CHAPTER

# Robust Speech Recognition Under Noisy Ambient Conditions

6

**Kuldip K. Paliwal**

*School of Microelectronic Engineering, Griffith University, Brisbane, Australia*

**Kaisheng Yao**

*Speech Component Group, Microsoft Corporation, Redmond, Washington*

## ABSTRACT

*Automatic speech recognition is critical in natural human-centric interfaces for ambient intelligence. The performance of an automatic speech recognition system, however, degrades drastically when there is a mismatch between training and testing conditions. The aim of robust speech recognition is to overcome the mismatch problem so the result is a moderate and graceful degradation in recognition performance. In this chapter, we provide a brief overview of an automatic speech recognition system, describe sources of speech variability that cause mismatch between training and testing, and discuss some of the current techniques to achieve robust speech recognition.*

*Key words*: automatic speech, recognition, robust speech recognition, speech enhancement, robust speech feature, stochastic matching, model combination, speaker adaptation, microphone array.

## 6.1 INTRODUCTION

Ambient intelligence is the vision of a technology that will become invisibly embedded in our surroundings, enabled by simple and effortless interactions, context sensitive, and adaptive to users [1]. Automatic speech recognition is a core component that allows high-quality information access for ambient intelligence. However, it is a difficult problem and one with a long history that began with initial papers appearing in the 1950s [2, 3]. Thanks to the significant progress made in recent years in this area [4, 5], speech recognition technology, once confined to research laboratories, is now applied to some real-world applications, and a number of commercial speech recognition products (from Nuance, IBM, Microsoft, Nokia, etc.) are on the market. For example, with automatic voice mail transcription by speech recognition, a user can have a quick view of her voice mail without having to listen to it. Other applications include voice dialing on embedded speech recognition systems.

The main factors that have made speech recognition possible are advances in digital signal processing (DSP) and stochastic modeling algorithms. Signal processing techniques are important for extracting reliable acoustic features from the speech signal, and stochastic modeling algorithms are useful for representing speech utterances in the form of efficient models, such as hidden Markov models (HMMs), which simplify the speech recognition task. Other factors responsible for the commercial success of speech recognition technology include the availability of fast processors (in the form of DSP chips) and high-density memories at relatively low cost.

With the current state of the art in speech recognition technology, it is relatively easy to accomplish complex speech recognition tasks reasonably well in controlled laboratory environments. For example, it is now possible to achieve less than a 0.4% string error rate in a speaker-independent digit recognition task [6]. Even continuous speech from many speakers and from a vocabulary of 5000 words can be recognized with a word error rate below 4% [7]. This high level of performance is achievable only when the training and the test data match. When there is a mismatch between training and test data, performance degrades drastically.

Mismatch between training and test sets may occur because of changes in acoustic environments (background, channel mismatch, etc.), speakers, task domains, speaking styles, and the like [8]. Each of these sources of mismatch can cause severe distortion in recognition performance for ambient intelligence. For example, a continuous speech recognition system with a 5000-word vocabulary raised its word error rate from 15% in clean conditions to 69% in 10-dB to 20-dB signal-to-noise ratio (SNR) conditions [9, 10]. Similar degradations in recognition performance due to channel mismatch are observed. The recognition accuracy of the SPHINX speech recognition system on a speaker-independent alphanumeric task dropped from 85% to 20% correct when the close-talking Sennheiser microphone used in training was replaced by the omnidirectional Crown desktop microphone [11]. Similarly, when a digital recognition system is trained for a particular speaker, its accuracy can be easily 100%, but its performance degrades to as low as 50% when it is tested on a new speaker.

To understand the effect of mismatch between training and test conditions, we show in Figure 6.1 the performance of a speaker-dependent, isolated-word recognition system on speech corrupted by additive white noise. The recognition system uses a nine-word English e-set alphabet vocabulary where each word is represented by a single-mixture continuous Gaussian density HMM with five states. The figure shows recognition accuracy as a function of the SNR of the test speech under (1) mismatched conditions where the recognition system is trained on clean speech and tested on noisy speech, and (2) matched conditions where the training and the test speech data have the same SNR.

It can be seen from Figure 6.1 that the additive noise causes a drastic degradation in recognition performance under the mismatched conditions; with the matched conditions, however, the degradation is moderate and graceful. It may be noted here that if the SNR becomes too low (such as $-10$ dB), the result is very poor recognition performance even when the system operates under matched noise conditions. This is because the signal is completely swamped by noise and no useful information can be extracted from it during training or in testing.

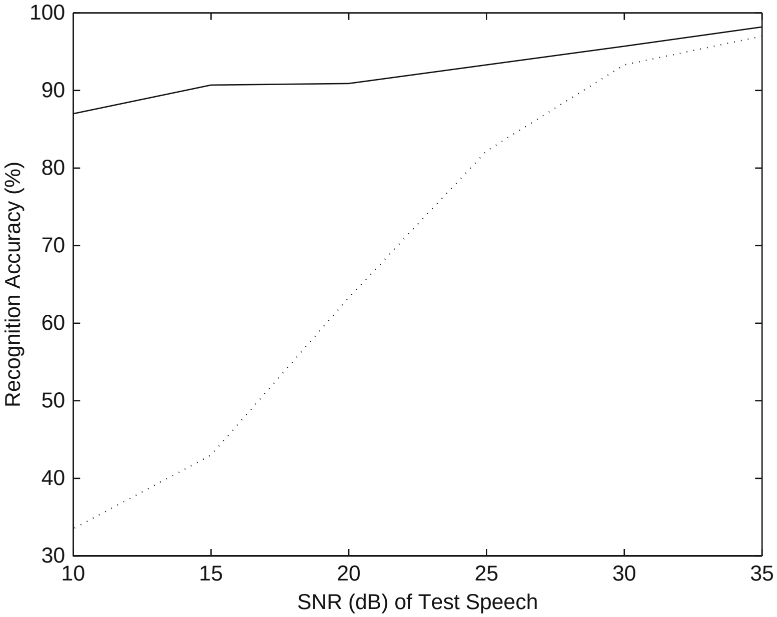

**FIGURE 6.1**

Effect of additive white noise on speech recognition performance under matched and mismatched conditions: training with clean speech (dotted line); training and testing with same-SNR speech (solid line).

When a speech recognition system is deployed in a real-life situation for ambient intelligence, there is bound to be a mismatch between training and testing that causes severe deterioration in recognition performance. The aim of a robust speech recognition system is to remove the effect of mismatch and achieve performance that is as graceful as obtained under matched conditions.

Note that devices used for ambient intelligence are usually small, low power, low weight, and (very important) low cost. A successful speech recognition system therefore needs to consider factors of practical implementation and system usage. These challenges include but are not limited to dealing with large volumes of incoming recognition requests, prompt response, and hardware constraints such as low memory and fixed-point arithmetic on DSP chips.

In this chapter, we provide only a glimpse of robust speech recognition and describe briefly some of the popular techniques used for this purpose. (For more details, see [12–21].) We will focus here mainly on techniques to handle mismatches resulting from changes in acoustic environments (e.g., channel and noise distortions). Some of these are equally applicable to mismatches resulting from speaker variability. The chapter is organized as follows: Section 6.2 provides a brief overview of the automatic speech recognition process. Different sources of variability in speech signals are discussed in Section 6.3. Robust speech recognition techniques are briefly described in Section 6.4. Section 6.5 concludes the chapter.

## 6.2 SPEECH RECOGNITION OVERVIEW

The objective of an automatic speech recognition system is to take the speech waveform of an unknown (input) utterance, and classify it as one of a set of spoken words, phrases, or sentences. Typically, this is done in two steps (as shown in Figure 6.2). In the first step, an acoustic front-end is used to perform feature analysis

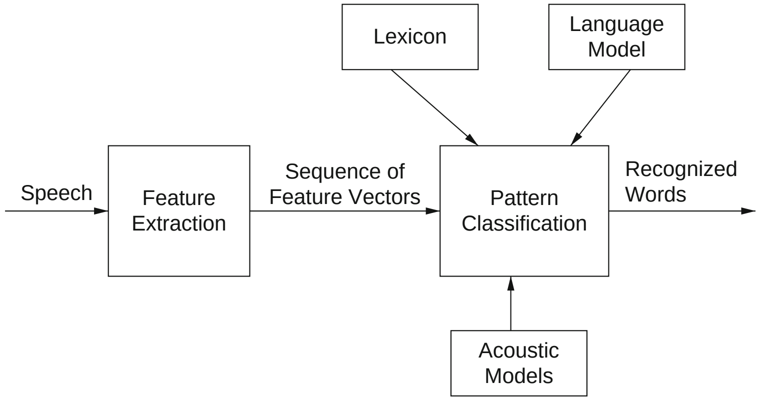

**FIGURE 6.2**

Block diagram of an automatic speech recognition system.

of the speech signal at the rate of about 100 frames per second to extract a set of features. This produces a sequence of feature vectors that characterizes the speech utterance sequentially in time.

The second step deals with pattern classification, where the sequence of feature vectors is compared against the machine's knowledge of speech (in the form of acoustics, lexicon, syntax, semantics, etc.) to arrive at a transcription of the input utterance.

Currently, most speech recognition systems use a statistical framework to carry out the pattern classification task, and they generally recognize the input speech utterance as a sequence of words. Consider a sequence of feature vectors,

$$\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$$

representing the $T$ frames of the input speech utterance. The task of the system is to find a word sequence,

$$W = \{w_1, w_2, \ldots, w_K\}$$

that maximizes the a posteriori probability of the observation sequence $\mathbf{Y}$; that is, the recognized word sequence,

$$\hat{W} = \{\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_{\hat{K}}\}$$

is given by the following equation:

$$\hat{W} = \underset{W}{\text{argmax}} \ \Pr(W|\mathbf{Y}) \tag{6.1}$$

In this Equation (6.1), maximization of the a posteriori probability $\Pr(W|\mathbf{Y})$ is over all possible word sequences $\{w_1, w_2, \ldots, w_K\}$ for all possible values of $K$. For a large-vocabulary continuous-speech system, this is a computationally exorbitant task. Fast search algorithms are available in the literature to carry it out [22–24].

Applying Bayes's rule and noting that $\Pr(\mathbf{Y})$ is independent of $W$, Equation (6.1) can be written as

$$\hat{W} = \underset{W}{\text{argmax}} \ \Pr(\mathbf{Y}|W) \cdot \Pr(W) \tag{6.2}$$

This is known as the maximum a posteriori probability (MAP) decision rule in the statistical pattern recognition literature [25].

Equation (6.2) indicates that we need two probabilities $\Pr(\mathbf{Y}|W)$ and $\Pr(W)$ to carry out the recognition task. These are computed through the acoustic and language models, respectively, which are briefly described as follows:

*Acoustic models*. The acoustic models are used to compute the probability $\Pr(\mathbf{Y}|W)$. To do this, we need the probability of an observed sequence of feature vectors for each of the words in the vocabulary. This is done by representing each word by a hidden Markov model (HMM) [27] and estimating the HMM parameters from an independent (and preferably large) speech data set during the training phase. To capture the sequential nature of speech, the left-to-right HMMs are used to model individual words. For a large-vocabulary continuous-speech recognition system, it is not possible to have one HMM for each word, so we seek smaller units (subword units) to characterize these probabilities. Examples of subword units are phonemes, demisyllables, and syllables. If there are $M$ phonemes in the (English) language, we can have $M$ HMMs, each estimated from the training data belonging to a particular phoneme. These are called context-independent models. For a large-vocabulary speech recognition system, such models are not adequate, and one requires context-dependent modeling to get good recognition performance.

Current recognition systems use HMMs for all possible left and right contexts for each phoneme (triphone models). Once the acoustic models (in the form of HMMs) are available for individual subword units (e.g., triphones) from the training phase, the word models are constructed from the subword models according to the transcription of the words (in terms of subword units) contained in the lexicon.

*Language model.* The language model is used to compute the probability Pr($W$). Note that Pr($W$) is independent of the observed feature vector sequence **Y**. Like acoustic models, the language model is estimated from a large, independent corpus of training data. Among different language models proposed in the literature, the $N$-gram model (where $N$ is typically 2, 3, or 4) is perhaps the most popular and simplest for representing the syntactic, semantic, and pragmatic sources of knowledge. In it, the probability of the current word depends on $N-1$ preceding words. Thus, it is very effective for capturing local dependencies between words. In an $N$-gram model, the probability $\Pr(w_k | w_1, w_2, \ldots, w_{k-1})$ is approximated by $\Pr(w_k | w_{k-1}, w_{k-2}, \ldots, w_{k-N+1})$. As an example, we show the procedure for calculating the probability Pr($W$) using the trigram model ($N = 3$).

$$\begin{aligned}\Pr(W) &= \Pr(w_1, w_2, \ldots, w_K)\\ &= \prod_{k=1}^{K} \Pr(w_k | w_1, w_2, \ldots, w_{k-1})\\ &= \prod_{k=1}^{K} \Pr(w_k | w_{k-1}, w_{k-2})\end{aligned} \tag{6.3}$$

Thus, we compute the acoustic and language models in the training phase from the data available for training the speech recognizer. Let us denote the set of acoustic models (i.e., subword HMMs) by $\Lambda_{\mathbf{X}}$ and the set of $N$-gram models by $\Upsilon_W$. Then the MAP decision rule (Equation (6.2)) can be written in terms of these models as follows:

$$\hat{W} = \underset{W}{\operatorname{argmax}} \Pr(\mathbf{Y} | W, \Lambda_{\mathbf{X}}) \cdot \Pr(W | \Upsilon_W) \tag{6.4}$$

During the test phase, the recognizer uses the acoustic and language models to compute the probabilities $\Pr(\mathbf{Y} | W, \Lambda_{\mathbf{X}})$ and $\Pr(W | \Upsilon_W)$ and to carry out the recognition of the input utterance according to the MAP decision rule given by Equation (6.4).

## 6.3 VARIABILITY IN THE SPEECH SIGNAL

Robust speech recognition deals with the mismatch between the training and testing conditions. Most of this mismatch is due to variability in the speech signal resulting from various sources, some of which are listed here:

*Background noise.* When speech is recorded in a given acoustic environment, the resulting signal is the sum of the speech produced by a speaker and the background (or ambient) noise. This additive noise generally has a colored spectrum, whose shape depends on the source that generates it. In an office environment, background noise results from sources such as computers, printers, typewriters, air conditioners, telephones, fans, background, conversations, and so forth. In a moving-car environment, it can be due to engine, wind, tires, road, and the like. Sources of background noise for other environments (such as a telephone booth, industrial plant, plane cockpit) can also be easily identified. Depending on the source, the background noise can be stationary (e.g., a fan or an air-conditioner) or nonstationary (e.g., a moving car).

Though spectral shape as well as level of background noise causes a degradation in recognition performance, the latter has a greater effect. When the background noise is at a relatively high level, it produces the Lombard effect [29], which changes even the characteristics of the speech signal produced by a given speaker.

*Room reverberation.* Room reverberation causes a convolutional distortion; that is, the reverberated speech signal can be modeled as a convolution of the intended speech signal, with the impulse response characterizing the distortion. The amount of reverberation distortion is determined by the room acoustics and the position of speaker and microphone within the room. When a speaker is at a relatively large distance from the microphone, the reverberation distortion becomes serious and can significantly affect speech recognition performance [30].

*Microphone characteristics.* A microphone acts on the speech signal as a linear filter (approximately) and causes convolution distortion. Since different types of microphones have different frequency responses, their mismatch during training and test conditions causes severe recognition performance degradation [11].

*Transmission channel.* When a speech recognizer is accessed through a telephone or mobile phone, the transmission channel used is totally unknown and unpredictable. This causes mismatch between training and testing, and speech recognition performance suffers because of it. A transmission channel acts like a linear filter on the speech signal and causes convolution distortion. Mobile telephony introduces another distortion, resulting from speech coders, which also affects recognition performance adversely [31].

*Intra-speaker variability.* When a person speaks the same word twice at different times of day, the resulting utterances show different acoustic characteristics. This intra-speaker variability is mainly caused by changes in the health and emotional state of the speaker.

*Inter-speaker variability.* Inter-speaker variability is one of the main sources of mismatch between training and testing conditions and is a major cause of the degraded performance of a speech recognizer. Differences in the length and shape of the vocal tract, dialect, pronunciation, and articulatory habits are some examples of this variability.

Most of the sources of speech variability discussed produce additive distortion (e.g., background noise) and/or convolution distortion (e.g., microphone mismatch) in the speech signal. A common model that describes these distortions and helps in understanding robust speech recognition techniques (discussed in the next section) is as follows:

$$\mathbf{y}_t = \mathbf{x}_t * \mathbf{h}_t + \mathbf{w}_t \tag{6.5}$$

where the symbol $*$ denotes the convolution operation and the subscript $t$ is the time index. $\mathbf{x}_t$ and $\mathbf{y}_t$ are the clean speech signal and the distorted signal, respectively. $\mathbf{h}_t$ and $\mathbf{w}_t$ each denote convolution distortion and the additive noise signal. In this equation, both distortions are nonstationary, but they can be assumed stationary for simplicity of analysis. Further assuming that $\mathbf{y}_t$, $\mathbf{x}_t$, and $\mathbf{w}_t$ are uncorrelated, we have a power spectrum after the short-time Fourier analysis of the distorted signal as

$$P_{yy}(m, f) = P_{xx}(m, f)|H(f)|^2 + P_{ww}(f) \tag{6.6}$$

where $f$ is the frequency variable and $m$ is the frame index; $H(f)$ denotes the Fourier transform of $\mathbf{h}_t$; and $P_{yy}(m, f)$, $P_{xx}(m, f)$, and $P_{ww}(f)$ are the power spectra of $\mathbf{y}_t$, $\mathbf{x}_t$, and $\mathbf{w}_t$, respectively. If there is only convolution distortion present in the signal, Equation (6.6) can be written as

$$\log P_{yy}(m, f) = \log P_{xx}(m, f) + 2\log |H(f)| \tag{6.7}$$

When the signal is corrupted by additive noise distortion (i.e., there is no convolution distortion), then Equation (6.6) can be written as

$$P_{yy}(m, f) = P_{xx}(m, f) + P_{ww}(f) \tag{6.8}$$

Equations (6.6), (6.7), and (6.8) form the basis of a number of the robust speech recognition techniques discussed in the next section.

## 6.4 ROBUST SPEECH RECOGNITION TECHNIQUES

As mentioned earlier, robust speech recognition deals with the problem resulting from the mismatch between training and testing. A speech recognizer is considered robust if it (approximately) maintains good recognition performance even if this mismatch exists.

Some researchers believe that one can solve the mismatch problem by increasing the size of the training data set and including all possible speech variations in it. However, this approach, called matched training or multi-condition training, solves the problem only to some extent. The models computed from this large training data set may be diffused and diluted (i.e., they may have large variance). As a result, their performance may be relatively poor for all test conditions. By increasing the size of the training data set and including all possible speech variations in it, one is only improving generalization capability at the cost of recognition performance.

To really solve the robust speech recognition problem, one has to understand the basic characteristics of the speech signal and the effect of different sources of distortion and variability, and then capture this knowledge during the feature extraction and acoustic-modeling stages. If the mismatch still remains, one must use small amounts of adaptation data prior to testing the recognition system for fine tuning.

In this chapter, we concentrate on the mismatch between training and testing conditions resulting from variability in the speech signal. This means that we have to handle the mismatch between the acoustic models $\Lambda_{\mathbf{X}}$ and the observation sequence **Y**, without worrying about the language model $\Upsilon_W$.

### 6.4.1 Speech Enhancement Techniques

The aim of a speech enhancement system is to suppress the noise in a noisy speech signal. For robust speech recognition, such a system is used as a preprocessor to a speech recognizer. Since it produces a clean speech signal, no changes in the recognition system are necessary to make it robust. A number of speech enhancement techniques have been reported in the literature [32]. They include spectral subtraction [33, 34, 41], Wiener and Kalman filtering [35], MMSE estimation [36], comb filtering [32], subspace methods [37, 38], and phase spectrum compensation [39, 40].

These techniques were originally developed with the aim of improving the intelligibility of noisy speech, but they can be used for robust speech recognition as well. The technique that has been used most for this purpose is spectral subtraction, in which the power spectrum of clean speech $P_{xx}(m, f)$ is estimated by explicitly subtracting the noise power spectrum $P_{ww}(f)$ from the noisy speech power spectrum $P_{yy}(m, f)$ using Equation (6.8). This requires information about the noise power spectrum, which can be estimated from the nonspeech frames detected by voice activity detection (VAD). However, it is not always possible to detect the nonspeech frames correctly, which affects the estimation of the noise power spectrum and may result in poor speech enhancement. Hence, a practical spectral subtraction scheme has the form

$$\hat{P}_{xx}(m, f) = \begin{cases} P_{yy}(m, f) - \alpha P_{ww}(f) & \textit{if } \hat{P}_{xx}(m, f) \geq \beta P_{ww}(f) \\ \beta P_{ww}(f) & \textit{otherwise} \end{cases} \tag{6.9}$$

where $\alpha$ is an oversubtraction factor and $\beta$ sets a spectral floor to avoid the enhanced spectra from becoming negative. Wiener filtering, which is closely related to spectral subtraction [44], was recently used in an industrial standard [45, 46] for speech recognition.

Since VAD itself is hard to tune for correct determination of speech and nonspeech events, especially in low SNR and highly nonstationary noise conditions, there are methods that estimate the noise power spectrum without VAD. In [42, 43], based on the observations that speech and background noise are usually statistically independent and that the power spectrum $P_{yy}(m, f)$ frequently decays to the noise power spectrum $P_{ww}(f)$, minimum statistics methods estimated the noise power spectrum by tracking and smoothing spectral minima in each frequency band.

Phase spectrum compensation (PSC) [39, 40] is a recently proposed approach to speech enhancement in which the noisy magnitude spectrum is recombined with a changed phase spectrum to produce a modified complex spectrum. During synthesis, the low-energy components of the modified complex spectrum cancel out more than the high-energy components, thus reducing background noise.

The PSC procedure is as follows. First, the noisy speech signal $\mathbf{y}_t$ is transformed via $N$-point short-time Fourier transform into the complex spectrum $Y(m, k)$ at frame $m$. Second, the noisy complex spectrum is offset by an additive real-valued frequency-dependent $\Xi(k)$ function:

$$Y_{\Xi}(m,k) = Y(m,k) + \Xi(k) \tag{6.10}$$

where $\Xi(k)$ should be anti-symmetric about $F_s/2$ (half the sampling rate). A simple anti-symmetric $\Xi(k)$ function may be as follows:

$$\Xi(k) = \begin{cases} +\xi & 0 \le k < \dfrac{N}{2} \\ -\xi & \dfrac{N}{2} \le k \le N-1 \end{cases} \tag{6.11}$$

where $\xi$ is a real-valued constant and $N$ is the length of frequency analysis assumed to be even. $Y_{\Xi}(m, k)$ is used to compute the changed phase spectrum through the arctangent function

$$\angle \mathrm{Y}_{\Xi}(m,k) = \arctan\left(\frac{Im\{\mathrm{Y}_{\Xi}(m,k)\}}{Re\{\mathrm{Y}_{\Xi}(m,k)\}}\right) \tag{6.12}$$

where $Im\{\cdot\}$ and $Re\{\cdot\}$ denote imaginary and real operators, respectively. The phase spectrum is combined with the noisy magnitude spectrum to produce a modified complex spectrum:

$$\hat{X}_{\Xi}(m,k) = |\mathrm{Y}(m,k)| \mathrm{e}^{j\angle Y_{\Xi}(m,\, k)} \tag{6.13}$$

In the synthesis stage, the complex spectrum of Equation (6.13) is converted to a time-domain representation. Because of the additive offset introduced in Equation (6.10), the modified complex spectrum $\hat{X}_{\Xi}(m,k)$ may not be conjugate symmetric and the resulting time-domain signal may be complex. In the proposed PSC method, the imaginary component is discarded. The enhanced signal $\hat{x}_t$ is produced by employing the overlap-and-add procedure.

All of the above methods can be tuned to achieve a certain degree of noise reduction at the cost of some speech distortion. Note that, for an ASR system, speech distortion is usually difficult to compensate for, but residual noise remaining after speech enhancement can be post-processed using techniques discussed later in this chapter. Hence, an optimal set of speech enhancement parameters for ASR systems usually allows more residual noise than that for a human listener. For this reason, the optimal speech enhancement parameters for ASR and for a human listener can be very different.

### 6.4.2 Robust Feature Selection and Extraction Methods

Selection of proper acoustic features is perhaps the most important task in the design of a robust speech recognition system, as it directly affects system performance. These features should be selected with the following criteria in mind:

- They should contain the maximum information necessary for speech recognition.
- They should be insensitive to speaker characteristics, manner of speaking, background noise, channel distortion, and so forth.
- We should be able to estimate them accurately and reliably.
- We should be able to estimate them through a computationally efficient procedure.
- They should have a physical meaning (preferably consistent with the human auditory perception process).

Obviously, it is very difficult to select a set of acoustic features that satisfies all of these requirements, and a great deal of research has gone into identifying them (see [54, 58–60, 62] and references therein for different front-ends).

Once the features are selected, the task of the acoustic front-end is to extract them from the speech signal. It does this by dividing the signal into overlapping time frames and computing the values of the features for each frame. The complexity of the acoustic front-end depends on the type of features selected. They may be as simple as the energy and zero-crossing rate of the waveform during each frame. A better, but more complex, method for feature analysis is based on the source/system model of the speech production system. It is generally considered that the system part of this model represents the vocal tract response and contains most of the linguistic information necessary for speech recognition. The power spectrum of each speech frame contains information about the source part (in the form of a fine structure) and vocal tract system part (in the form of a smooth spectral envelope). The acoustic front-end computes the smooth spectral envelope from the power spectrum by removing the fine structure. Once the envelope is estimated, it can be represented in terms of a few parameters (such as cepstral coefficients). These parameters are used as acoustic features in a speech recognition system. Human listeners, in contrast, can recognize speech even in the presence of large amounts of noise and channel distortions. Therefore, it is argued that the acoustic front-end can be made more robust to these distortions by utilizing the properties of the human auditory system.

The Mel filter-bank analysis procedure [55] is based on the fact that the frequency sensitivity of the human ear is greater at low frequencies than at high frequencies. Therefore, the analysis computes the power spectrum of a given speech frame by using a nonuniform filter bank, where filter bandwidth increases logarithmically with filter frequency (according to the Mel scale). The Mel frequency cepstral coefficients (MFCCs) representing the smooth spectral envelope are computed from the power spectrum using homomorphic analysis. The MFCC feature and its time derivatives [56] are now the most widely used speech features. Inclusion of time derivatives in the feature set improves recognition performance in matched as well as mismatched acoustic conditions [69].

PLP analysis [57] uses more detailed properties of the human auditory system than does Mel filter-bank analysis to compute the power spectrum. In addition to a nonuniform filter bank (where filters are spaced according to the Bark scale), it uses an equal loudness curve and the intensity-loudness power law to better model the auditory system. The cepstral features are estimated from the resulting power spectrum using LP analysis.

Recent discriminative feature extraction methods [64, 65, 67] use a posteriori probabilities of a set of models that are related to classification accuracy. For example, the tandem features [64] are computed with a multilayer perceptron (MLP) to first discriminatively transform multiple feature vectors (typically several frames of MFCC or PLP features). The outputs of the MLPs approximate the a posteriori probabilities of selected phones given the snapshot of these input frames. For the a posteriori probabilities to be used as input to HMM, they are transformed by principal component analysis (PCA). Training MLPs increases the a posteriori probabilities of the selected phones in comparison to the probabilities of competing phones. It is found that augmenting the feature with other features such MFCCs and PLPs is preferable when training and testing environments have mismatches [66].

The feature-space minimum phone error (fMPE) method [67] is another example of discriminative feature extraction. Using a discriminative training method [68], it estimates a matrix to transform a high-dimension vector consisting of a posteriori probabilities of Gaussian components in HMM into a time-dependent vector that adds to the original input vector. These Gaussian components are further updated on the extracted feature. The procedures for estimating the transform and for updating the Gaussian components are repeated several times. It is interesting that the fMPE feature is not appended to the original feature vector, as is common with other feature extraction methods such as the previously mentioned tandem feature or the delta/acceleration parameters. A discussion in [114] suggests that fMPE may be analyzed in light of feature compensation methods such as SPLICE [113] (see Section 6.4.4).

Heteroscedastic linear discriminant analysis (HLDA) [71] defines an objective function of discrimination using the actual covariance of Gaussian components in HMM as follows,

$$Q_{HLDA}(\mathbf{A}) = \sum_{n} \gamma_n \log \frac{|\mathbf{A}|\Sigma_b \mathbf{A}^T}{\mathbf{A}\Sigma^{(n)}\mathbf{A}^T} \tag{6.14}$$

where $\gamma_n$ is the total occupation posterior probability of component $n$, and $\Sigma^{(n)}$ is its covariance; $\Sigma_b$ is the between-class covariance. The matrix $\mathbf{A}$ is not constrained to be full rank, so it can have fewer rows than columns, which results in dimension reduction.

By maximizing the objective function, HLDA allows decorrelation of the elements in feature vectors while rotating them to achieve dimension reduction with little loss of discrimination. In practice, five to nine adjacent frames of feature vectors are concatenated, forming a slice with about 200 feature dimensions. The slice is then analyzed using HLDA, and the feature dimensions can be reduced to, say, 39. Since much of the discrimination of the original high-dimensional slice is kept, HLDA-processed features can yield improved performance over those with similar dimension but without such processing.

### 6.4.3 Feature Normalization Techniques

Variability not related to discriminating speech content may be reduced by feature normalization techniques. For instance, cepstral mean normalization (CMN) [72] assumes that the interfering distortion is stationary and convolutional (see Equation (6.7)), and suppresses it by subtracting the long-term cepstral mean vector (over the input utterance) from the current cepstral vector. This technique is currently very popular for overcoming channel mismatch distortion. When the channel is slowly varying with time, its effect can be eliminated by highpass filtering (e.g., RASTA) of the sequence of cepstral feature vectors [73, 76].

Cepstral variance normalization (CVN) [77] reduces mismatches by normalizing the second moment of the distribution of speech to a fixed value. It is usually combined with CMN and other techniques, such as speaker adaptation, and yields good results. Some embedded speech recognition systems [78] use CMN and CVN because of their low implementation costs. CMN and CVN are further extended in [79] to equalize histograms of training and testing data.

Vocal tract length normalization (VTLN) warps the frequency axis of the magnitude spectrum to reduce the shifting of formant frequencies due to speaker variations. This is achieved by linearly scaling the filter-bank center frequencies within the front-end feature extraction to approximate a canonical formant frequency scaling [80]. Given a previously trained acoustic model, a grid search on likelihoods against different frequency-warping parameters is performed to find the optimal likelihood of an utterance. Once all utterances are normalized, the acoustic models are re-estimated. This is repeated until the VTLN parameters of all utterances are stabilized. It was observed in [81] that VTLN may be considered a special type of maximum-likelihood linear regression method (see Section 6.4.5, subsection Adaptation-Based Compensation).

### 6.4.4 Stereo Data-Based Feature Enhancement

In some applications, stereo data may be available. For instance, hands-free microphones can be placed in a car to collect noisy speech along with speech collected via close-talking microphones. It is also possible to generate noisy speech by artificially adding noise to clean speech. Stereo data can be used to estimate mapping from noisy to clean speech, and this mapping can be used to enhance noisy speech features in the testing phase.

Usually, a minimum mean-square error (MMSE) scheme [82, 83, 86, 113] is used to enhance the feature vector:

$$\hat{X}_t = \boldsymbol{\varepsilon}\left(\mathbf{X}_t | \mathbf{Y}_t\right) \tag{6.15}$$

The noisy observation $\mathbf{y}_t$ is assumed to be distributed in multiple Gaussian components $\sum_n c_n N(\mathbf{y}_t; \mu_y^{(n)}, \Sigma_y^{(n)})$ with a mixture weight $c_n$ for Gaussian component $n$. This is similar for clean speech distribution. Assuming that the clean speech $\mathbf{x}_t$ and the noisy speech are distributed jointly within each component $n$, an estimate of the clean speech in the component is

$$\begin{aligned} \boldsymbol{\varepsilon}\Big(\mathbf{x}_t|\mathbf{y}_t, n) &= \mu_x^{(n)} + \Sigma_{xy}^{(n)}(\Sigma_y^{(n)})^{-1}(\mathbf{y}_t - \mu_y^{(n)}) \\ &= \mathbf{A}^{(n)}\mathbf{y}_t + \mathbf{b}^{(n)} \end{aligned} \tag{6.16}$$

Because there are correlations among clean and noisy speech, as shown in $\Sigma_{xy}^{(n)}$, matrix $\mathbf{A}^{(n)}$ is usually full, representing a rotation from $\mathbf{y}_t$ to the cleaned speech $\mathbf{x}_t$. $\mathbf{b}^{(n)}$ serves as a bias within the component. The MMSE estimate (Eq. 6.15) is then a weighted average of the above estimates, with weights being a posteriori probabilities of Gaussian components given a noisy observation sequence.

The SPLICE method [113] assumes only bias $\mathbf{b}^{(n)}$ to be estimated within each Gaussian component $n$. Using stereo data, a training procedure estimates the bias as

$$\mathbf{b}^{(n)} = \frac{\sum_t \gamma_n(t)(\mathbf{x}_t - \mathbf{y}_t)}{\sum_t \gamma_n(t)} \tag{6.17}$$

where

$$\gamma_n(t) = \frac{c_n N(\mathbf{y}_t; \mu_y^{(n)}, \Sigma_y^{(n)})}{\sum_n c_n N(\mathbf{y}_t; \mu_y^{(n)}, \Sigma_y^{(n)})} \tag{6.18}$$

is the a posteriori probability of Gaussian component $n$ given noisy observation $\mathbf{y}_t$. During testing, at each time $t$, a Gaussian component $n^*$ is selected as $n^* = \text{argmax}_n\, \gamma_n(t)$ and the enhanced feature is obtained by subtracting the bias of the selected Gaussian component from noisy observation as $\hat{x}_t = \mathbf{y}_t - \mathbf{b}^{n^*}$.

When stereo data is not available, statistics in a noisy environment may be predicted using a model of the environment. This type of method is described in Section 6.4.5.

### 6.4.5 The Stochastic Matching Framework

This section describes methods in a stochastic matching framework [94] that can further improve the performance of features extracted as discussed in Sections 6.4.1 through 6.4.4. To achieve this, knowledge of acoustic models is usually employed. In a typical stochastic matching scenario, HMMs are computed during the training phase from a large collection of data coming from a number of speakers and environments. Stochastic matching is carried out for a new speaker or environment either in feature-space or in model space using a small amount of speaker-specific or environment-specific adaptation data.

Let us denote this adaptation data by $\mathbf{Y}$, and let us assume that the transcription of $\mathbf{Y}$ is available. Let us denote this transcription by $W$. This data is utilized to design a transformation $\mathbf{G}_\theta$ in the model space:

$$\mathbf{\Lambda_Y} = \mathbf{G}_\theta(\mathbf{\Lambda_X}) \tag{6.19}$$

or a transformation $\boldsymbol{F}_\theta$ in the feature-space:

$$\hat{\mathbf{Y}} = \mathbf{F}_\theta(\mathbf{Y}) \tag{6.20}$$

The functional forms of the transformations are assumed to be known from our prior knowledge of the source of mismatch, and $\theta$ are the associated parameters. These parameters are estimated so as to provide the best match between the transformed models $\Lambda_\mathbf{Y}$ and the adaptation data $\mathbf{Y}$, or the enhanced feature $\hat{\mathbf{Y}}$ and the original model $\Lambda_\mathbf{X}$. Usually, the maximum likelihood formulation is used to estimate $\hat{\theta}$ for model space adaptation as follows:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \Pr(\mathbf{Y}|\theta, W, \mathbf{\Lambda_X}) \cdot \Pr(W|\Upsilon_W) \tag{6.21}$$

The maximization in this equation can be carried out using the expectation-maximization (EM) algorithm [94]. Parameters for feature-space enhancement can be similarly derived.

We may categorize methods in stochastic matching according to the following criteria

- Whether a parametric function representing environment distortions is used
- Whether a method works in model space or in feature-space

### Model-Based Model Adaptation

One way to achieve robust speech recognition is by training a new set of models from scratch every time the test condition changes. These models may be saved on a large disk and be invoked whenever a working condition is detected [88, 89]. This approach requires a large disk space for saving the models, which is very expensive for applications such as embedded speech recognition.

Another approach [11, 84, 87, 90, 106, 107, 111] is to capture some information about the mismatch during the training phase and use it for model adaptation. The mismatch may be modeled as a parametric transformation of HMM parameters, and the parameters may be meaningful. Using Equations (6.7) and (6.8), the joint additive and convolutive distortion compensation method (JAC) [90] relates a noisy observation in the log-spectral domain to clean speech log-spectra $\log P_{xx}(m,f)$, additive noise $\log P_{ww}(f)$, and channel distortion $H(f)$ as

$$\begin{aligned}
\log P_{yy}(m,f) &= \log(P_{xx}(m,f)|H(f)|^2 + P_{ww}(f)) \\
&= \log P_{xx}(m,f) + \log|H(f)|^2 + \log\left(1 + \frac{P_{ww}(f)}{P_{xx}(m,f)|H(f)|^2}\right) \\
&= \log P_{xx}(m,f) + \log|H(f)|^2 + g(P_{xx}(m,f), |H(f)|^2, P_{ww}(f))
\end{aligned} \tag{6.22}$$

where

$$g(x,y,z) = \log(1 + \exp(\log z - \log x - \log y)) \tag{6.23}$$

To use the functions just described to represent effects in the model space, at every Gaussian component $n$, we expand them using a first-order Taylor series around clean speech mean $\mu_x^{(n)}$, noise mean $\mu_n$, and channel distortion mean $\mu_b$ as

$$\begin{aligned}\log P_{yy}(m,f) \approx\ & C^{-1}(\mu_x^{(n)} + \mu_b) + g(C^{-1}\mu_x^{(n)}, C^{-1}\mu_b, C^{-1}\mu_n) \\ & + \mathbf{G}_x(C\log P_{xx}(m,f) - \mu_x^{(n)}) + \mathbf{G}_b(C\log|H(f)|^2 - \mu_b) \\ & + \mathbf{G}_n(C\log P_{ww}(f) - \mu_n)\end{aligned} \tag{6.24}$$

where $C$ and $C^{-1}$ each denote the discrete cosine transformation and its inverse. $\mathbf{G}_x$, $\mathbf{G}_b$, and $\mathbf{G}_n$ each denote the first-order differential of noisy observation log $P_{yy}(m,f)$ with respect to clean speech, channel distortion, and noise.

The channel distortion and noise mean may be estimated using the EM algorithm. Given these estimations, JAC reduces the mismatch caused by channel distortion and noise by transforming clean speech mean $\mu_x^{(n)}$ to noisy speech mean $\hat{\mu}_y^{(n)}$ as

$$\hat{\mu}_y^{(n)} = \mu_x^{(n)} + \mu_b + C\,g(C^{-1}\mu_x^{(n)}, C^{-1}\mu_b, C^{-1}\mu_n) \tag{6.25}$$

Whereas JAC compensates mean $\mu_x^{(n)}$ only, the parallel model combination (PMC) [87] method and VTS methods [84] further compensate distortion effects on speech covariance. In addition to the distortions parameters $\mu_b$ and $\mu_n$, the original clean speech model mean $\mu_x^{(n)}$ can be updated [115].

These methods provide a framework for incorporating independent concurrent signals (speech and noise) using a parametric function with a small number of meaningful parameters. The nonlinear function (Eq. 6.25) allows the acoustic model mean to be quickly adapted to noisy conditions, thus allowing robust performance in slowly time-varying noise [106–108]. This is also very useful for embedded speech recognition [85] in mobile environments, where distortion parameters $\mu_b$ and $\mu_n$ must be estimated from the utterance in use and noise conditions for one utterance can be very different from those for others.

The computational cost of model-space methods is usually high because they apply the nonlinear function (Eq. 6.25) on every speech mean. In [112], a scheme was devised to reduce the number of times that (Eq. 6.25) is computed, so that the JAC method can still be applied on embedded devices with very limited computational resources. In this method, clean speech mean vectors are first vector-quantized to have a centroid mean vector $\mu_x^{c(n)}$ for every cluster $c(n)$. Then, for each cluster, the following bias is computed:

$$\mathbf{b}^{c(n)} = \mu_b + Cg(C^{-1}\mu_x^{c(n)}, C^{-1}\mu_b, C^{-1}\mu_n) \tag{6.26}$$

which represents the last two terms in function (Eq. 6.25). Finally, this bias is applied to every mean vector within the cluster to compensate environment distortion effects.

### Model-Based Feature Enhancement

Similar to the model-based model adaptation method, model-based feature enhancement methods [82, 91–93, 109, 110] use a parametric function of distortion and an optimal estimation of the function's parameters. In VTS-based feature enhancement method [109, 110], a clean-speech GMM, together with the estimate of distortion parameters, is used to produce a GMM for noisy speech. Statistics of noisy speech can be predicted by VTS decomposition of (Eq. 6.22) around the clean speech mean vector, noise, and channel distortion. For example, for a Gaussian component $n$, the noisy speech mean vector $\mu_y^{(n)}$ can be predicted using equation (Eq. 6.25). The variance of noisy speech, $\Sigma_y^{(n)}$, can be similarly derived. With the noisy-speech GMM, the a posteriori probability of component $n$ given a noisy observation can be estimated in Equation (6.18). We can then obtain an estimate of clean speech $\mathbf{x}_t$ using (Eq. 6.15) and

$$\varepsilon(\mathbf{x}_t|\mathbf{y}_t, n) = \mathbf{y}_t - \mu_h - Cg(C^{-1}\mu_x^{(n)}, C^{-1}\mu_h, C^{-1}\mu_n) \tag{6.27}$$

In the feature-space methods, clean-speech GMMs for enhancement have far fewer Gaussian components than do those in acoustic models for recognition. Therefore, the cost for computing the nonlinear function $g(\cdot)$ on every Gaussian component in the feature-space methods is much lower than that in model-space model adaptation methods (Section 6.4.5). However, since the latter can adapt every parameter in HMMs for speech recognition, they compensate environment effects in much more detail than does enhancement of feature vectors alone, and so, usually perform better than model-based feature enhancement methods.

### Adaptation-Based Compensation

The methods in this section, although originally developed for speaker adaptation, are equally useful for handling other sources of mismatch (background noise, microphone, and channel mismatch distortion). They follow the same concept in (Eq. 6.19) for model-space and (Eq. 6.20) feature-space stochastic matching, but do not necessarily use a parametric function with parameters representing environment distortions. The several functional forms include a simple cepstral bias [94–96], linear affine transformation [97–99], and nonlinear transformation realized through an MLP [100]. The linear affine transformation is currently the most popular choice, and the resulting formulation (given by Eq. (6.21)) is called maximum likelihood linear regression (MLLR) [98, 99]. Particularly, feature-space MLLR, or fMLLR, is a variant of MLLR that has the same linear transforms applied on means and covariance. In fMLLR, the adapted mean $\hat{\mu}^{(n)}$ and covariance $\hat{\Sigma}^{(n)}$ at Gaussian component $n$ are

$$\hat{\mu}^{(n)} = \mathbf{A}\mu^{(n)} + \mathbf{b}$$
$$\hat{\Sigma}^{(n)} = \mathbf{A}\Sigma^{(n)}\mathbf{A}^T$$

where $\mathbf{A}$ and $\mathbf{b}$ are, respectively, the transformation matrix and the bias. The likelihood of observation $\mathbf{y}_t$ at Gaussian component $(n)$ can be expressed as

$$\Pr(\mathbf{y}_t|\theta, \mathbf{\Lambda}_X) = |\mathbf{A}|^{-1}N(\mathbf{A}^{-1}(\mathbf{y}_t - \mathbf{b}); \mu^{(n)}, \Sigma^{(n)}) \tag{6.28}$$

Because this likelihood computation does not change the model parameters, fMLLR is appealing in applications where modifying HMM parameters is expensive.

To improve performance of linear affine transformations, it is a common and powerful practice to use a regression tree. When the amount of adaptation data is limited, a global transform can be tied to all Gaussian components. With sufficient data, a transform can be specific to a leaf node of the regression tree that represents a group of Gaussian components sharing some common characteristics. With sufficient data, then, these components are adapted in the sense of piecewise linear transformations, which can to some extent compensate any nonlinear distortions.

Instead of using transformation-based adaptation, one can adapt the HMMs directly using the MAP algorithm [101, 102]. MAP incorporates prior knowledge of HMM parameters to get the MAP estimate for the new speaker using speaker-specific or environment-specific adaptation data. For a particular Gaussian mean, with prior mean $\mu_0^{(n)}$, the estimate is

$$\mu^{(n)} = \frac{\tau\mu_0^{(n)} + \sum_{t=1}^{T}\gamma_n(t)\mathbf{y}_t}{\tau + \sum_{t=1}^{T}\gamma_n(t)} \tag{6.29}$$

where $\tau$ controls the balance between the maximum likelihood estimate of the mean from the data and the prior mean, and $\gamma_n(t)$ is the posterior probability of Gaussian component $n$ at time $t$ given the observation sequence.

Though the MAP algorithm provides an optimal solution, it converges slowly and requires a relatively large amount of adaptation data. For better and faster adaptation, MAP can be combined with transformation-based methods [105, 120–123]. Another way to improve MAP's convergence speed is through structural MAP (SMAP) [124]. In SMAP, Gaussian components are organized into a tree structure and a mean offset at each layer is estimated for the Gaussians in it. At the root of the tree, the offset is an average of the maximum likelihood estimate of its mean shift and its a priori mean shift. This offset is then propagated to its children. The mean offset at a child is an average of the maximum likelihood estimate of its mean shift and this propagated prior mean shift from its parent. The computation is carried on recursively from the root of the tree to the leaves.

There has been some recent interest in fast adaptation techniques (such as cluster adaptive training (CAT) [103, 125], speaker-adaptive training (SAT) [26], eigenvoice techniques [126], and eigen-MLLR [104, 127]). Unlike MLLR and MAP, these methods use information about the characteristics of an HMM set for particular speakers or environments. They may be seen as an extension of speaker clustering. Rather than taking a hard decision about speaker style, which may lead to adaptation fragmentation and a poor choice of speaker group, these methods form a weighted sum of "speaker cluster" HMMs, and use this interpolated model to represent the current speaker. The few parameters of these interpolation weights can be viewed as representing a new speaker in a "speaker space" spanned by these "speaker clusters."

The number of parameters in these methods can be controlled to be far smaller than the number of parameters to be estimated in other methods. For example, for a full-matrix MLLR transform, the number of parameters is $D \times (D + 1)$, and $D$ is the feature vector size. The simplest MLLR with diagonal transformation and a bias vector has $2 \times D$ parameters to be estimated. For a feature vector with 39 dimensions, the full-matrix MLLR transform has 1560 parameters and the simplest MLLR has 78. However, it is reported in [103] that, with only 8 speaker clusters, a CAT method can outperform MLLR.

It is important in these fast adaptation methods to have proper "speaker space." Otherwise, they can easily saturate their performance. Differences among these methods mainly lie in how the "speaker space" is formed. CAT uses individual speaker cluster models, which have a common variance and mixture weights and only their Gaussian mean values vary. Thus the mean for a particular Gaussian for a particular speaker is

$$\hat{\mu} = \sum_k \lambda_k \mu_k \tag{6.30}$$

where $\lambda_k$ is a specific weight of mean vector $\mu_k$ in cluster $k$.

In fact, not only can CAT be fast because of the small number of $\lambda_k$ to be estimated during recognition, but its performance can be as good as that of an MLLR-adapted system. This is achieved using a scheme that updates these cluster means during training. That is, in CAT both weight $\lambda_k$ and speaker cluster $\mu_k$ are iteratively and jointly updated during training. During testing, only weights need to be estimated.

A variant of CAT represents cluster mean $\mu_k$ using a set of MLLR transforms of a "canonical model" as follows:

$$\mu_k = \sum_l c_l \mathbf{A}_{kl} \mu_0 \tag{6.31}$$

where $\mu_0$ is the canonical mean. $\mathbf{A}_{kl}$ are the MLLR transformations, and $c_l$ is the weight of transformation $\mathbf{A}_{kl}$. The number of parameters to be estimated during training is the number of parameters in the MLLR transforms plus that in the "canonical model," which could be lower than that in the CAT scheme in (Eq. 6.30). This variant of CAT, when combined with speaker adaptive training (SAT) [26], has been widely applied.

The eigenvoice technique [126] differs from CAT in that it finds cluster speakers using principle component analysis (PCA) of sets of "supervectors" constructed from all of the mean values in speaker-dependent HMMs [126]. Weights $\lambda_k$ are estimated as they are in CAT, however, there is no update of speaker clusters. A variant of the eigenvoice technique represents cluster speakers using PCA of sets of MLLR transforms that are applied on a speaker-independent and environment-independent HMM [104, 127].

### Uncertainty in Feature Enhancement

The previously mentioned feature-space methods, albeit low in cost, make point estimates of cleaned speech as if the estimates were exactly the original clean speech **X**. Since the clean speech may be difficult to recover, especially in conditions such as negative instantaneous SNR, models for recognition need to be adapted to be "aware" of the conditions' inaccuracy. One widely used and simple approach is retraining of models from enhanced features. For example, in [113], the SPLICE feature enhancement method is combined with matched condition training to obtain a lower word-error rate than achieved with feature enhancement alone.

Uncertainty decoding is a recent approach that uses an estimated degree of uncertainty, which may mask some undesirable distortion effects. In [118, 119], the MAP decision rule (Eq. 6.4) is modified to incorporate uncertainty due to noise $\mathbf{N}$ as follows:

$$\begin{aligned}\hat{W} &= \underset{\mathrm{W}}{\operatorname{argmax}}\ \Pr(W|\Upsilon_W)\Pr(\mathbf{Y}|W,\mathbf{\Lambda}_{\mathbf{X}})\\ &= \underset{\mathrm{W}}{\operatorname{argmax}}\ \Pr(W|\Upsilon_W)\int_{\mathbf{X}}\Pr(\mathbf{Y}|\mathbf{X},\theta)\Pr(\mathbf{X}|W,\mathbf{\Lambda}_{\mathbf{X}})d\mathbf{X}\end{aligned} \tag{6.32}$$

where the likelihood of noisy speech, given clean speech, is evaluated as follows:

$$\Pr(\mathbf{Y}|\mathbf{X},\theta) = \int_{\mathbf{N}}\Pr(\mathbf{Y}|\mathbf{X},\mathbf{N})\Pr(\mathbf{N}|\theta)d\mathbf{N} \tag{6.33}$$

A natural choice for approximating (6.33) is to use $M$ Gaussian components. At time $t$, we then have

$$\Pr(\mathbf{y}_t|\mathbf{x}_t,\theta) = \sum_{m=1}^{M}\Pr(m|\theta)\Pr(\mathbf{y}_t|\mathbf{x}_t,m,\theta) \tag{6.34}$$

We may train the Gaussian components based on noisy speech, and then

$$\Pr(m|\theta) \approx \Pr(m|\mathbf{y}_t,\theta) \tag{6.35}$$

Further approximation can be done by choosing the most likely component $m^*$. Then we have

$$\Pr(\mathbf{y}_t|\mathbf{x}_t,\theta) \approx \Pr(\mathbf{y}_t|\mathbf{x}_t,m^*,\theta) \tag{6.36}$$

This conditional likelihood is plugged into (Eq. 6.32). At time $t$, (Eq. 6.32) is a convolution of Gaussian components. Assuming $\mathbf{y}_t$ is linearly transformed from $\mathbf{x}_t$, we have

$$\Pr(\mathbf{y}_t|\theta,\mathbf{\Lambda}_{\mathbf{x}}) = \sum_n c_n \mathrm{N}(\mathbf{A}^{m^*}\mathbf{y}_t + \mathbf{b}^{m^*};\mu_x^{(n)},\Sigma_x^{(n)} + \Sigma_b^{m^*}) \tag{6.37}$$

Compared to the feature enhancement in the fMLLR described in Section 6.4.5, the additional cost is a result of adding global variance $\Sigma_b^{m^*}$ to the original variance $\Sigma_x^{(n)}$ of each component. If this global variance is diagonal, the additional cost is insignificant.

In addition to incorporating uncertainty, another advantage of this decision rule is that one may choose a simplified model to evaluate Pr($\mathbf{Y}$|$\mathbf{X}$, θ). In the simplest scheme [116], a single Gaussian is used to model noisy speech, so that a global variance is used in (Eq. 6.37) to enlarge the original Gaussian variance of the clean acoustic model. With more Gaussian components used, (Eq. 6.37) becomes a cluster-dependent computation that has Gaussian component variances enlarged with a cluster-dependent variance, and uses cluster-dependent speech enhancement. A simpler scheme in [117] uses cluster-dependent enhancement of speech but applies a global variance to enlarge the original Gaussian variances.

### 6.4.6 Special Transducer Arrangement to Solve the Cocktail Party Problem

It has been observed that a speech recognizer can be made robust to adverse acoustic environments if the transducers can be favorably arranged. In the simplest case, if we use a unidirectional microphone and place it near the mouth, we can reduce distortion due to background noise and reverberation. This will improve the SNR of the recorded speech. If we use two microphones, one to capture the noise signal and the other to pick up the noisy speech signal, we can apply adaptive filtering algorithms, such as least mean squares (LMS), to achieve speech enhancement. This cancels both stationary and nonstationary noise and improves recognition performance in the presence of noise [47, 48].

What is more challenging to state-of-the-art speech recognition systems is recognizing speech in the presence of competing speech-like distortions. This is known as the cocktail party problem [20], and it is very apparent in hands-free speech recognition applications (e.g., teleconferencing, a moving car), where it is not possible to use a close-talking microphone. The common wisdom is to apply a set of well-spaced microphone arrays and to properly fuse the signals they recorded. With an adaptive beamforming procedure used jointly with a source localization procedure [49–52], the SNR of speech is increased in both stationary and nonstationary acoustic environments (background noise and reverberation). In addition to beamforming, there is independent component analysis (ICA) [53], a recent approach for enhancement of desired speech experiencing types of other interference. ICA assumes mutual statistical independence between the desired speech and interference. Both beamforming and ICA may introduce some distortions to the enhanced speech signal. To deal with the problem, we can use methods introduced in previous sections (e.g., CMN) to remove some residual distortion in the enhanced signal.

## 6.5 SUMMARY

In this chapter, we addressed robust speech recognition for ambient intelligence. Mismatch between training and testing conditions causes severe degradation in speech recognition performance. The aim of robust speech recognition is to overcome this mismatch so that degradation in performance becomes moderate and graceful. We concentrated here on mismatch resulting from variability in the speech signal. The sources for this variability include additive background noise, channel and microphone mismatches, speaker mismatch, and different accents, stress types, and speaking styles.

A number of widely used robust speech recognition techniques were briefly described. These range from simple to complex. To build a robust system for ambient intelligence, it is a good idea to start with simple approaches, including feature normalization methods such as CMN, which have advantages of easy implementation and low computational cost. To further boost robustness, we may use a proper adaptation method such as MLLR.

However, in some situations involving ambient intelligence, these approaches may not be sufficient. For instance, MLLR requires larger amounts of adaptation data than do methods that use a parametric function of environment effects on clean speech. The parametric function allows model space adaptation or feature-space enhancement with small amounts of data. Thus, if high performance is required with small amounts of adaptation data, some model-based methods can be used, but they may not be applied to any feature types or to compensate any distortion types. For instance, CVN cannot work with the parametric function (6.24) in JAC.

In the design of a practical robust speech recognition system for ambient intelligence, computational complexity is a very important factor. Thus, it is worthwhile to revise robust speech recognition methods in order to achieve simplified procedures, albeit with some performance losses. Balancing performance and computational cost for robust speech recognition for ambient intelligence will be a design art.

## REFERENCES

[1] Aghajan H, Augusto J, Delgado R. Human-Centric Interfaces for Ambient Intelligence. Elsevier; 2009.

[2] Davis KH, Biddulph R, Balashek S. Automatic recognition of spoken digits. J Acoust Soc Am 1952;24:637.

[3] Dudley H, Balashek S. Automatic recognition of phonetic patterns in speech. J Acoust Soc Am 1958;30:721–32.

[4] Lee CH, Soong FK, Paliwal KK, editors. Automatic Speech Recognition: Advanced Topics. Kluwer Academic Publishers; 1996.

[5] Young S. A review of large-vocabulary continuous-speech recognition. IEEE Signal Processing Magazine 1996;13:45–57.

[6] Li J, Lee CH. Soft margin feature extraction for automatic speech recognition. Proc INTERSPEECH 2007;30–3.

[7] Macherey W, Haferkamp L, Schluter R, Ney H. Investigations on error minimizing training criteria for discriminative training in automatic speech recognition. Proc INTERSPEECH 2005;2133–6.

[8] Junqua JC. Impact of the unknown communication channel on automatic speech recognition: A review. Proc EUROSPEECH 1997;KN29–32.

[9] Parihar N, Picone J, Pearce D, Hirsch H. Performance analysis of the AURORA large vocabulary baseline system. EUSIPCO 2004;553–6.

[10] Yeung S-K, Siu M-H. Improved performance of AURORA 4 using HTK and unsupervised MLLR adaptation. Proc INTERSPEECH 2004;161–4.

[11] Acero A, Stern RM. Environmental robustness in automatic speech recognition. Proc ICASSP 1990;849–952.

[12] Junqua JC, Haton JP, editors. Robustness in Automatic Speech Recognition. Kluwer Academic Publishers; 1996.

[13] Juang BH. Speech recognition in adverse environments. Computer Speech and Language 1991;5:275–94.

[14] Gong Y. Speech recognition in noisy environments: A survey. Computer Speech and Language 1995;16:261–91.

[15] Furui S. Recent advances in robust speech recognition. In: Proc. ESCANATO Workshop on Robust Speech Recognition for Unknown Communication Channels. 1997. p. 11–20.

[16] Bellegarda JR. Statistical techniques for robust ASR: Review and perspectives. Proc EUROSPEECH 1997;KN33–6.

[17] Lee CH. Adaptive compensation for robust speech recognition. In: Proc. IEEE Workshop on Automatic Speech Recognition and Understanding. 1997. p. 357–64.

[18] Lee CH. On stochastic feature and model compensation approaches to robust speech recognition. Speech Communication 1998;25:29–47.

[19] Woodland PC. Speaker adaptation for continuous density HMMs: A Review. In: ITRW on Adaptation Methods for Speech Recognition. 2001. p. 11–9.

[20] Huang XD, Acero A, Hon H-W. Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall PTR; 2001.

[21] Hirsch H-G, Pearce D. The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy environments. Proc ASR 2000;181–8.

[22] Ney H, Aubert X. Dynamic programming search strategies: From digit strings to large vocabulary word graphs. In: Lee CH, Soong FK, Paliwal KK, editors. Automatic Speech Recognition: Advanced Topics. Kluwer Academic Publishers; 1996. p. 384–411.

[23] Gopalakrishnan PS, Bahl LR. Fast search techniques. In: Lee CH, Soong FK, Paliwal KK, editors. Automatic Speech Recognition: Advanced Topics. Kluwer Academic Publishers; 1996. p. 413–28.

[24] Schwartz R, Nguyen L, Makhoul J. Multiple-pass search strategies. In: Lee CH, Soong FK, Paliwal KK, editors. Automatic Speech Recognition: Advanced Topics. Kluwer Academic Publishers; 1996. p. 429–56.

[25] Duda RO, Hart PE. Pattern Classification and Scene Analysis. Wiley; 1973.

[26] Anastasakos T, McDonough J, Schwartz R, Makhoul J. A compact model for speaker-adaptive training. Proc ICSLP 1996;2:1137–40.

[27] Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 1989;77:257–86.

[28] Rabiner L, Juang BH. Fundamentals of Speech Recognition. Prentice Hall; 1993.

[29] Junqua JC, Anglade Y. Acoustic and perceptual studies of Lombard speech: Application to isolated-words automatic speech recognition. Proc ICASSP 1990;841–4.

[30] Nakamura S, Takiguchi T, Shikano K. Noise and room acoustics distorted speech recognition by HMM composition. Proc ICASSP 1996;69–72.

[31] Lilly BT, Paliwal KK. Effect of speech coders on speech recognition performance. Proc ICSLP 1996;2344–7.

[32] Lim JS, Oppenheim AV. Enhancement and bandwidth compression of noisy speech. Proc IEEE 1979;67:1586–604.

[33] Boll SF. Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans Acoust, Speech Signal Proc ASSP-27 1979;113–20.

[34] Berouti M, Schwartz R, Makhoul J. Enhancement of speech corrupted by additive noise. Proc ICASSP 1979;208–11.

[35] Paliwal KK, Basu A. A speech enhancement based on Kalman filtering. Proc ICASSP 1987;177–80.

[36] Ephraim Y, Malah D. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. IEEE Trans Acoust Speech Signal Proc 1985;33(2):443–5.

[37] Dendrinos M, Bakamidis S, Carayannis G. Speech enhancement from noise: A regenerative approach. Speech Commun 1991;10(2):45–57.

[38] Ephraim Y, Trees HV. A signal subspace approach for speech enhancement. IEEE Trans. Speech Audio Proc 1995;3(4):251–66.

[39] Wójcicki K, Milacic M, Stark A, Lyons J, Paliwal KK. Exploiting conjugate symmetry of the short-time Fourier spectrum for speech enhancement. IEEE Sign Proc Lett 2008;15:461–4.

[40] Stark A, Wójcicki K, Lyons J, Paliwal KK. Noise driven short-time phase spectrum compensation procedure for speech enhancement. Proc INTERSPEECH 2008;549–53.

[41] Van Compernolle D. Noise adaptation in a hidden Markov model speech recognition system. Comput Speech Lang 1989;3:151–67.

[42] Martin R. Spectral subtraction based on minimum statistics. In: Proc. European Signal Processing Conference. 1994. p. 1182–5.

[43] Martin R. Noise power spectral density estimation based on optimal smoothing and minimum statistics. IEEE Trans Speech Audio Proc 2001;9(5):504–12.

[44] Virag N. Signal channel speech enhancement based on masking properties of the human auditory system. IEEE Trans Speech Audio Proc 1999;7(2):126–37.

[45] Agarwal A, Cheng YM. Two-stage Mel-warped Wiener filter for robust speech recognition. Proc ASRU 1999;67–70.

[46] ETSI. Standard speech processing, transmission and quality tests (STQ); distributed speech recognition; front-end feature extraction algorithm; compression algorithms. ETSI ES 201 108, v1.1.2. 2004.

[47] Powell G, Darlington P, Wheeler P. Practical adaptive noise reduction in the aircraft cockpit environment. Proc ICASSP 1987;ETSI ES 201 108, v1.1.2. 2004.

[48] Nakadai Y, Sugamura N. A speech recognition method for noise environments using dual inputs. Proc ICSLP 1990;1141–4.

[49] Flanagan JL, Johnston JD, Zahn R, Elko GW. Computer-steered microphone arrays for sound transduction in large rooms. J Acoust Soc Amer 1985;78:1508–18.

[50] Silverman HF, Kirtman SE. A two-stage algorithm for determining talker location from linear microphone data. Comput Speech Lang 1992;6:129–52.

[51] Omologo M, Svaizer P. Acoustic source location in noisy and reverberant environment using CSP analysis. Proc ICASSP 1996;921–4.

[52] Lleida E, Fernandez J, Masgrau E. Robust continuous speech recognition system based on a microphone array. Proc ICASSP 1998;241–4.

[53] Bell A, Sejnowski T. An information-maximization approach to blind separation and blind deconvolution. Neural Comput 1995;7:1129–59.

[54] Picone JW. Signal modeling techniques in speech recognition. Proc IEEE 1993;81(9):1215–47.

[55] Davis SB, Mermelstein P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans Acoust, Speech Signal Proc 1980;28(4):357–60.

[56] Furui S. Speaker-independent isolated word recognition using dynamic features of speech spectrum. IEEE Trans Acoust Speech Signal Proc 1986;34:52–9.

[57] Hermansky H. Perceptual linear predictive (PLP) analysis of speech. J Acoust Soc Am 1990;87(4):1738–52.

[58] Paliwal KK, Sagisaka Y. Cyclic autocorrelation-based linear prediction analysis of speech. Proc EUROSPEECH 1997;279–82.

[59] Johnston JD, Brandenburg K. Wideband coding—Perceptual considerations for speech and music. In: Furui S, Sondhi MM, editors. Advances in Speech Signal Processing. Marcel Dekker; 1992. p. 109–49.

[60] Lilly BT, Paliwal KK. Auditory masking based acoustic front-end for robust speech recognition. In: Proc. IEEE Region 10 Conf. on Speech and Image Technologies for Computing and Communications. 1997. p. 165–8.

[61] Varga A, Moore R. Hidden Markov model decomposition of speech and noise. Proc ICASSP 1990;845–8.

[62] Paliwal KK. Spectral subband centroids as features for speech recognition. In: Proc. IEEE Workshop on Automatic Speech Recognition and Understanding. 1997. p. 124–31.

[63] Paliwal KK. Decorrelated and filtered filter-bank energies for robust speech recognition. Proc EUROSPEECH 1999;85–8.

[64] Hermansky H, Ellis D, Sharma S. Tandem connectionist feature stream extraction for conventional HMM systems. Proc ICASSP 2000;1635–8.

[65] Zhu Q, Stolcke A, Chen B, Morgan N. Using MLP features in SRI's conversational speech recognition system. In: Proc. 9th European Conference on Speech Communication and Technology. 2005. p. 2141–4.

[66] Benitez C, Burget L, Chen B, Dupont S, Garudadri H, Hermansky H, et al. Robust ASR front-end using spectral-based and discriminant features: Experiments on the Aurora tasks. Proc EUROSPEECH 2001;429–32.

[67] Povey D, Kingsbury B, Mangu L, Saon G, Soltau H, Zweig G. fMPE: Discriminatively trained features for speech recognition. Proc ICASSP 2005;961–4.

[68] Povey D, Woodland PC. Minimum phone error and I-smoothing for improved discriminative training. Proc ICASSP 2002;105–8.

[69] Hanson BA, Appelbaum TH, Junqua JC. Spectral dynamics for speech recognition under adverse conditions. In: Lee CH, Soong FK, Paliwal KK, editors. Automatic Speech Recognition: Advanced Topics. Kluwer Academic Publishers; 1996. p. 331–56.

[70] Theodoridis S, Koutroumbas K. Pattern recognition. Academic Press Elsevier.

[71] Kumar N, Andreou AG. Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. Speech Communication 1994;26:283–97.

[72] Atal BS. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. J Acoust Soc Amer 1974;55:1304–12.

[73] Geller D, Umbach RH, Ney H. Improvements in speech recognition for voice dialing in car environment. In: Proc. ECSA Workshop on Speech Processing in Adverse Conditions. 1992. p. 203–6.

[74] Murveit H, Butzberger J, Weintraub M. Reduced channel dependence for speech recognition. In: Proc. Speech and Natural Language Workshop (DARPA). 1992. p. 280–4.

[75] Hermansky H, Morgan N. RASTA processing of speech. IEEE Trans Speech Audio Proc 1994;2(4):578–89.

[76] Aikawa K, Singer H, Kawahara H, Tohkura Y. A dynamic cepstrum incorporating time-frequency masking and its application to continuous speech recognition. Proc ICASSP 1993;2:668–71.

[77] Haeb-Umbach R, Aubert X, Beyerlein P, Klakow D, Ullrich M, Wendemuth A, et al. Acoustic modeling in the Philips Hub-4 continuous speech recognition system. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop.

[78] Iso-Sipila J, Moberg M, Viikki O. Multi-lingual speaker-independent voice user interface for mobile devices. Proc ICASSP 2006;1081–4.

[79] Hilger F, Ney H. Quantile based histogram equalization for noise robust large vocabulary speech recognition. IEEE Trans Speech Audio Proc 2006;14(3):845–54.

[80] Lee L, Rose R. Speaker normalization using efficient frequency warping procedures. Proc ICASSP 1996.

[81] Pitz M, Ney H. Vocal tract normalization as linear transformation of MFCC. Proc EUROSPEECH 2003;1445–8.

[82] Stern RM, Acero A, Liu FH, Ohshima Y. Signal processing for robust speech recognition. In: Lee CH, Soong FK, Paliwal KK, editors. Automatic Speech Recognition: Advanced Topics. Kluwer Academic Publishers; 1996. p. 357–84.

[83] Cui X, Afify M, Gao Y. MMSE-based stereo feature stochastic mapping for noise robust speech recognition. Proc ICASSP 2008;4077–80.

[84] Moreno P, Raj B, Stern RM. A vector Taylor series approach for environment independent speech recognition. Proc ICASSP 1996;733–6.

[85] Yao K, Netsch L, Viswanathan V. Speaker-independent name recognition using improved compensation and acoustic modeling methods for mobile applications. Proc ICASSP 2006;173–6.

[86] Neumeyer L, Weintraub M. Probabilistic optimum filtering for robust speech recognition. Proc ICASSP 1994;1:417–29.

[87] Gales MJF, Young SJ. An improved approach to the hidden Markov model decomposition of speech and noise. Proc ICASSP 1992;233–6.

[88] Akbacak M, Hansen J. Environmental sniffing: Robust digit recognition for an in-vehicle environment. Proc EUROSPEECH 2003;2177–80.

[89] Zhang Z, Furui S. Piecewise-linear transformation-based HMM adaptation for noisy speech. Proc ASRU 2001;159–62.

[90] Gong Y. A method of joint compensation of additive and convolutive distortions for speaker-independent speech recognition. IEEE Trans Speech Audio Proc 2005;13:975–83.

[91] Varga AP, Moore RK. Simultaneous recognition of concurrent speech signals using hidden Markov model decomposition. Proc EUROSPEECH 1991;1175–8.

[92] Rose R, Hofstetter E, Reynolds D. Integrated models of speech and background with application to speaker identification in noise. IEEE Trans Speech Audio Proc 1994;2:245–57.

[93] Ephraim Y. Gain adapted hidden Markov models for recognition of clean and noisy speech. IEEE Trans Signal Proc 1992;40:1303–16.

[94] Sankar A, Lee CH. A maximum likelihood approach to stochastic matching for robust speech recognition. IEEE Trans. Speech Audio Proc 1996;4:190–202.

[95] Rahim MG, Juang BH. Signal bias removal by maximum likelihood estimation for robust telephone speech recognition. IEEE Trans. Speech Audio Proc 1996;4:19–30.

[96] Zhao Y. An acoustic-phonetic based speaker adaptation technique improving speaker independent continuous speech recognition. IEEE Trans Speech Audio Proc 1994;2:380–94.

[97] Digalakis VV, Rtischev D, Neumeyer LG. Speaker adaptation using constrained estimation of Gaussian mixtures. IEEE Trans Speech Audio Proc 1995;3:357–66.

[98] Leggetter CJ, Woodland PC. Flexible speaker adaptation for large vocabulary speech recognition. Comput Speech Lang 1995;9:171–86.

[99] Gales MJF, Woodland PC. Mean and variance adaptation within the MLLR framework. Comput Speech Lang 1996;10:249–64.

[100] Abrash V, Sankar A, Franco H, Cohen M. Acoustic adaptation using nonlinear transformations of HMM parameters. Proc ICASSP 1996;729–32.

[101] Lee CH, Lin CH, Juang BH. A study on speaker adaptation of the parameters of continuous density hidden Markov models. IEEE Trans Signal Proc 1991;39:806–14.

[102] Gauvain JL, Lee CH. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Trans Speech Audio Proc 1994;2:291–8.

[103] Gales MJ. Cluster adaptive training of hidden Markov models. IEEE Trans Speech Audio Proc 2000;8(4):417–28.

[104] Mak B, Hsiao R. Kernel eigenspace-based MLLR adaptation. IEEE Trans Audio, Speech Lang Proc 2007;15(3):784–95.

[105] Digalakis VV, Neumeyer LG. Speaker adaptation using combined transformation and Bayesian methods. IEEE Trans Speech Audio Proc 1996;4:294–300.

[106] Kim NS. Non-stationary environment compensation based on sequential estimation. IEEE Signal Proc Lett 1998;3:57–9.

[107] Yao K, Paliwal KK, Nakamura S. Noise adaptive speech recognition based on sequential parameter estimation. Speech Commun 2004;42:5–23.

[108] Yao K, Nakamura S. Sequential noise compensation by sequential Monte Carlo method. In: Dietterich TG, Becker S, Ghahramani Z, editors. Advances in Neural Information Processing Systems 14. MIT Press; 2001. p. 1205–12.

[109] Kim DY, Un C, Kim NS. Speech recognition in noisy environments using first-order vector Taylor series. Speech Commun 1998;24:39–49.

[110] Acero A, Deng L, Kristjansson T, Zhang J. HMM adaptation using vector Taylor series for noisy speech recognition. Proc ICSLP 2000;869–72.

[111] Sagayama S, Yamaguchi Y, Takahashi S, Takahashi J. Jacobian approach to fast acoustic model adaptation. Proc ICASSP 1997;835–8.

[112] Yao K. Systems and methods employing stochastic bias compensation and Bayesian joint additive/convolutive compensation in automatic speech recognition. U.S. Patent 20070033027, 2007.

[113] Deng L, Acero A, Plumpe M, Huang XD. Large vocabulary speech recognition under adverse acoustic environments. Proc ICSLP 2000;806–9.

[114] Deng L, Wu J, Droppo J, Acero A. Analysis and comparison of two speech feature extraction/compensation algorithms. IEEE Signal Proc Lett 2005;2(6):477–80.

[115] Hu Y, Huo Q. Irrelevant variability normalization based HMM training using VTS approximation of an explicit model of environmental distortions. Proc INTERSPEECH 2007;1042–2045.

[116] Arrowwood J, Clements M. Using observation uncertainty in HMM decoding. Proc ICSLP 2002;3:1561–4.

[117] Deng L, Droppo J, Acero A. Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion. IEEE Trans. Speech Audio Proc 2005;13:412–21.

[118] Gales MJ, van Dalen RC. Predictive linear transforms for noise robust speech recognition. Proc ASRU 2007;59–64.

[119] Liao H, Gales MJ. Issues with uncertainty decoding for noise robust automatic speech recognition. Speech Commun 2008;50:265–77.

[120] Zavaliagkos G, Schwartz R, McDonough J. Maximum a posteriori adaptation for large scale HMM recognizers. Proc ICASSP 1996;725–8.

[121] Nagesh V, Gillick L. Studies in transformation-based adaptation. Proc ICASSP 1997;1031–4.

[122] Ishii J, Tonomura M. Speaker normalization and adaptation based on linear transformation. Proc ICASSP 1997;1055–8.

[123] Siohan O, Chesta C, Lee CH. Joint maximum a posteriori adaptation of transformation and HMM parameters. IEEE Trans Speech Audio Proc 2001;9(4):417–28.

[124] Shinoda K, Lee CH. A structural Bayes approach to speaker adaptation. IEEE Trans Speech Audio Proc 2001;9:276–87.

[125] Gales MJF. Cluster adaptive training for speech recognition. Proc ICSLP 1998;1783–6.

[126] Kuhn R, Nguyen P, Junqua JC, Goldwasser L, Niedzielski N, Fincke S, et al. Eigenvoices for speaker adaptation. Proc ICSLP 1998;1771–4.

[127] Chen KT, Liau WW, Wang HM, Lee LS. Fast speaker adaptation using eigenspace-based maximum likelihood linear regression. Proc ICSLP 2000;742–5.