{x}_t)$ degenerates to the distribution of additive noise $\mathbf{n}_t$ as

$$p(\mathbf{y}_t|\mathbf{x}_t) \approx \mathcal{N}(\mathbf{n}_t; \boldsymbol{\mu}_n, \mathbf{\Sigma}_n). \quad (117)$$

Then the distribution of distorted speech in Eq. (108) also becomes the distribution of additive noise

$$p_\Lambda(\mathbf{y}_t|\theta_t) = \mathcal{N}(\mathbf{n}_t; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n). \tag{118}$$

With Eq. (118), the distribution of every state is the same. Therefore, the current frame cannot contribute to differentiating states using acoustic model scores. This is the biggest theoretical issue with front-end uncertainty decoding, although SPLICE with uncertainty decoding can circumvent this issue with additional processing [239].

2) Model JUD: In front-end JUD, its conditional distribution is completely decoupled from the acoustic model used for recognition. In contrast, model JUD [192] links them together with

$$p(\mathbf{y}_t|\mathbf{x}_t) \approx \mathcal{N}\left(\mathbf{y}_t; f_\mu(\mathbf{x}_t, r_m), f_\Sigma(\mathbf{x}_t, r_m)\right), \tag{119}$$

where r_m is the regression class index of acoustic model Gaussian component m, generated with the method in [245]. The joint distribution of clean and noisy speech can be modeled similarly as in Eq. (61) by replacing the front-end component index k with the regression class index r_m. With a similar derivation as in front-end JUD, the likelihood of distorted speech can be denoted by

$$p_\Lambda(\mathbf{y}_t|\theta_t) \approx \sum_m c(m)|\mathbf{A}(r_m)|\mathcal{N}(\mathbf{A}(r_m)\mathbf{y}_t + \mathbf{b}(r_m); \boldsymbol{\mu}_x(m), \boldsymbol{\Sigma}_x(m) + \boldsymbol{\Sigma}_b(r_m)) \tag{120}$$

Comparing Eq. (120) with Eq. (116), we can tell that the difference between model JUD and front-end JUD is that in front-end JUD only the best component k^* selected in front-end processing is passed to modify the likelihood evaluation during decoding while in model JUD every Gaussian component is associated with a regression-class-dependent transform. Therefore, in model JUD, the distorted feature $\mathbf{y}_t$ is transformed by multiple transforms, similar to CMLLR [120]. However, it differs from CMLLR due to the regression-class-dependent variance term $\boldsymbol{\Sigma}_b(r_m)$.

There are several extensions of model JUD. $\boldsymbol{\Sigma}_b(r_m)$ in Eq. (120) is a full covariance matrix. This brings a large computational cost when evaluating the likelihood. One direct solution is to diagonalize it, however, this solution turns out to have poor performance [244]. Predictive CMLLR (PCMLLR) [211] can be used to avoid the full covariance matrix by applying a CMLLR-like transform in the feature space transformed by model JUD

$$\tilde{\mathbf{y}}_t = \mathbf{A}(r_m)\mathbf{y}_t + \mathbf{b}(r_m). \tag{121}$$

The likelihood for PCMLLR decoding is given by

$$p_\Lambda(\mathbf{y}_t|\theta_t) \approx \sum_m c(m)|\mathbf{A}_p(r_m)||\mathbf{A}(r_m)|\mathcal{N}(\mathbf{A}_p(r_m)\tilde{\mathbf{y}}_t + \mathbf{b}_p(r_m); \boldsymbol{\mu}_x(m), \boldsymbol{\Sigma}_x(m)) \tag{122}$$

$\mathbf{A}_p(r_m)$ and $\mathbf{b}_p(r_m)$ are obtained by using CMLLR, with the statistics obtained from the model JUD transformed feature $\tilde{\mathbf{y}}_t$. With Eq. (122), the clean acoustic model is unchanged.

Another alternative is with VTS-JUD [213], [246] in which the likelihood is computed as

$$p_\Lambda(\mathbf{y}_t|\theta_t) \approx \sum_m c(m)\mathcal{N}(\mathbf{y}_t; \mathbf{H}(r_m)(\boldsymbol{\mu}_x(m) - \mathbf{b}(r_m)), \operatorname{diag}(\mathbf{H}(r_m)(\boldsymbol{\Sigma}_x(m) + \boldsymbol{\Sigma}_b(r_m))\mathbf{H}^T(r_m))) \tag{123}$$

where $\mathbf{H}(r_m) = \mathbf{A}^{-1}(r_m)$. VTS-JUD can be considered as the model space implementation of model JUD, very similar to VTS but with less computational cost. In [247], noise CMLLR is also proposed to extend the conventional CMLLR in Section III-B to reflect additional uncertainty from noisy features by introducing a covariance bias with the same form as Eq. (119). All of model JUD, VTS-JUD, and PCMLLR use VTS in Section V to calculate the regression-class-dependent transforms. If the number of regression classes is identical to the number of Gaussians, it can be proven that all of these methods are the same as VTS. By using regression classes, some computational cost can be saved. For example, in Eq. (123) of VTS-JUD, although it still needs to apply transforms to every Gaussian mean and variance of the clean acoustic model, the cost of calculating transforms is reduced because they are now regression-class-dependent instead of Gaussian-dependent.

Recently, subspace Gaussian mixture models (SGMM) are proposed in [248] with better performance than GMMs. In [249], an extension of JUD when using SGMMs is presented with good improvements in noisy conditions.

D. Missing-Feature Approaches

Missing-feature approaches [250], [251], also known as missing-data approaches, introduce the concept of uncertainty into feature processing. The methods are based on the inherent redundancy in the speech signal: one may still be able to recognize speech effectively even with only a fraction of the spectro-temporal information in the speech signal. They attempt to determine which time-frequency cells are unreliable due to the introduction of noise or other types of interference. These unreliable cells are either ignored or filled in by estimates of their putative values in subsequent processing [252], [253].

There are two major types of missing-feature approaches, namely, feature vector imputation and classifier modification [254]. Feature imputation methods treat unreliable spectral components as missing components and attempt to reconstruct them by utilizing spectrum statistics. There are several typical methods. In correlation-based reconstruction [255], the spectral samples are considered to be the output of a Gaussian wide-sense stationary random process which implies that the means of the spectral vectors and the covariance between spectral components are independent of their positions in the spectrogram. A joint distribution of unreliable components and reliable neighborhood components can be constructed and the reconstruction is then estimated using a bounded MAP estimation procedure. On the other hand, in cluster-based reconstruction [255] the unreliable components are reconstructed only based on the relationships among the components within individual vectors. Soft-mask-based MMSE estimation is similar to cluster-based reconstruction, but with soft masks [256].

In the second category of missing-feature approaches, classifier modification, one may discern between class-conditional imputation and marginalization. In class-conditional imputation, HMM state-specific estimates are derived for the missing components [257]. Marginalization, on the other hand, directly performs optimal classification based on the observed reliable and unreliable components. One extreme and popularly-used case is where only the reliable component is used during recognition.

While with feature vector imputation recognition can be done with features that may be different from the reconstructed log-spectral vectors, it was, until recently, common understanding that state-based imputation and marginalization precluded the use of cepstral features for recognition. This was a major drawback, since the log-spectral features to be used instead exhibit strong spatial correlations, which either resulted in a loss of recognition accuracy at comparable acoustic model size or required significantly more mixture components to achieve competitive recognition rates, compared to cepstral features. However, recently it has been demonstrated that the techniques can be applied in any domain that is a linear transform of log-spectra [258]. In [259] it has been shown that (cepstral) features directly computed from the masked spectrum can outperform imputation techniques, as long as variance normalization is applied on the resulting features. Given the ideal binary mask, recognition on masked speech has been shown to outperform recognition on reconstructed speech. However, mask estimation is never perfect, and as the quality of the mask estimation degrades, recognition on reconstructed speech begins to outperform recognition on masked speech [260].

The most difficult part of missing-feature methods is the accurate estimation of spectral masks which identify unreliable spectrum cells. The estimation can be performed in multiple ways: SNR-dependent estimation [261]–[263], Bayesian estimation [264], [265], and with perceptual criteria [266], [267]. Also, deep neural networks have been employed for supervised learning of the mapping of the features to the desired soft mask target [268].

However, it is impossible to estimate the mask perfectly. Unreliable mask estimation significantly reduces the recognition accuracy of missing-feature approaches [265]. This problem can be remedied to some extent by using soft masks [264], [265], [269] which use a probability to represent the reliability of a spectrum cell. Strictly speaking, missing feature approaches using soft masks can be categorized as uncertainty processing methods, but not those that use binary masks. In [258] it was shown how soft masks can be used with imputation techniques. Further, the estimation of the ideal ratio mask, a soft mask version of the ideal binary mask, was shown to outperform the estimated ideal binary mask in [268]. There is a close link between missing data approaches employing a soft mask and optimal MMSE estimation of the clean speech features, as was shown, among others, in [270].

Instead of treating the mask estimation and the classification as two separate tasks, combining the two promises superior performance. A first approach in this direction was the speech fragment decoder of [253]. The fragment decoder simultaneously searches for the optimal mask and the optimal HMM state sequence. Its initial limitations, which were that ASR had to be carried out in the spectral domain and that the time-frequency fragments were formed prior to the ASR decoding stage and therefore could not benefit form the powerful ASR acoustic models, have been recently overcome [271]–[274]. In [272] ASR-driven mask estimation is proposed. Similarly, the bidirectional speech decoding of [274] also exploits the modeling power of the ASR models for mask estimation. It generates multiple candidate ASR features at every time frame, with each candidate corresponding to a particular back-end acoustic phonetic unit. The ASR decoder then selects the most appropriate candidate via a maximum likelihood criterion. Ultimately one could envision an iterative process, where a baseline recognizer will generate first hypotheses for mask estimation. Using the estimated mask ASR is improved which in turn results in improved mask estimation [272].

To summarize, the state of the art in missing data techniques has matured in recent years and the method has become a high-performance noise robust ASR technique also for medium to large vocabulary tasks.

VII. Compensation with Joint Model Training

Most noise-robust methods assume that the ASR recognizer has been trained from clean speech, and in the testing stage noise robustness methods are used to reduce the mismatch between the clean acoustic model and distorted speech with either feature enhancement or model adaptation techniques. However, it is very difficult to collect clean training data in most practical ASR deployment scenarios. Usually, the training set may contain distorted speech data obtained in all kinds of environments. There are several issues related to the acoustic model trained from multi-style training data. First, the assumption of most noise-robust methods is no longer valid. For example, in the explicit modeling technique discussed in Section V, such as vector Taylor series (VTS), the explicit distortion model assumes that the speech model is only trained from clean data. Another issue is that the trained model is too broad to model the data from all environments. It fails to give a sharp distribution of speech classes because it needs to cover the factors from different environments.

All of these problems can be solved with joint model training which applies the same process at both the training and testing stage so that the same sources of variability can be removed consistently. More specifically, the feature compensation or model adaptation technique used in the test stage is also used in the training stage so that a pseudo-clean acoustic model is obtained in training. Using the criterion of whether the ASR models are trained jointly with the process of feature compensation or model adaptation in the test stage, we can categorize most existing noise-robust techniques in the literature into two broad classes: disjoint and joint model training. The disjoint model training methods are straightforward. We will focus on joint model training in this section.

Among the joint model training methods, the most prominent set of techniques are based on a paradigm called noise adaptive training (NAT) first published in year 2000 [138], which can also be viewed as a hybrid strategy of feature enhancement and model adaptation. One specific example of NAT is the

multi-style training of models in the feature enhanced domain, where noisy training data is first cleaned by feature compensation methods, and subsequently the enhanced features are used to retrain the acoustic model for the evaluation of enhanced test features. This feature-space NAT (fNAT) strategy can achieve better performance than the standard noisy matched scheme, because it applies consistent processing during the training and testing phases while eliminating residual mismatch in an otherwise disjoint training paradigm. The feature compensation can be any form of noise reduction or feature enhancement. The model compensation can be any form of MLE or discriminative training, where it is multi-style training operating on the feature-compensated training data.

fNAT is popular because it is easy to implement and has been shown to be very effective, and hence it has been adopted as one of the two major evaluation paradigms, called multi-style training (after denoising), in the popular series of Aurora tasks. However, fNAT decouples the optimization objective of the feature compensation and model training parts, which are not jointly optimized under a common objective function. In contrast, the model-space NAT (mNAT) methods jointly train a canonical acoustic model and a set of transforms or distortion parameters under MLE or discriminative training criteria, with examples such as source normalization training [275], joint adaptive training (JAT) [241], irrelevant variability normalization (IVN) [121], [276], and VTS-NAT [199], [277]. All of these model-space joint model training methods share the same spirit with speaker adaptive training (SAT) [278], proposed in 1996 for speaker adaptation. One difference between SAT and NAT methods is whether there is a golden target for canonical model learning. In NAT, the golden target is the truly clean speech features or the model trained from it. However, in SAT, there is no such predefined golden speaker as the target.

A. Speaker Adaptive and Source Normalization Training

General adaptation methods, such as MLLR and CMLLR, are initially proposed for speaker adaptation. A speaker-independent acoustic model is obtained from a multi-speaker training set using the standard MLE method. In testing, speaker-dependent transforms are estimated for specific speakers. However, the acoustic model estimated in this way may be a good model for average speakers, but not optimal for any specific speaker. SAT [278] is proposed to train a canonical acoustic model with less inter-speaker variability. A compact HMM model Λ_c and the speaker-dependent transforms $\mathcal{W} = (\mathbf{W}^{(1)}, \mathbf{W}^{(2)} \dots \mathbf{W}^{(R)})$ are jointly estimated from a R-speaker training set by maximizing the likelihood of the training data

$$(\hat{\Lambda}_c, \hat{\mathcal{W}}) = \arg\max_{\Lambda_c, \mathcal{W}} \Pi_{r=1}^{R} \mathcal{L}(\mathbf{Y}^{(r)}; \mathbf{W}^{(r)}(\Lambda_c)), \qquad (124)$$

where $\mathbf{Y}^{(r)}$ is the observation sequence of speaker r. A transform $W^{(r)} = [\mathbf{A}^{(r)}\mathbf{b}^{(r)}]$ for speaker r in the training set maps the compact model Λ_c to a speaker dependent model in the same way as the speaker adaptation methods used in the testing stage. With the compact model Λ_c, the speaker-specific variation in the training stage is reduced and the trained compact model represents the phonetic variation more accurately.

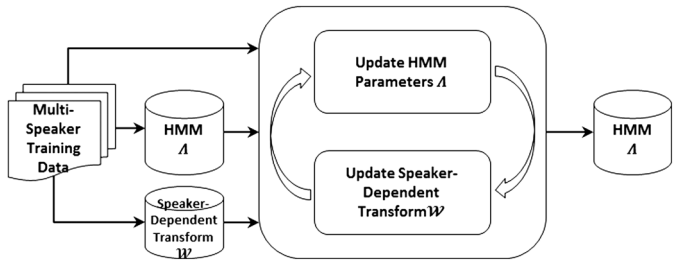


Fig. 3. Speaker adaptive training.

Eq. (124) can be solved with the EM algorithm by maximizing the auxiliary Q function when the MLLR transform is used

$$Q = \sum_{r,t,m} \gamma_t^{(r)}(m) \log \mathcal{N}(\mathbf{y}_t^{(r)}; \mathbf{A}^{(r)}\boldsymbol{\mu}(m) + \mathbf{b}^{(r)}, \boldsymbol{\Sigma}(m)). \qquad (125)$$

As shown in Fig. 3, SAT is done with an iterative two-stage scheme. In the first stage, the auxiliary Q function is maximized with respect to the speaker-dependent transforms $\mathcal{W}$ while keeping the Gaussian model parameters of the compact model Λ_c fixed. By setting the derivative of Q with respect to $\mathcal{W}$ to 0, the solution of $\mathcal{W}$ can be obtained as in Section III-B. In the second stage, the model parameters are updated by maximizing the auxiliary Q function while keeping the speaker-dependent transforms $\mathcal{W}$ fixed. By setting the derivative of Q with respect to $\boldsymbol{\mu}(m)$ and $\boldsymbol{\Sigma}(m)$ to 0, the model parameters can be obtained.

While the SAT formulations are derived with MLLR as the adaptation method, a similar process can also be applied to other adaptation methods, such as CMLLR described in III-B. Although initially proposed to reduce speaker-specific variation in training, SAT can also be used to reduce environment-specific variation when MLLR or CMLLR is used to adapt models in noisy environments.

In some real applications, we need to adapt to a cluster of distortions such as a group of speakers or background noise instead of individual speakers. Source normalization (SNT) training [275] generalizes SAT, by introducing another hidden variable to model distortion sources. SNT subsumes SAT by extending the speaker ID to a hidden variable in training and testing. In SNT the distortion sources (e.g. a speaker) do not have to be tagged; they are discovered by unsupervised training with the EM algorithm. SNT was used to explicitly address environment-specific normalization in 1997 [275]. In [275], an environment can refer to speaker, handset, transmission channel or noise background condition.

MLLR and CMLLR are general adaptation technologies, and cannot work as effectively as the noise-specific explicit modeling methods in noise-robust ASR tasks. Therefore, the SAT-like methods are not as popular as the model space noise adaptive training methods of the next section which are coupled with the explicit distortion model methods in Section V.

B. Model Space Noise Adaptive Training

The model space noise adaptive training (mNAT) scheme is very similar to SAT in Fig. 3. The speaker-dependent transforms $\mathcal{W}$ are replaced with the distortion model $\Phi = (\Phi^{(1)}, \Phi^{(2)}\Phi^{(R)})$, where R is the total number of

training utterances. Every utterance $r = 1R$, has its own utterance-dependent noise, channel, and adapted HMM parameters. However, all utterances share the same set of canonical HMM parameters. Similar to SAT, the mNAT methods are effective when the same model adaptation methods are used in both training and testing stages. The representative mNAT methods are joint adaptive training (JAT) [241], irrelevant variability normalization (IVN) [276], and VTS-NAT [199], [277]. The JAT work uses joint uncertainty decoding (JUD) [192], [240] as its model adaptation scheme. The IVN work uses the VTS algorithm presented in [187] for model adaptation. The VTS-NAT work is coupled with VTS adaptation in [131], [132], which is described in detail in Section V-B1.

VTS-NAT model estimation is also done with an iterative two-stage scheme. In the first stage, the auxiliary Q function is maximized with respect to the utterance-dependent distortion model parameters $\Phi^{(r)}$ while keeping the canonical Gaussian model parameters fixed. The auxiliary Q function can be written as

$$Q = \sum_{t,r,m} \gamma_t^{(r)}(m) \log p_{\hat{\Lambda}^{(r)}} \left(\mathbf{y}_t^{(r)} | m \right), \qquad (126)$$

where $\hat{\Lambda}^{(r)}$ denotes the adapted model for utterance r. Comparing this with the auxiliary function in Eq. (86), an additional term r is summed in Eq. (126) to include all the training utterances. The solution is the same as in Section V-B1.

In the second stage, the canonical model parameters are updated by maximizing the auxiliary Q function while keeping the utterance-dependent distortion model parameters $\Phi^{(r)}$ fixed. The mean parameters of the canonical model are obtained by taking the derivative of Q with respect to them and setting the result to zero.

Similar to the solution of noise variance update in VTS, a Newton's method can be used to update the model variance [199]. All of the joint model training techniques learn the canonical model to represent the pseudo-clean speech model, and the transforms are then used to represent the non-linguistic variability such as environmental variations. It has been well established that joint training methods achieve consistent improvement over the disjoint training methods. The latter are much easier to implement, and easier to train the acoustic model parameters alone without making them compact and without removing the non-linguistic variability.

Coupled with model JUD instead of VTS, JAT [241] is another variation of NAT. Similar to VTS-NAT, the adaptive transform in JAT is parameterized, and its parameters are jointly trained with the HMM parameters by the same kind of maximum likelihood criterion. While most noise adaptive training studies are based on the maximum likelihood criterion, discriminative adaptive training can be used to further improve accuracy [279]. To do so, standard MLE-based noise adaptive training [241] is first performed to get the HMM parameters and distortion model parameters. Then, the distortion model parameters are fixed and the discriminative training criterion is applied to further optimize the HMM parameters. While JAT is initially proposed to handle GMMs, it is extended in [208] to work with subspace GMMs to get further improvement.

The idea of irrelevant variability normalization (IVN) is a very general concept. The argument is that HMMs trained from a large amount of diversified data, which consists of different speakers, acoustic environments, channels etc., may tend to fit the variability of data irrelevant to phonetic classification. The term IVN is proposed in [280] to build a better decision tree that has better modeling capability and generalizability by removing the speaker factors during the decision tree building process. Then from 2002, IVN is widely used as a noise-robustness method for jointly training the front-end and back-end together for stochastic vector mapping [281], which maps the corrupted speech feature to a clean speech feature by a transform. Every environment can have a bias vector [281], [282], or one environment-dependent transform [283], or even multiple environment transforms [284].

Although some forms of IVN are similar to SAT, IVN is designed for noise robustness by using environment-dependent transforms and biases to map the corrupted speech feature to the clean speech feature. In [276], IVN is further linked with VTS (VTS-IVN) by using explicit distortion modeling to characterize the distortion caused by noise.

VIII. Summary and Future Directions

In this paper, we have provided an overview of noise-robust ASR techniques guided by a unified mathematical framework. Since noise robustness for ASR is a very large subject, a number of topics had to be excluded to keep the overview reasonably concise. Among the topics excluded are robustness against room reverberation, blind speaker separation, microphone array processing, highly nonstationary noise, and voice activity detection. The included topics have covered major core techniques in the field, many of which are currently exploited in modern speech recognition systems.

To offer insight into the distinct capabilities of these techniques and their connections, we have conducted this overview using the taxonomy-oriented approach. We have used five key attributes—feature vs. model domain processing, explicit vs. implicit distortion modeling, use of prior knowledge about distortion or otherwise, deterministic vs. uncertain processing, and joint vs. disjoint training, to organize the vast amount of material and to demonstrate the commonalities and differences among the plethora of noise-robust ASR methods surveyed in this paper. We conclude this paper by summarizing the methods surveyed in this paper in Table II using the five distinct attributes discussed in this study. Note that the column with the heading "explicit modeling" refers to the use of an explicit model for the physical relation between clean and distorted speech. As such, signal processing methods, such as PLP and RASTA, while having auditory modeling inside, are not classified as explicit distortion modeling. CMN is considered to be an explicit modeling method because it can remove the convolutive channel effect while CMVN is considered to be a representative of implicit modeling because cepstral variance normalization does not explicitly address any distortion. We classify observation uncertainty and front-end uncertainty decoding as hybrid (domain) methods in Table II because the uncertainty is obtained in the feature space and is then passed

TABLE II
A SUMMARY OF THE REPRESENTATIVE METHODS IN NOISE-ROBUST ASR SURVEYED IN THIS PAPER. THEY ARE ARRANGED ALPHABETICALLY

Methods	Time of publication	Model vs. feature domain	Explicit vs. implicit distortion modeling	Use prior knowledge about distortion or not	Deterministic vs. uncertainty processing	Joint vs. disjoint training
ANN-HMM hybrid systems [63]	1994	feature	implicit	not use	deterministic	disjoint
Bayesian predictive classification (BPC) [231]	1997	model	implicit	not use	uncertainty	disjoint
bottle-neck feature [68]	2007	feature	implicit	not use	deterministic	disjoint
cepstral mean normalization (CMN) [81]	1974	feature	explicit	not use	deterministic	disjoint
cepstral mean and variance normalization (CMVN) [82]	1998	feature	implicit	not use	deterministic	disjoint
constrained MLLR (CMLLR) [120]	1998	both	implicit	not use	deterministic	disjoint
context-dependent deep neural network HMM (CD-DNN-HMM) [71], [73]	2010	feature	implicit	not use	deterministic	disjoint
empirical cepstral compensation [5], [135], [136]	1990	feature	implicit	use	deterministic	disjoint
exemplar-based reconstruction use non-negative matrix factorization [163], [164]	2010	feature	explicit	use	deterministic	disjoint
eigenvoice [154]	2000	model	implicit	use	deterministic	disjoint
ETSI advanced front-end (AFE) [108]	2002	feature	explicit	not use	deterministic	disjoint
feature space noise adaptive training (NAT) [138]	2000	feature	implicit	both	deterministic	joint
front-end uncertainty decoding [239], [240]	2002	hybrid	implicit	use	uncertainty	disjoint
histogram equalization method (HEQ) [83]	2003	feature	implicit	use	deterministic	disjoint
irrelevant variability normalization [276], [281]	2002	both	both	both	deterministic	joint
joint adaptive training (JAT) [241]	2007	hybrid	explicit	not use	uncertainty	joint
joint uncertainty decoding (JUD) [240]	2005	hybrid	explicit	not use	uncertainty	disjoint
Mel-warped Wiener filtering [109], [114]	2002	feature	explicit	not use	deterministic	disjoint
maximum likelihood linear regression (MLLR) [119]	1995	model	implicit	not use	deterministic	disjoint
missing feature [250], [252], [255]	1994	feature	implicit	not use	uncertainty	disjoint
multi-style training [21]	1987	model	implicit	use	deterministic	disjoint
observation uncertainty [235], [236], [238]	2002	hybrid	implicit	both	uncertainty	disjoint
parallel model combination (PMC) [133]	1995	model	explicit	not use	deterministic	disjoint
perceptual linear prediction (PLP) [46]	1985	feature	implicit	not use	deterministic	disjoint
relative spectral processing (RASTA) [49], [285]	1991	feature	implicit	not use	deterministic	disjoint
speaker adaptive training (SAT) [278]	1996	model	implicit	not use	deterministic	joint
spectral subtraction [103]	1979	feature	explicit	not use	deterministic	disjoint
stereo piecewise linear compensation for environment (SPLICE) [138]	2000	feature	implicit	use	deterministic	disjoint
TANDEM [64]	2000	feature	implicit	not use	deterministic	disjoint
TempoRAL Pattern (TRAP) processing [65]	1998	feature	implicit	not use	deterministic	disjoint
unscented transform (UT) [216], [219]	2006	model	explicit	not use	deterministic	disjoint
variable-parameter HMM (VPHMM) [176]	2007	model	implicit	use	deterministic	disjoint
vector Taylor series (VTS) model adaptation [132], [134], [185]	1996	model	explicit	not use	deterministic	disjoint
VTS feature enhancement [134], [195], [198]	1996	feature	explicit	not use	deterministic	disjoint
VTS-JUD [213], [246]	2009	model	explicit	not use	deterministic	disjoint
VTS-NAT [199], [277]	2009	model	explicit	not use	deterministic	joint
Wiener filter [106]	1979	feature	explicit	not use	deterministic	disjoint

to the back-end recognizer by modifying the model covariance with a bias.

In our survey we note that some methods were proposed a long time ago, and they were revived when more advanced technologies were adopted. As an example, VTS was first proposed in 1996 [134] for both model adaptation and feature enhancement. But VTS has only quite recently demonstrated an advantage with the advanced online re-estimation for all the distortion parameters [131], [132]. Another example is the famous Wiener filter, proposed as early as 1979 [106] to improve the performance on noisy speech. Only after some 20 years, in 2002, two-stage Mel-warped Wiener filtering was proposed to boost the performance of Wiener filtering in several key aspects, and has become the main component of the ETSI advanced front-end. Furthermore, ANN-HMM hybrid systems [63] were studied in the 1990s, and again only after 20 years, expanded to deep architectures with improved learning algorithms to achieve much greater success in ASR and in noise robustness in particular [71], [73]. Hence, understanding current well-established technologies is important for providing a foundation for further technology development, and is one of the goals of this overview paper.

In the early years of noise-robust ASR research, the focus was mostly on feature-domain methods due to their efficient runtime implementation. Runtime efficiency is always a factor when it comes to deployment of noise-robustness technologies. A good example is CMLLR which can be effectively realized in the feature space with very small cost although it can also be implemented in the model space with a much larger cost by transforming all of the acoustic model parameters instead of very limited ones (e.g. a single vector per frame) as in MLLR. Feature normalization methods come with very low cost and hence are widely used. But they address noise-robustness problems in an implicit way. In contrast, spectral subtraction, Wiener filtering, and VTS feature enhancement use explicit distortion modeling to remove noise, and are more effective. Note that most feature-domain methods are decoupled from the ASR objective function, hence they may not perform as well as

model-domain methods. In contrast, while typically achieving higher accuracy than feature-domain methods, model-domain methods usually incur significantly larger computational costs. With increasing computational power, research on model-domain methods is expected to become increasingly active. With the introduction of bottleneck features enabled by the DNN, and the use of DNN technology itself as acoustic models for ASR, we also expect increasing activity in neural-network-based noise-robustness methods for ASR in the coming years.

As we have reviewed in this paper, many of the model and feature domain methods use explicit distortion models to describe the physical relationship between clean and distorted or noisy speech. Because the physical constraints are explicitly represented in the models, the explicit distortion modeling methods require only a relatively small number of distortion parameters to be estimated. In contrast to the general-purpose techniques, they also exhibit high performance due to the explicit exploitation of the distorted speech generation process. One of the most important explicit distortion modeling techniques is VTS model adaptation [131], [132], which has been discussed in detail in this paper. However, all of the model adaptation methods using such explicit distortion models rely on validity of the physical constraints expressed by the particular features used. Even with a simple feature normalization technology such as CMN, the distortion model such as in Eq. (14) is no longer valid. As a result, model adaption using explicit distortion modeling cannot be easily combined with all of the feature post-processing technologies, such as CMN, HLDA, and fMPE etc. One solution is to use VTS feature enhancement, which still utilizes explicit distortion modeling and can also be combined with other feature post-processing technologies, despite the small accuracy gap between VTS feature enhancement and VTS model adaptation [198]. Another advantage of VTS feature enhancement is that it can reduce the computational cost of VTS model adaptation, which is always an important concern in ASR system deployment. JUD [192], PCMLLR [211], and VTS-JUD [246] have been developed also addressing such concerns. As shown in Section V, better distortion modeling results in better algorithm performance, but also incurs a large computation cost (e.g., UT [216], [219]). While most adaptation is a one-to-one mapping between the clean Gaussian and the distorted Gaussian due to easy implementation, recent work has appeared covering distributional mappings between the GMMs [220]. In conclusion, research in explicit distortion modeling is expected to grow, and an important direction will be how to combine better modeling with runtime efficiency and how to make it capable of working with other feature processing methods.

Without explicit distortion modeling, it is very difficult to predict the impact of noise on clean speech during testing if the acoustic model is trained only from clean speech. The more prior knowledge of that impact we have, the more we can better recognize the corrupted speech during testing. The methods utilizing prior knowledge about distortion discussed in this paper are motivated by such reasoning. Methods like SPLICE learn the environment-dependent mapping from corrupted speech to clean speech. The extreme case is exemplar-based reconstruction with NMF, which restores cleaned speech by constructing the noisy speech with pre-trained clean speech and noise exemplars and by keeping only the clean speech exemplars. However, the main challenge in the future direction is that methods utilizing prior knowledge need to preform also well for unseen environments, and we expect more research in this direction in the future.

Since neither feature enhancement nor model adaptation is perfect, there always exists uncertainty in feature or model space, and uncertainty processing has been designed to address this issue as reviewed in this paper. The initial study in the model space modified the decision rule to use the minimax rule or the BPC rule. Although the mathematics is well grounded, the computational cost is very large and it is difficult to define the model neighborhood. Research was subsequently switched to the feature space, resulting in the methods exploiting observation uncertainty and JUD. The latter also serves as an excellent approximation to VTS with much lower computational cost. Future research in uncertainty processing is expected to focus on the feature space, and on combining the technique with advances in other areas such as explicit distortion modeling.

Joint training is a good way to obtain canonical acoustic models. Feature-space NAT is a common practice that is now widely used while model-space NAT is much harder to develop and deploy partly because of the difficulty in finding closed-form solutions in model learning and because of the computational complexity. Despite the difficulty, joint training is more promising in the long run because it removes the irrelevant variability to phonetic classification during training, and multi-style training data is easier to obtain than clean training data in real world applications. Better integrated algorithm design, improved joint optimization of model and transform parameters, and clever use of metadata as labels for the otherwise hidden "distortion condition" variables are all promising future research directions.

By comparing the methods in Table II, we clearly see the advantages of explicit distortion modeling, using prior knowledge, uncertainty, and joint training methods over their counterparts. When developing a noise-robust ASR method, their combinations should be explored. For real-world applications, there are also some other factors to consider. For example, there is always a tradeoff between high accuracy and low computational cost. Special attention should also be paid to non-stationary noise. Some methods such as NMF can handle non-stationary noise very well because the noise exemplars are extracted from a large dictionary which can consist of different types of noise. The effectiveness of many frame-by-frame feature compensation methods (e.g., spectral subtraction and Wiener filtering) depends on whether the noise tracking module is good at tracking non-stationary noise. Some methods such as the standard VTS may not directly handle non-stationary noise well because they assume the noise is Gaussian distributed. This problem can be solved by relaxing the assumption using a time-dependent noise estimate (e.g., [286]).

Finally, the recent acoustic modeling technology, CD-DNN-HMM, brings new challenges to conventional noise-robustness technologies. We classify CD-DNN-HMM as a feature-based noise robust ASR technology since its layer-by-layer setup provides a feature extraction strategy that automatically derives

powerful noise-resistant features from primitive raw data for senone classification. In [77], the CD-DNN-HMM trained with multi-style data easily matches the state-of-the-art performance obtained with complicated conventional noise-robustness technology on GMM systems [227]. With deep and wide hidden layers, the DNN provides a very strong normalization to heterogeneous data [77], [78], [287]. The noise, channel, and speaker factors may already be well normalized by the complex nonlinear transform inside the DNN. In other words, the layer-by-layer feature extraction strategy in deep learning provides an opportunity to automatically derive powerful features from primitive raw data for HMM state classification. However, this does not mean that the noise-robustness technologies are not necessary when used together with CD-DNN-HMM. It is shown in [288]–[290] that a robust front-end is still helpful if the CD-DNN-HMM is trained with clean data, and tested with noisy data. In a multi-style training setup, although some robust front-ends cannot benefit DNNs [77], [289], VTS with explicit distortion modeling and DOLPHIN (dominance based locational and power-spectral characteristics integration) are still very useful to improve the ASR performance [288], [290]. One possible reason is that the nonlinear distortion model used in VTS and the spatial information used in DOLPHIN are not available to the DNN. Therefore, one potential way to work with a CD-DNN-HMM is to incorporate technologies utilizing the information not explicitly exploited in DNN training. The explicit modeling technologies such as VTS should work very well. The DNN also makes it easy to work on all kinds of acoustic features, which may be hard for a GMM. For example, log-filter-bank features are usually not used as the input to a GMM because of the correlation among feature dimensions. In [78], [291], it is shown that log-filter-bank features are significantly better than the widely-used MFCCs when used as the input to the DNN. And in [292], the filter bank is replaced with a filter bank layer that is learned jointly with the rest of DNN by taking the linear frequency power spectrum as the input. Furthermore, some of the noise-robustness methods, as surveyed in this paper, have the underlying assumption that a GMM is used for state likelihood evaluation. If the CD-DNN-HMM is used, this assumption is no longer valid. One potential solution is to use the DNN-derived bottleneck features in a GMM-HMM system, thereby utilizing both the power of DNN nonlinear normalization and the GMM assumption validating many of the currently successful noise-robustness approaches described in this paper. The impact of using the DNN and DNN-induced bottleneck features on noise-robustness ASR deserve intensive research.

ACKNOWLEDGMENT

The authors want to thank Jon Grossmann for proofreading the manuscript.

REFERENCES

[1] X. Huang, A. Acero, and H. W. Hon, *Spoken Language Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[2] L. Deng and D. O'Shaughnessy, *Speech Processing—A Dynamic and Optimization-Oriented Approach*. New York, NY, USA: Marcel Dekker, 2003.

[3] X. He and L. Deng, "Speech-centric information processing: An optimization-oriented approach," *Proc. IEEE*, vol. 101, no. 5, pp. 1116–1135, May 2013.

[4] L. Deng, K. Wang, H. Hon, A. Acero, and X. Huang, "Distributed speech processing in mipad's multimodal user interface," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 8, pp. 605–619, Nov. 2002.

[5] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*. Cambridge, U.K.: Cambridge Univ. Press, 1993.

[6] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Commun.*, vol. 16, pp. 261–291, 1995.

[7] C. H. Lee, "On stochastic feature and model compensation approaches to robust speech recognition," *Speech Commun.*, vol. 25, pp. 29–47, 1998.

[8] Q. Huo and C. H. Lee, "Robust speech recognition based on adaptive classification and decision strategies," *Speech Commun.*, vol. 34, no. 1–2, pp. 175–194, 2001.

[9] J. Droppo and A. Acero, "Environmental robustness," in *Handbook of Speech Process.*, J. Benesty, M. M. Sondhi, and Y. Huang, Eds. New York, NY, USA: Springer, 2008, ch. 33.

[10] L. Deng, "Front-end, back-end, and hybrid techniques for noise-robust speech recognition," in *Robust Speech Recognition of Uncertain or Missing Data: Theory and Application*. New York, NY, USA: Springer, 2011, pp. 67–99.

[11] R. Haeb-Umbach, "Uncertainty decoding and conditional bayesian estimation," in *Robust Speech Recognition of Uncertain or Missing Data: Theory and Application*. New York, NY, USA: Springer, 2011, pp. 9–34.

[12] M. J. F. Gales, "Model-based approaches to handling uncertainty," in *Robust Speech Recognition of Uncertain or Missing Data: Theory and Application*. New York, NY, USA: Springer, 2011, pp. 101–125.

[13] K. Kumatani, J. W. McDonough, and B. Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 127–140, Nov. 2012.

[14] D. Kolossa and R. Haeb-Umbach, *Robust speech recognition of uncertain or missing data: theory and applications*. New York, NY, USA: Springer, 2011.

[15] T. Virtanen, R. Singh, and B. Raj, *Techniques for noise robustness in automatic speech recognition*. New York, NY, USA: Wiley, 2012.

[16] S. B. Davis and P. Mermelstein, "Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Audio, Speech, Signal Process.*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[17] I. Cohen and S. Gannot, "Spectral enhancement methods," in *Handbook of Speech Process.*, J. Benesty, M. M. Sondhi, and Y. Huang, Eds. New York, NY, USA: Springer, 2008, ch. 44.

[18] L. Deng, J. Droppo, and A. Acero, "Estimating cepstrum of speech under the presence of noise using a joint prior of static and dynamic features," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 12, no. 3, pp. 218–233, May 2004.

[19] L. Deng, J. Droppo, and A. Acero, "Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 2, pp. 133–143, Mar. 2004.

[20] V. Leutnant and R. Haeb-Umbach, "An analytic derivation of a phase-sensitive observation model for noise-robust speech recognition," in *Proc. Interspeech*, 2009, pp. 2395–2398.

[21] R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word speech recognition," in *Proc. ICASSP*, 1987, pp. 705–708.

[22] V. Ion and R. Haeb-Umbach, "A novel uncertainty decoding rule with applications to transmission error robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 5, pp. 1047–1060, Jul. 2008.

[23] A. Varga, H. J. M. Steeneken, M. Tomlinson, and D. Jones, "The NOISEX–92 study on the effect of additive noise on automatic speech recognition," Tech. Rep., DRA Speech Research Unit 1992.

[24] A. Schmidt-Nielsen, E. Marsh, J. Tardelli, P. Gatewood, E. Kreamer, T. Tremain, C. Cieri, and J. Wright, "Speech in noisy environments (SPINE) evaluation audio," Linguistic Data Consortium 2000.

[25] H. Christensen, J. Barker, N. Ma, and P. Green, "The CHiME corpus: A resource and a challenge for computational hearing in multisource environments.," in *Proc. Interspeech*, 2010, pp. 1918–1921.

[26] H. G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR*, 2000.

[27] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, S. Euler, and J. Allen, "SPEECHDAT-CAR. a large speech database for automotive environments.," in *Proc. LREC*, 2000.

[28] N. Parihar and J. Picone, "Aurora working group: DSR front end LVCSR evaluation AU/384/02," Inst. for Signal and Inf., Process. Mississippi State Univ., 2002, Tech. Rep..

[29] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Workshop Speech Natural Lang.*, 1992, pp. 357–362, Association for Computational Linguistics.

[30] H. G. Hirsch, "Aurora-5 experimental framework for the performance evaluation of speech recognition in case of a hands-free speech input in noisy environments," Niederrhein Univ. of Applied Sciences, 2007.

[31] S. Makino, T.-W. Lee, and H. Sawada, *Blind Speech Separation*. New York, NY, USA: Springer, 2007.

[32] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. New York, NY, USA: Springer, 2007.

[33] P. A. Naylor and N. D. Gaubitch, *Speech Dereverberation*. New York, NY, USA: Springer, 2010.

[34] M. Woelfel and J. McDonough, *Distant Speech Recognition*. New York, NY, USA: Wiley, 2009.

[35] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann, "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Processing Mag.*, vol. 29, no. 6, pp. 114–126, Nov. 2012.

[36] R. Haeb-Umbach and A. Krueger, "Reverberant speech recognition," in *Noise Robustness in Automatic Speech Recognition*, T. Virtanen, R. Singh, and B. Raj, Eds. New York, NY, USA: Wiley, 2012.

[37] P. Smaragdis, "Extraction of speech from mixture signals," in *Noise Robustness in Automatic Speech Recognition*, T. Virtanen, R. Singh, and B. Raj, Eds. New York, NY, USA: Wiley, 2012.

[38] S. Furui, L. Deng, M. Gales, H. Ney, and K. Tokuda, "Special issue on fundamental technologies in modern speech recognition," *IEEE Signal Process. Mag*, vol. 29, no. 6, Nov. 2012.

[39] Z. Zhang, Z. Liu, and M. Sinclair etc., "Multi-sensory microphones for robust speech detection, enhancement and recognition," in *Proc. ICASSP*, 2004, pp. 781–784.

[40] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 11, no. 6, pp. 568–580, Nov. 2003.

[41] T. Yoshioka and T. Nakatani, "Noise model transfer: Novel approach to robustness against nonstationary noise," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2182–2192, Oct. 2013.

[42] M. Souden, S. Araki, K. Kinoshita, T. Nakatani, and H. Sawada, "A multichannel mmse-based framework for speech source separation and noise reduction," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 9, pp. 1913–1928, Sep. 2013.

[43] C. W. Han, S. J. Kang, and N. S. Kim, "Reverberation and noise robust feature compensation based on imm," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 8, pp. 1598–1611, Aug. 2013.

[44] S. Mosayyebpour, M. Esmaeili, and T. A. Gulliver, "Single-microphone early and late reverberation suppression in noisy speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 2, pp. 322–335, Feb. 2013.

[45] C. Demir, M. Saraclar, and A. T. Cemgil, "Single-channel speech-music separation for robust ASR with mixture models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 4, pp. 725–736, Apr. 2013.

[46] H. Hermansky, B. A. Hanson, and H. Wakita, "Perceptually based linear predictive analysis of speech," in *Proc. ICASSP*, 1985, vol. I, pp. 509–512.

[47] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *JASA*, vol. 87, no. 4, pp. 1738–1752, 1990.

[48] B. A. Hanson and T. H. Applehaum, "Subband or cepstral domain filtering for recognition of Lombard and channel-distorted speech," in *Proc. ICASSP*, 1993, vol. II, pp. 79–82.

[49] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "Compensation for the effect of communication channel in auditory-like analysis of speech (RASTA-PLP)," in *Proc. Eur. Conf. Speech Technol.*, 1991, pp. 1367–1370.

[50] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, Oct. 1994.

[51] C. Avendano, S. van Vuuren, and H. Hermansky, "Data-based RASTA-like filter design for channel normalization in ASR," in *Proc. ICSLP*, 1996, pp. 2087–2090.

[52] D. S. Kim, S. Y. Lee, and R. M. Kil, "Auditory processing of speech signals for robust speech recognition in real-world noisy environments," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 1, pp. 55–69, Jan. 1999.

[53] A. M. A. Ali, J. Van der Spiegel, and P. Mueller, "Robust auditory-based speech processing using the average localized synchrony detection," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 279–292, Jul. 2002.

[54] U. H. Yapanel and J. H. L. Hansen, "A new perceptually motivated MVDR-based acoustic front-end (PMVDR) for robust automatic speech recognition," *Speech Commun.*, vol. 50, no. 2, pp. 142–152, 2008.

[55] C. Kim and R. M. Stern, "Feature extraction for robust speech recognition based on maximizing the sharpness of the power distribution and on power flooring," in *Proc. ICASSP*, 2010, pp. 4574–4577.

[56] F. Müller and A. Mertins, "Contextual invariant-integration features for improved speaker-independent speech recognition," *Speech Commun.*, vol. 53, no. 6, pp. 830–841, 2011.

[57] N. Moritz, J. Anemuller, and B. Kollmeier, "Amplitude modulation spectrogram based features for robust speech recognition in noisy and reverberant environments," in *Proc. ICASSP*, 2011, pp. 5492–5495.

[58] Y. Shao, S. Srinivasan, Z. Jin, and D. Wang, "A computational auditory scene analysis system for speech segregation and robust speech recognition," *Comput. Speech Lang.*, vol. 24, no. 1, pp. 77–93, 2010.

[59] A. Fazel and S. Chakrabartty, "Sparse auditory reproducing kernel (SPARK) features for noise-robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 4, pp. 1362–1371, May 2012.

[60] N. Moritz, M. Schädler, K. Adiloglu, B. Meyer, T. Jürgens, T. Gerkmann, B. Kollmeier, S. Doclo, and S. Goetze, "noise robust distant automatic speech recognition utilizing nmf based source separation and auditory feature extraction," in *Proc. 2nd CHiME Workshop Mach. Listen. Multisource Environ.*, 2013.

[61] R. Stern and N. Morgan, "Features based on auditory physiology and perception," in *Techniques for Noise Robustness in Automatic Speech Recognition*. New York, NY, USA: Wiley, 2012.

[62] Y. H. Chiu, B. Raj, and R. M. Stern, "Learning-based auditory encoding for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 3, pp. 900–914, Mar. 2012.

[63] H. Bourlard and N. Morgan, *Connectionist speech recognition - A Hybrid approach*. Norwell, MA, USA: Kluwer, 1994.

[64] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000, vol. 3, pp. 1635–1638.

[65] H. Hermansky and S. Sharma, "TRAPs - classifiers of temporal patterns," in *Proc. ICSLP*, 1998.

[66] P. Jain, H. Hermansky, and B. Kingsbury, "Distributed speech recognition using noise-robust MFCC and TRAPS-estimated manner features," in *Proc. Interspeech*, 2002.

[67] B. Chen, Q. Zhu, and N. Morgan, "Learning long-term temporal features in LVCSR using neural networks," in *Proc. Interspeech*, 2004.

[68] F. Grézl, M. Karafiát, S. Kontár, and J. Cernocký, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. ICASSP*, 2007, vol. IV, pp. 757–760.

[69] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Commun.*, vol. 26, no. 4, pp. 283–297, 1998.

[70] Z. Tüske, R. Schlüter, N. Hermann, and M. Sundermeyer, "Context-dependent MLPs for LVCSR: Tandem, hybrid or both?," in *Proc. Interspeech*, 2012, pp. 18–21.

[71] D. Yu, L. Deng, and G. Dahl, "Roles of pretraining and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2010.

[72] T. N. Sainath, B. Kingsbury, H. Soltau, and B. Ramabhadran, "Optimization techniques to improve training speed of deep neural networks for large speech tasks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 11, pp. 2267–2276, Nov. 2013.

[73] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[74] G. Hinton, L. Deng, D. Yu, and G. Dahl *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[75] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Speech Audio Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[76] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novák, and A. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc. ASRU*, 2011, pp. 30–35.

[77] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7398–7402.

[78] J. Li, D. Yu, J. T. Huang, and Y. Gong, "Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM," in *Proc. IEEE SLT*, 2012, pp. 131–136.

[79] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust ASR," in *Proc. ICASSP*, 2012, pp. 4085–4088.

[80] 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*, 2012, pp. 22–25.

[81] B. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *JASA*, vol. 55, pp. 1304–1312, 1974.

[82] O. Viikki, D. Bye, and K. Laurila, "A recursive feature vector normalization approach for robust speech recognition in noise," in *Proc. ICASSP*, 1998, pp. 733–736.

[83] S. Molau, F. Hilger, and H. Ney, "Feature space normalization in adverse acoustic conditions," in *Proc. ICASSP*, 2003, vol. 1, pp. 656–659.

[84] A. Acero and X. Huang, "Augmented cepstral normalization for robust speech recognition," in *Proc. IEEE Workshop Autom. Speech Recogn.*, 1995.

[85] A. Anastasakos, F. Kubala, J. Makhoul, and R. Schwartz, "Adaptation to new microphones using tied-mixture normalization," in *Proc. ICASSP*, 1994, vol. I, pp. 433–436.

[86] C. P. Chen and J. Bilmes, "Mva processing of speech features," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 257–270, Jan. 2007.

[87] Y. H. Suk, S. H. Choi, and H. S. Lee, "Cepstrum third-order normalization method for noisy speech recognition," *Electron. Lett.*, vol. 35, no. 7, pp. 527–528, 1999.

[88] C. W. Hsu and L. S. Lee, "Higher order cepstral moment normalization for improved robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 2, pp. 205–220, Feb. 2009.

[89] X. Xiao, J. Li, C. E. Siong, and H. Li, "Feature normalization using structured full transforms for robust speech recognition," in *Proc. Interspeech*, 2011, pp. 693–696.

[90] S. Dharanipragada and M. Padmanabhan, "A nonlinear unsupervised adaptation technique for speech recognition," in *Proc. ICASSP*, 2000, pp. 556–559.

[91] A. de la Torre, A. M. Peinado, J. C. Segura, J. L. Perez-Cordoba, M. C. Benitez, and A. J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 355–366, 2005.

[92] F. Hilger and H. Ney, "Quantile based histogram equalization for noise robust large vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 3, pp. 845–854, May 2006.

[93] S. Molau, M. Pitz, and H. Ney, "Histogram based normalization in the acoustic feature space," in *Proc. ASRU*, 2001, pp. 21–24.

[94] J. C. Segura, C. Benitez, A. Torre, A. J. Rubio, and J. Ramirez, "Cepstral domain segmental nonlinear feature transformations for robust speech recognition," *IEEE Signal Process. Lett.*, vol. 11, no. 5, pp. 517–520, May 2004.

[95] F. Hilger and H. Ney, "Quantile based histogram equalization for noise robust speech recognition," in *Proc. Eurospeech*, 2001, vol. 2, pp. 1135–1138.

[96] S. H. Lin, Y. M. Yeh, and B. Chen, "Exploiting polynomial-fit histogram equalization and temporal average for robust speech recognition," in *Proc. ICSLP*, 2006, pp. 2522–2525.

[97] B. Liu, L. Dai, J. Li, and R. H. Wang, "Double gaussian based feature normalization for robust speech," in *Proc. Int. Symp. Chinese Spoken Lang. Process.*, 2004, pp. 705–708.

[98] L. García, J. C. Segura, J. Ramrez, A. de la Torre, and C. Bentez, "Parametric nonlinear feature equalization for robust speech recognition," in *Proc. ICASSP*, 2006, vol. I, pp. 529–532.

[99] Y. Suh, M. Ji, and H. Kim, "Probabilistic class histogram equalization for robust speech recognition," *IEEE Signal Process. Lett.*, vol. 14, no. 4, pp. 287–290, Apr. 2007.

[100] L. García, C. B. Ortuzar, A. de la Torre, and J. C. Segura, "Class-based parametric approximation to histogram equalization for ASR," *IEEE Signal Process. Lett.*, vol. 19, no. 7, pp. 415–418, Jul. 2012.

[101] X. Xiao, E. S. Chng, and H. Li, "Attribute-based histogram equalization (HEQ) and its adaptation for robust speech recognition," in *Proc. Interspeech*, 2013, pp. 876–880.

[102] S. N. Tsai and L. S. Lee, "A new feature extraction front-end for robust speech recognition using progressive histogram equalization and mulit-eigenvector temporal filtering," in *Proc. ICSLP*, 2004, pp. 165–168.

[103] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans Audio, Speech, Signal Process.*, vol. 27, no. 2, pp. 113–120, Apr. 1979.

[104] J. Taghia, J. Taghia, N. Mohammadiha, J. Sang, V. Bouse, and R. Martin, "An evaluation of noise power spectral density estimation algorithms in adverse acoustic environments," in *Proc. ICASSP*, 2011, pp. 4640–4643.

[105] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by additive noise," in *Proc. ICASSP*, 1979, pp. 208–211.

[106] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proc. IEEE*, vol. 67, no. 12, pp. 1586–1604, Dec. 1979.

[107] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

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

[109] D. Macho, L. Mauuary, B. Noé, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a noise-robust DSR front-end on aurora databases," in *Proc. ICSLP*, 2002, pp. 17–20.

[110] Y. M. Cheng and D. Macho, "SNR-dependent waveform processing for robust speech recognition," in *Proc. ICASSP*, 2001, vol. 1, pp. 305–308.

[111] L. Mauuary, "Blind equalization in the cepstral domain for robust telephone based speech recognition," *EUSPICO*, vol. 1, pp. 359–363, 1998.

[112] A. Agarwal and Y. M. Cheng, "Two-stage mel-warped wiener filter for robust speech recognition," in *Proc. ASRU*, 1999, pp. 67–70.

[113] B. Noé, J. Sienel, D. Jouvet, L. Mauuary, L. Boves, J. De Veth, and F. de Wet, "Noise reduction for noise robust feature extraction for distributed speech recognition," in *Proc. Eurospeech*, 2001.

[114] J. Li, B. Liu, R. H. Wang, and L. Dai, "A complexity reduction of ETSI advanced front-end for DSR," in *Proc. ICASSP*, 2004, vol. 1, pp. 61–64.

[115] M. Padmanabhan, G. Saon, and G. Zweig, "Lattice-based unsupervised MLLR for speaker adaptation," in *Proc. ISCA ITRW ASR*, 2000, pp. 128–131.

[116] K. Shinoda and C. H. Lee, "A structural bayes approach to speaker adaptation," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 3, pp. 276–287, Mar. 2001.

[117] O. Siohan, C. Chesta, and C. H. Lee, "Joint maximum a posteriori adaptation of transformation and HMM parameters," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 4, pp. 417–428, May 2001.

[118] O. Siohan, T. A. Myrvoll, and C. H. Lee, "Structural maximum a posteriori linear regression for fast HMM adaptation," *Computer, Speech, Lang.*, vol. 16, no. 1, pp. 5–24, 2002.

[119] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput., Speech, Lang.*, vol. 9, no. 2, pp. 171–185, 1995.

[120] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput., Speech, Lang.*, vol. 12, pp. 75–98, 1998.

[121] J. Wu and Q. Huo, "Supervised adaptation of MCE-trained CDHMMs using minimum classification error linear regression," in *Proc. ICASSP*, 2002, vol. I, pp. 605–608.

[122] X. He and W. Chou, "Minimum classification error linear regression for acoustic model adaptation of continuous density HMMs," in *Proc. ICASSP*, 2003, vol. I, pp. 556–559.

[123] K. Yu, M. J. F. Gales, and P. C. Woodland, "Unsupervised adaptation with discriminative mapping transforms," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 4, pp. 714–723, May 2009.

[124] L. Wang and P. C. Woodland, "MPE-based discriminative linear transform for speaker adaptation," in *Proc. ICASSP*, 2004, pp. 321–324.

[125] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[126] M. G. Rahim and B. H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 1, pp. 19–30, Jan. 1996.

[127] M. J. F. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR framework," *Comput., Speech, Lang.*, vol. 10, pp. 249–264, 1996.

[128] G. Saon, J. M. Huerta, and E. E. Jan, "Robust digit recognition in noisy environments: The IBM aurora 2 system," in *Proc. Interspeech*, 2001, pp. 629–632.

[129] G. Saon, G. Zweig, and M. Padmanabhan, "Linear feature space projections for speaker adaptation," in *Proc. ICASSP*, 2001, vol. 1, pp. 325–328.

[130] X. Cui and A. Alwan, "Noise robust speech recognition using feature compensation based on polynomial regression of utterance SNR," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 6, pp. 1161–1172, Nov. 2005.

[131] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, "High-performance HMM adaptation with joint compensation of additive and convolutive distortions via vector Taylor series," in *Proc. ASRU*, 2007, pp. 65–70.

[132] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, "A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions," *Comput., Speech, Lang.*, vol. 23, no. 3, pp. 389–405, 2009.

[133] M. J. F. Gales, "Model-based techniques for noise robust speech recognition," Ph.D. dissertation, Univ. of Cambridge, Cambridge, U.K., 1995.

[134] P. J. Moreno, "Speech recognition in noisy environments," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, USA, 1996.

[135] R. Stern, A. Acero, F. H. Liu, and Y. Ohshima, "Signal processing for robust speech recognition," in *Automatic Speech and Speaker Recognition: Advanced Topics*, C.-H. Lee, F. K. Soong, and K. K. Paliwal, Eds. Norwell, MA, USA: Kluwer, 1996, ch. 15, pp. 357–384.

[136] A. Acero and R. Stern, "Environmental robustness in automatic speech recognition," in *Proc. ICASSP*, 1990, vol. 2, pp. 849–852.

[137] F.-H. Liu, R. M. Stern, A. Acero, and P. Moreno, "Environment normalization for robust speech recognition using direct cepstral comparison," in *Proc. ICASSP*, 1994, vol. I, pp. 61–64.

[138] L. Deng, A. Acero, M. Plumpe, and X. Huang, "Large vocabulary speech recognition under adverse acoustic environment," in *Proc. ICSLP*, 2000, vol. 3, pp. 806–809.

[139] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. D. Huang, "High-performance robust speech recognition using stereo training data," in *Proc. ICASSP*, 2001, pp. 301–304.

[140] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. ICASSP*, 2005, vol. 1, pp. 961–964.

[141] L. Deng, J. Wu, J. Droppo, and A. Acero, "Analysis and comparison of two speech feature extraction/compensation algorithms," *IEEE Signal Process. Lett.*, vol. 12, no. 6, pp. 477–480, Nov. 2005.

[142] 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. ICASSP*, 1997, vol. 11, pp. 49–52.

[143] J. Droppo and A. Acero, "Maximum mutual information SPLICE transform for seen and unseen conditions," in *Proc. Interspeech*, 2005, pp. 989–992.

[144] J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE algorithm on the AURORA2 database," in *Proc. Eurospeech*, 2001, pp. 217–220.

[145] X. Cui, M. Afify, and Y. Gao, "MMSE-based stereo feature stochastic mapping for noise robust speech recognition," in *Proc. ICASSP*, 2008, pp. 4077–4080.

[146] M. Afify, X. Cui, and Y. Gao, "Stereo-based stochastic mapping for robust speech recognition," in *Proc. ICASSP*, 2007, vol. IV, pp. 377–380.

[147] M. Afify, X. Cui, and Y. Gao, "Stereo-based stochastic mapping for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 7, pp. 1325–1334, Sep. 2009.

[148] X. Cui, M. Afify, and Y. Gao, "N-best based stochastic mapping on stereo HMM for noise robust speech recognition," in *Proc. Interspeech*, 2008, pp. 1261–1264.

[149] X. Cui, M. Afify, and Y. Gao, "Stereo-based stochastic mapping with discriminative training for noise robust speech recognition," in *Proc. ICASSP*, 2009, pp. 3933–3936.

[150] J. Du, Y. Hu, L. R. Dai, and R. H. Wang, "HMM-based pseudo-clean speech synthesis for splice algorithm," in *Proc. ICASSP*, 2010, pp. 4570–4573.

[151] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, 2000, vol. 3, pp. 1315–1318.

[152] 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. ICASSP*, 2013, pp. 6822–6826.

[153] M. J. F. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 417–428, Jul. 2000.

[154] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 695–707, Nov. 2000.

[155] 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. ICSLP*, 2000, vol. 3, pp. 742–745.

[156] 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. ICASSP*, 2001, vol. 1, pp. 345–348.

[157] Y. Tsao and C. H. Lee, "An ensemble speaker and speaking environment modeling approach to robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 17, no. 5, pp. 1025–1037, Jul. 2009.

[158] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Statist. Soc. B.*, vol. 58, no. 1, pp. 267–288, 1996.

[159] X. Xiao, J. Li, E. S. Chng, and H. Li, "Lasso environment model combination for robust speech recognition," in *Proc. ICASSP*, 2012, pp. 4305–4308.

[160] X. Cui, J. Xue, and B. Zhou, "Improving online incremental speaker adaptation with eigen feature space MLLR," in *Proc. ASRU*, 2009, pp. 136–140.

[161] 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. ICASSP*, 2011, pp. 5048–5051.

[162] T. N. Sainath, B. Ramabhadran, M. Picheny, D. Nahamoo, and D. Kanevsky, "Exemplar-based sparse representation features: From TIMIT to LVCSR," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2598–2613, Nov. 2011.

[163] J. F. Gemmeke and T. Virtanen, "Noise robust exemplar-based connected digit recognition," in *Proc. ICASSP*, 2010, pp. 4546–4549.

[164] B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh, "Non-negative matrix factorization based compensation of music for automatic speech recognition," in *Proc. Interspeech*, 2010, pp. 717–720.

[165] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2067–2080, Sep. 2011.

[166] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NIPS*, 2000, pp. 556–562.

[167] M. N. Schmidt and R. K. Olsson, "Linear regression on sparse features for single-channel speech separation," in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust.*, 2007, pp. 26–29.

[168] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007.

[169] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2140–2151, Oct. 2013.

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

[171] 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. ICASSP*, 2012, pp. 4681–4684.

[172] J. F. Gemmeke, L. ten Bosch, L. Boves, and B. Cranen, "Using sparse representations for exemplar based continuous digit recognition," in *Proc. EUSIPCO*, 2009, pp. 1755–1759.

[173] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 1–12, Jan. 2007.

[174] K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "speech denoising using nonnegative matrix factorization with priors," in *Proc. ICASSP*, 2008, pp. 4029–4032.

[175] E. M. Grais and H. Erdogan, "Discriminative nonnegative dictionary learning using cross-coherence penalties for single channel source separation," in *Proc. Interspeech*, 2013, pp. 808–812.

[176] X. Cui and Y. Gong, "A study of variable-parameter gaussian mixture hidden markov modeling for noisy speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1366–1376, May 2007.
[177] D. Yu, L. Deng, Y. Gong, and A. Acero, "A novel framework and training algorithm for variable-parameter hidden Markov models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 7, pp. 1348–1360, Sep. 2009.
[178] 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*, 2008, pp. 1253–1256.
[179] D. Yu, L. Deng, Y. Gong, and A. Acero, "Discriminative training of variable-parameter HMMs for noise robust speech recognition," in *Proc. Interspeech*, 2008, pp. 285–288.
[180] N. Cheng, X. Liu, and L. Wang, "Generalized variable parameter HMMs for noise robust speech recognition," in *Proc. Interspeech*, 2011, pp. 481–484.
[181] Y. Li, X. Liu, and L. Wang, "Feature space generalized variable parameter HMMs for noise robust recognition," in *Proc. Interspeech*, 2013, pp. 2968–2972.
[182] Y. Minami and S. Furui, "A maximum likelihood procedure for a universal adaptation method based on HMM composition," in *Proc. ICASSP*, 1995, pp. 129–132.
[183] Y. Minami and S. Furui, "Adaptation method based on HMM composition and EM algorithm," in *Proc. ICASSP*, 1996, pp. 327–330.
[184] A. Sankar and C.-H. Lee, "Robust speech recognition based on stochastic matching," in *Proc. ICASSP*, 1995, pp. 121–124.
[185] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proc. ICSLP*, 2000, pp. 869–872.
[186] S. Sagayama, Y. Yamaguchi, S. Takahashi, and J. Takahashi, "Jacobian approach to fast acoustic model adaptation," in *Proc. ICASSP*, 1997, vol. 2, pp. 835–838.
[187] D. Y. Kim, C. K. Un, and N. S. Kim, "Speech recognition in noisy environments using first-order vector Taylor series," *Speech Commun.*, vol. 24, no. 1, pp. 39–49, 1998.
[188] Y. Gong, "A method of joint compensation of additive and convolutive distortions for speaker-independent speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 975–983, Sep. 2005.
[189] V. Stouten, "Robust automatic speech recognition in time-varying environments," Ph.D. dissertation, K. U. Leuven, Leuven, Belgium, 2006.
[190] R. A. Gopinath, M. Gales, P. S. Gopalakrishnan, S. Balakrishnan-Aiyer, and M. A. Picheny, "Robust speech recognition in noise - performance of the IBM continuous speech recognizer on the ARPA noise spoken task," in *Proc. DARPA Workshop Spoken Lang. Syst. Technol.*, 1995, pp. 127–130.
[191] Y. Zhao, "Frequency-domain maximum likelihood estimation for automatic speech recognition in additive and convolutive noises," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 3, pp. 255–266, May 2000.
[192] H. Liao and M. J. F. Gales, Joint uncertainty decoding for robust large vocabulary speech recognition Univ. of Cambridge, Cambridge, U.K., Tech. Rep. CUED/TR552, 2006.
[193] Y. Zhao and B. H. Juang, "Nonlinear compensation using the gauss-newton method for noise-robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 8, pp. 2191–2206, Nov. 2012.
[194] Y. Zhao and B. H. Juang, "A comparative study of noise estimation algorithms for VTS-based robust speech recognition," in *Proc. Interspeech*, 2010, pp. 2090–2093.
[195] V. Stouten, H. Van Hamme, K. Demuynck, and P. Wambacq, "Robust speech recognition using model-based feature enhancement," in *Proc. Eurospeech*, 2003, pp. 17–20.
[196] J. Droppo, L. Deng, and A. Acero, "A comparison of three non-linear observation models for noisy speech features," in *Proc. Eurospeech*, 2003, pp. 681–684.
[197] J. Li, L. Deng, D. Yu, and Y. Gong, "Towards high-accuracy low-cost noisy robust speech recognition exploiting structured mode," in *Proc. ICML Workshop Learn. Architectures, Representat., Optimization Speech Vis. Inf. Process.*, 2011.
[198] J. Li, M. L. Seltzer, and Y. Gong, "Improvements to VTS feature enhancement," in *Proc. ICASSP*, 2012, pp. 4677–4680.
[199] O. Kalinli, M. L. Seltzer, J. Droppo, and A. Acero, "Noise adaptive training for robust automatic speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 1889–1901, Nov. 2010.
[200] J. Li, M. L. Seltzer, and Y. Gong, "Efficient VTS adaptation using Jacobian approximation," in *Proc. Interspeech*, 2012, pp. 1906–1909.
[201] N. S. Kim, "Statistical linear approximation for environment compensation," *IEEE Signal Process. Lett.*, vol. 5, no. 1, pp. 8–10, Jan. 1998.
[202] V. Stouten, H. Van Hamme, and P. Wambacq, "Effect of phase-sensitive environment model and higher order VTS on noisy speech feature enhancement," in *Proc. ICASSP*, 2005, vol. I, pp. 433–436.
[203] J. Du and Q. Huo, "A feature compensation approach using high-order vector taylor series approximation of an explicit distortion model for noisy speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2285–2293, Nov. 2011.
[204] J. Du and Q. Huo, "A speech enhancement approach using piecewise linear approximation of an explicit model of environmental distortions," in *Proc. Interspeech*, 2008, pp. 569–572.
[205] K. Kalgaonkar, M. L. Seltzer, and A. Acero, "Noise robust model adaptation using linear spline interpolation," in *Proc. ASRU*, 2009, pp. 199–204.
[206] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, "HMM adaptation using a phase-sensitive acoustic distortion model for environment-robust speech recognition," in *Proc. ICASSP*, 2008, pp. 4069–4072.
[207] M. J. F. Gales and F. Flego, Discriminative classifiers and generative kernels for noise robust speech recognition Univ. of Cambridge, Cambridge, U.K., Tech. Rep. CUED/TR605, 2008.
[208] L. Lu, A. Ghoshal, and S. Renals, "Noise adaptive training for subspace gaussian mixture models," in *Proc. Interspeech*, 2013, pp. 3492–3496.
[209] B. Frey, L. Deng, A. Acero, and T. Kristjansson, "ALGONQUIN: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition," in *Proc. Interspeech*, 2001, pp. 901–904.
[210] R. C. van Dalen and M. J. F. Gales, "Extended VTS for noise-robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 733–743, May 2011.
[211] M. J. F. Gales and R. C. van Dalen, "Predictive linear transforms for noise robust speech recognition," in *Proc. ASRU*, 2007, pp. 59–64.
[212] H. Liao and M. J. F. Gales, "Issues with uncertainty decoding for noise robust automatic speech recognition," *Speech Commun.*, vol. 50, no. 4, pp. 265–277, 2008.
[213] H. Xu, M. J. F. Gales, and K. K. Chin, "Joint uncertainty decoding with predictive methods for noise robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 6, pp. 1665–1676, Nov. 2011.
[214] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, May 2004.
[215] V. Stouten, H. Van Hamme, and P. Wambacq, "Kalman and unscented Kalman filter feature enhancement for noise robust ASR," in *Proc. Interspeech*, 2005, pp. 953–956.
[216] Y. Hu and Q. Huo, "An HMM compensation approach using unscented transformation for noisy speech recognition," in *Proc. ISCSLP*, 2006.
[217] H. Xu and K. K. Chin, "Comparison of estimation techniques in joint uncertainty decoding for noise robust speech recognition," in *Proc. Interspeech*, 2009, pp. 2403–2406.
[218] F. Faubel, J. McDonough, and D. Klakow, "On expectation maximization based channel and noise estimation beyond the vector Taylor series expansion," in *Proc. ICASSP*, 2010, pp. 4294–4297.
[219] J. Li, D. Yu, Y. Gong, and L. Deng, "Unscented transform with online distortion estimation for HMM adaptation," in *Proc. Interspeech*, 2010, pp. 1660–1663.
[220] R. C. van Dalen and M. J. F. Gales, "A variational perspective on noise-robust speech recognition," in *Proc. ASRU*, 2011, pp. 125–130.
[221] V. Leutnant, A. Krueger, and R. Haeb-Umbach, "A versatile Gaussian splitting approach to non-linear state estimation and its application to noise-robust ASR," in *Proc. Interspeech*, 2011, pp. 1641–1644.
[222] M. J. F. Gales, "Acoustic factorisation," in *Proc. ASRU*, 2001, pp. 77–80.
[223] L. García, C. Bentíez, J. C. Segura, and S. Umesh, "Combining speaker and noise feature normalization techniques for automatic speech recognition," in *Proc. ICASSP*, 2011, pp. 5496–5499.
[224] M. L. Seltzer and A. Acero, "Separating speaker and environmental variability using factored transforms," in *Proc. Interspeech*, 2011, pp. 1097–1100.
[225] M. Rouvier, M. Bouallegue, D. Matrouf, and G. Linarès, "Factor analysis based session variability compensation for automatic speech recognition," in *Proc. ASRU*, 2011, pp. 141–145.
[226] M. Karafiát, L. Burget, P. Matejka, O. Glembek, and J. Cernocký, "iVector-based discriminative adaptation for automatic speech recognition," in *Proc. ASRU*, 2011, pp. 152–157.
[227] Y. Wang and M. J. F. Gales, "Speaker and noise factorisation for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 7, pp. 2149–2158, Sep. 2012.

[228] Y. Wang and M. J. F. Gales, "An explicit independence constraint for factorised adaptation in speech recognition," in *Proc. Interspeech*, 2013, pp. 1233–1237.

[229] H. Jiang and L. Deng, "A robust compensation strategy for extraneous acoustic variations in spontaneous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 1, pp. 9–17, Jan. 2002.

[230] N. Merhav and C. H. Lee, "A minimax classification approach with application to robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 1, no. 1, pp. 90–100, Jan. 1993.

[231] Q. Huo, H. Jiang, and C. H. Lee, "A Bayesian predictive classification approach to robust speech recognition," in *Proc. ICASSP*, 1997, pp. 1547–1550.

[232] Q. Huo and C. H. Lee, "A Bayesian predictive classification approach to robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 2, pp. 200–204, Nov. 2000.

[233] M. Afify, O. Siohan, and C. H. Lee, "Upper and lower bounds on the mean of noisy speech: Application to minimax classification," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 2, pp. 79–88, Feb. 2002.

[234] J. Du and Q. Huo, "IVN-based joint training of GMM and HMMs using an improved VTS-based feature compensation for noisy speech recognition," in *Proc. Interspeech*, 2012, pp. 1227–1230.

[235] J. A. Arrowood and M. A. Clements, "Using observation uncertainty in HMM decoding," in *Proc. Interspeech*, 2002, pp. 1561–1564.

[236] L. Deng, J. Droppo, and A. Acero, "Exploiting variances in robust feature extraction based on a parametric model of speech distortion," in *Proc. Interspeech*, 2002, pp. 2449–2452.

[237] L. Deng, J. Droppo, and A. Acero, "Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 412–421, May 2005.

[238] V. Stouten, H. Van Hamme, and P. Wambacq, "Model-based feature enhancement with uncertainty decoding for noise robust ASR," *Speech Commun.*, vol. 48, no. 11, pp. 1502–1514, 2006.

[239] J. Droppo, L. Deng, and A. Acero, "Uncertainty decoding with SPLICE for noise robust speech recognition," in *Proc. ICASSP*, 2002, vol. 1, pp. 57–60.

[240] H. Liao and M. J. F. Gales, "Joint uncertainty decoding for noise robust speech recognition," in *Proc. Interspeech*, 2005, pp. 3129–3132.

[241] H. Liao and M. J. F. Gales, "Adaptive training with joint uncertainty decoding for robust recognition of noisy data," in *Proc. ICASSP*, 2007, vol. 4, pp. 389–392.

[242] S. Srinivasan and D. Wang, "Transforming binary uncertainties for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 7, pp. 2130–2140, Sep. 2007.

[243] R. F. Astudillo and R. Orglmeister, "Computing MMSE estimates and residual uncertainty directly in the feature domain of asr using stft domain speech distortion models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 5, pp. 1023–1034, Jul. 2013.

[244] H. Liao and M. J. F. Gales, "Issues with uncertainty decoding for noise robust speech recognition," in *Proc. Interspeech*, 2006, pp. 1121–1124.

[245] M. J. F. Gales, The generation and use of regression class trees for MLLR adaptation Univ. of Cambridge, Cambridge, U.K., Tech. Rep. CUED/TR263, 1996.

[246] H. Xu, M. J. F. Gales, and K. K. Chin, "Improving joint uncertainty decoding performance by predictive methods for noise robust speech recognition," in *Proc. ASRU*, 2009, pp. 222–227.

[247] D. K. Kim and M. J. F. Gales, "Noisy constrained maximum-likelihood linear regression for noise-robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 2, pp. 315–325, Feb. 2011.

[248] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Karafiat, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas, "Subspace gaussian mixture models for speech recognition," in *Proc. ICASSP*, 2010, pp. 4330–4333.

[249] L. Lu, K. K. Chin, A. Ghoshal, and S. Renals, "Joint uncertainty decoding for noise robust subspace gaussian mixture models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 9, pp. 1791–1804, Sep. 2013.

[250] M. Cooke, P. D. Green, and M. Crawford, "Handling missing data in speech recognition," in *Proc. ICSLP*, 1994, pp. 1555–1558.

[251] R. Lippmann and B. Carlson, "Using missing feature theory to actively select features for robust speech recognition with interruptions, filtering and noise," in *Proc. Eurospeech*, 1997, pp. 37–40.

[252] M. Cooke, P. D. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech Commun.*, vol. 34, no. 3, pp. 267–285, 2001.

[253] J. P. Barker, M. P. Cooke, and D. P. W. Ellis, "Decoding speech in the presence of other sources," *Speech Commun.*, vol. 45, no. 1, pp. 5–25, 2005.

[254] B. Raj and R. M. Stern, "Missing-feature approaches in speech recognition," *IEEE Signal Process. Mag.*, vol. 22, no. 5, pp. 101–116, Sep. 2005.

[255] B. Raj, M. L. Seltzer, and R. M. Stern, "Reconstruction of missing features for robust speech recognition," *Speech Commun.*, vol. 43, no. 4, pp. 275–296, 2004.

[256] B. Raj and R. Singh, "Reconstructing spectral vectors with uncertain spectrographic masks for robust speech recognition," in *Proc. ASRU*, 2005, pp. 65–70.

[257] L. Josifovski, M. Cooke, P. D. Green, and A. Vizinho, "State based imputation of missing data for robust speech recognition and speech enhancement," in *Proc. Eurospeech*, 1999, pp. 2837–2840.

[258] M. Van Segbroeck and H. Van Hamme, "Advances in missing feature techniques for robust large-vocabulary continuous speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 1, pp. 123–137, Jan. 2011.

[259] W. Hartmann, A. Narayanan, E. Fosler-Lussier, and D. Wang, "A direct masking approach to robust ASR," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 1993–2005, Oct. 2013.

[260] W. Hartmann and E. Fosler-Lussier, "Investigations into the incorporation of the ideal binary mask in ASR," in *Proc. ICASSP*, 2011, pp. 4804–4807.

[261] A. Vizinho, P. D. Green, M. Cooke, and L. Josifovski, "Missing data theory, spectral subtraction and signal-to-noise estimation for robust ASR: An integrated study," in *Proc. Eurospeech*, 1999, pp. 2407–2410.

[262] M. El-Maliki and A. Drygajlo, "Missing features detection and handling for robust speaker verification," in *Proc. Eurospeech*, 1999, pp. 975–978.

[263] J. van Hout and A. Alwan, "A novel approach to soft-mask estimation and log-spectral enhancement for robust speech recognition," in *Proc. ICASSP*, 2012, pp. 4105–4108.

[264] P. Renevey and A. Drygajlo, "Missing feature theory and probabilistic estimation of clean speech components for robust speech recognition," in *Proc. Eurospeech*, 1999, pp. 2627–2630.

[265] M. L. Seltzer, B. Raj, and R. M. Stern, "A bayesian classifier for spectrographic mask estimation for missing feature speech recognition," *Speech Commun.*, vol. 43, no. 4, pp. 379–393, 2004.

[266] J. Barker, M. Cooke, and P. D. Green, "Robust ASR based on clean speech models: An evaluation of missing data techniques for connected digit recognition in noise," in *Proc. Interspeech*, 2001, pp. 213–217.

[267] K. J. Palomäki, G. J. Brown, and D. L. Wang, "A binaural processor for missing data speech recognition in the presence of noise and small-room reverberation," *Speech Commun.*, vol. 43, no. 4, pp. 361–378, 2004.

[268] A. Narayanan and D. L. Wang, "Ideal ratio mask estimation using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7092–7096.

[269] J. Barker, L. Josifovski, M. Cooke, and P. D. Green, "Soft decisions in missing data techniques for robust automatic speech recognition," in *Proc. Interspeech*, 2000, pp. 373–376.

[270] J. A. Gonzalez, A. M. Peinado, N. M. Ning Ma, A. M. Gomez, and J. Barker, "Mmse-based missing-feature reconstruction with temporal modeling for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 3, pp. 624–635, Mar. 2013.

[271] N. Ma and J. Barker, "Coupling identification and reconstruction of missing features for noise-robust automatic speech recognition," in *Proc. Interspeech*, 2012, pp. 2638–2641.

[272] W. Hartmann and E. Fosler-Lussier, "ASR-driven top-down binary mask estimation using spectral priors," in *Proc. ICASSP*, 2012, pp. 4685–4688.

[273] W. Hartmann and E. Fosler-Lussier, "Improved model selection for the ASR-driven binary mask," in *Proc. Interspeech*, 2012, pp. 1203–1206.

[274] A. Narayanan and D. L. Wang, "Coupling binary masking and robust ASR," in *Proc. ICASSP*, 2013, pp. 6817–6821.

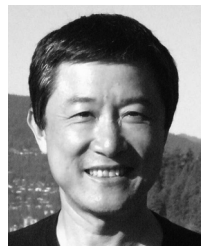
[275] Y. Gong, "Source normalization training for HMM applied to noisy telephone speech recognition," in *Proc. Eurospeech*, 1997, pp. 1555–1558.

[276] Y. Hu and Q. Huo, "Irrelevant variability normalization based HMM training using VTS approximation of an explicit model of environmental distortions," in *Proc. Interspeech*, 2007, pp. 1042–1045.

[277] O. Kalinli, M. L. Seltzer, and A. Acero, "Noise adaptive training using a vector Taylor series approach for noise robust automatic speech recognition," in *Proc. ICASSP*, 2009, pp. 3825–3828.

[278] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP*, 1996, vol. 2, pp. 1137–1140.
[279] F. Flego and M. J. F. Gales, "Discriminative adaptive training with VTS and JUD," in *Proc. ASRU*, 2009, pp. 170–175.
[280] Q. Huo and B. Ma, "Irrelevant variability normalization in learning HMM state tying from data based on phonetic decision-tree," in *Proc. ICASSP*, 1999, vol. 2, pp. 577–580.
[281] J. Wu and Q. Huo, "An environment compensated minimum classification error training approach and its evaluation on AURORA2 database," in *Proc. Interspeech*, 2002, pp. 453–456.
[282] J. Wu and Q. Huo, "An environment-compensated minimum classification error training approach based on stochastic vector mapping," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 6, pp. 2147–2155, Nov. 2006.
[283] Q. Huo and D. Zhu, "A maximum likelihood training approach to irrelevant variability compensation based on piecewise linear transformations," in *Proc. Interspeech*, 2006, pp. 1129–1132.
[284] D. Zhu and Q. Huo, "Irrelevant variability normalization based HMM training using MAP estimation of feature transforms for robust speech recognition," in *Proc. ICASSP*, 2008, pp. 4717–4720.
[285] N. Morgan and H. Hermansky, "RASTA extensions: Robustness to additive and convolutional noise," in *Proc. ESCA Workshop Speech Processing in Adverse Conditions*, 1992, pp. 115–118.
[286] T. Yoshioka and T. Nakatani, "Noise model transfer: Novel approach to robustness against nonstationary noise," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2182–2192, Oct. 2013.
[287] D. Yu, M. Seltzer, J. Li, J. T. Huang, and F. Seide, "Feature learning in deep neural networks–studies on speech recognition tasks," in *Proc. Int. Conf. Learn. Represent.*, 2013.
[288] B. Li and K. C. Sim, "Noise adaptive front-end normalization based on vector taylor series for deep neural networks in robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7408–7412.
[289] B. Li, Y. Tsao, and K. C. Sim, "An investigation of spectral restoration algorithms for deep neural networks based noise robust speech recognition," in *Proc. Interspeech*, 2013, pp. 3002–3006.
[290] M. Delcroix, Y. Kubo, T. Nakatani, and A. Nakamura, "Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling," in *Proc. Interspeech*, 2013, pp. 2992–2996.
[291] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proc. ICASSP*, 2012, pp. 4273–4276.
[292] T. N. Sainath, B. Kingsbury, A. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in *Proc. ASRU*, 2013, pp. 297–302.

Jinyu Li (M'08) received the Ph.D. degree from the Georgia Institute of Technology, Atlanta, in 2008 From 2000 to 2003, he was a Researcher in the Intel China Research Center and Research Manager in iFlytek Speech, China. Currently, he is a Principal Scientist Lead in Microsoft Corporation, Redmond, WA. He leads a team to design and improve speech modeling algorithms and technologies that ensure industry state-of-the-art speech recognition accuracy for Microsoft products. His major research interests cover several topics in speech recognition, including noise robustness, discriminative training, feature extraction, and machine learning methods.



Li Deng (SM'92–F'04) received the Ph.D. degree from the University of Wisconsin-Madison. He was an assistant professor (1989-1992), tenured associate professor (1992-1996), and tenured Full Professor (1996-1999) at the University of Waterloo, Canada. In 1999 he joined Microsoft Research, where he is currently a Principal Researcher. In the general areas of audio/speech/language technology, machine learning, and signal/information processing, he has published over 300 refereed papers in leading journals and conferences. He is a Fellow of the Acoustical Society of America, the IEEE, and the International Speech Communication Association. He served as Editor-in-Chief for the *IEEE Signal Processing Magazine* (2009-2011), which earned the highest impact factor in 2010 and 2011 among all IEEE publications and for which he received the 2012 IEEE SPS Meritorious Service Award. He recently served as General Chair of the IEEE ICASSP-2013, and currently serves as Editor-in-Chief for the IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING. His technical work since 2009 has focused on deep learning for speech recognition (with the 2013 IEEE SPS Best Paper Award), and on other areas of information processing and computer science including language and multimodal processing for a wide range of practical applications.

Yifan Gong (SM'93) received the Ph.D. (with highest honors) from the University of Henri Poincaré, France. He served the National Scientific Research Center (CNRS) and INRIA, France, as Research Engineer and then joined CNRS as Senior Research Scientist. He was a Visiting Research Fellow at the Communications Research Center of Canada. As Senior Member of Technical Staff, he worked for Texas Instruments at the Speech Technologies Lab, where he developed speech modeling technologies robust against noisy environments, designed systems, algorithms, and software for speech and speaker recognition, and delivered memory- and CPU-efficient recognizers for mobile devices. He joined Microsoft in 2004 and is currently a Principal Science Manager in the areas of speech modeling, computing infrastructure, and speech model development for speech products. His research interests include automatic speech recognition/interpretation, signal processing, algorithm development, and engineering process/infrastructure and management. He has authored over 130 publications and awarded over 30 patents. Specific contribution includes stochastic trajectory modeling, source normalization HMM training, joint compensation of additive and convolutional noises, and variable parameter HMM. In these areas, he gave tutorials and other invited presentations in international conferences. He has been serving as member of technical committee and session chair for many international conferences, and with IEEE Signal Processing Spoken Language Technical Committees from 1998 to 2002 and since 2013.

Reinhold Haeb-Umbach (M'89–SM'09) is a professor with the University of Paderborn, Germany. His main research interests are in the fields of statistical signal processing and pattern recognition, with applications to speech enhancement, acoustic beamforming and source separation, as well as automatic speech recognition. After having worked in industrial research laboratories for more than 10 years he joined academia as a full professor of Communications Engineering in 2001 He has published more than 150 papers in peer reviewed journals and conferences. He is the co-editor of the book *Robust Speech Recognition of Uncertain or Missing Data - Theory and Applications* (Springer, 2011).