A Maximum-Likelihood Approach to Stochastic Matching for Robust Speech Recognition [1]

Ananth Sankar and Chin-Hui Lee

Speech Research Department
AT&T Bell Laboratories
Murray Hill, NJ 07974

We present a maximum-likelihood (ML) stochastic matching approach to decrease the acoustic mismatch between a test utterance and a given set of speech models so as to reduce the recognition performance degradation caused by distortions in the test utterance and/or the model set. We assume that the speech signal is modeled by a set of subword hidden Markov models (HMM) Λ_X. The mismatch between the observed test utterance $\boldsymbol{Y}$ and the models Λ_X can be reduced in two ways: 1) by an inverse distortion function $F_\nu(.)$ that maps $\boldsymbol{Y}$ into an utterance $\boldsymbol{X}$ which matches better with the models Λ_X, and 2) by a model transformation function $G_\eta(.)$ that maps Λ_X to the transformed model Λ_Y which matches better with the utterance $\boldsymbol{Y}$. We assume the functional form of the transformations $F_\nu(.)$ or $G_\eta(.)$ and estimate the parameters ν or η in a maximum-likelihood manner using the expectation-maximization (EM) algorithm. The choice of the form of $F_\nu(.)$ or $G_\eta(.)$ is based on our prior knowledge of the nature of the acoustic mismatch. The stochastic matching algorithm operates only on the given test utterance and the given set of speech models, and no additional training data is required for the estimation of the mismatch prior to actual testing.

Experimental results are presented to study the properties of the proposed algorithm and to verify the efficacy of the approach in improving the performance of an HMM-based continuous speech recognition system in the presence of mismatch due to different transducers and transmission channels. The proposed stochastic matching algorithm is found to converge fast. Further, the recognition performance in mismatched conditions is greatly improved while the performance in matched conditions is well maintained. The stochastic matching algorithm was able to reduce the word error rate by about 70% in mismatched conditions.

[1] EDICS: SA 1.6.8

Permission to publish the Abstract is granted.

A Maximum-Likelihood Approach to Stochastic Matching for Robust Speech Recognition

Ananth Sankar [2] and Chin-Hui Lee

Speech Research Department
AT&T Bell Laboratories
Murray Hill, NJ 07974

1 Introduction

Recently there has been much interest in the problem of improving the performance of automatic speech recognition (ASR) systems in adverse environments. When there is a mismatch between the training and testing environments, ASR systems suffer a degradation in performance. The goal of robust speech recognition is to remove the effect of this mismatch so as to bring the recognition performance as close as possible to the matched conditions. In speech recognition, the speech is usually modeled by a set of hidden Markov models (HMM) Λ_X. During recognition the observed utterance $\boldsymbol{Y}$ is decoded using these models. Due to the mismatch between training and testing conditions, this often results in a degradation in performance compared to the matched conditions.

The mismatch between training and testing conditions can be viewed in the signal-space, the feature-space, or the model-space as shown in Figure 1. In Figure 1, S denotes the raw speech in the training environment. The mismatch between the training and testing environments is modeled by the distortion D_1 which transforms S to T. In speech recognition, some form of feature extraction is first carried out. These features are referred to as X in the training environment and Y in the testing environment. The mismatch between the two environments in the feature space is modeled by the function D_2 which transforms the features X to the features Y. Finally, the features X are used to build models Λ_X. The mismatch between the training and testing environments can be viewed in the model space as the transformation D_3 that maps Λ_X to Λ_Y. Some sources of mismatch include additive noise, channel and transducer mismatches which contribute a spectral tilt and a spectral shaping, speaker mismatch, different accents,

[2] Ananth Sankar is now with SRI International, Menlo Park, CA.

stress, and different speaking styles. Much of the recent work has been concentrated on the problems of additive noise and channel effects. For a detailed discussion of robust speech recognition in adverse environments, the reader is referred to [1].

Methods used to combat noise generally fall into three broad categories. In the first class of approaches, robust signal processing is used to reduce the sensitivity of the features to possible distortions. One approach involves spectral shaping such as liftering [2]. Here the idea is to deemphasize lower and higher order cepstral components because they have been found to be sensitive to channel and additive noise effects. Methods based on subtracting the long-term cepstral mean from the utterance have been proposed in [3]. This idea is commonly used to remove mismatches due to channels. Also in this class is the method presented in [4], where the spectral sequence is high-pass filtered to remove the effect of slowly varying channels. In methods based on auditory modeling [5, 6], signal processing is used that attempts to mimic the processing of the human ear in the hope that this will result in more robust features. The use of a signal-limiting preprocessor which can reduce the effect of noise on the speech features is discussed in [7]. Another approach to reducing the effect of noise is to inject a fraction of the ambient noise into the training data and retrain the system [8]. This technique is similar to *dithering*. There are also methods based on spectral subtraction [9, 10], where an estimate of the noise power spectrum is subtracted from each speech frame. This first class of approaches may be viewed as operating in the feature-space (Figure 1) as they generally involve some form of robust feature preprocessing.

In the second set of methods, some optimality criterion is used to form an estimate of a function of the clean speech. Formulations based on a minimum mean square error (MMSE) estimate of functions of the speech spectrum were presented in [11, 12, 13], where the corrupting noise was assumed to be an independent Gaussian process. Fur-

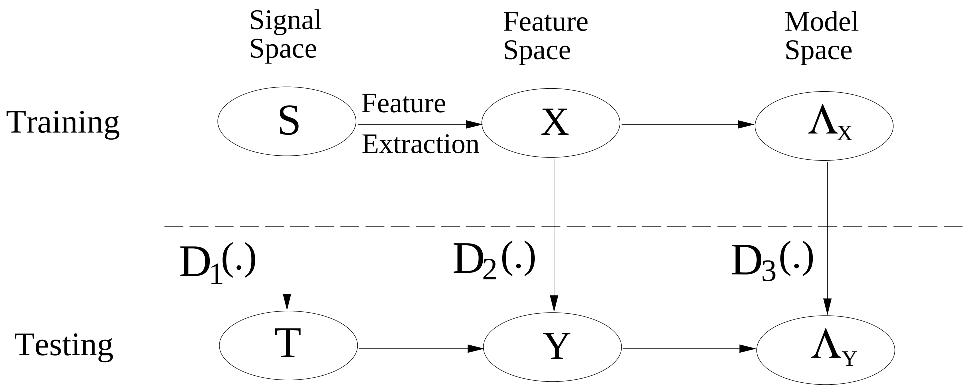

Figure 1: Mismatch in training and testing

thermore, each spectral bin was estimated separately, since it was assumed that the individual bins were independent. The correlation between separate spectral bins was modeled in [14], where the speech distribution was modeled as a mixture of Gaussians, with a diagonal covariance matrix for each mixture. Finally the temporal structure of speech was considered in [15, 16] by modeling the speech by a hidden Markov model (HMM). These approaches may be viewed either in the signal-space for speech enhancement, or in the feature-space for spectral compensation, depending on which representation is being estimated.

In the third set of methods, the noise is modeled and directly incorporated into the recognition process. One approach to this is based on noise masking [17, 18, 19]. Here the idea is to replace any filter bank energy by an appropriate noise level, if the signal energy falls below the noise level. Thus any information that is highly corrupted by noise is ignored. Such noise masking schemes were incorporated into an HMM speech recognizer in [20, 19]. In another approach called model decomposition [21, 22, 23], separate speech and noise HMMs are trained from training data. During recognition, the viterbi search is carried out in the combined state-space of the two models. This method has been shown to perform quite well, though it is necessary to have accurate models for both the speech and noise. The approach in [19] is similar to model decomposition. However, the original speech parameters are estimated from the noisy speech during recognition. The problem of estimating the original speech parameters from noisy data [19] was further studied in [24], where a more general interaction between the signal and noise models was allowed. In both [19] and [24], the signal is assumed to be modeled as a mixture of Gaussians. In yet another method in this framework [25], maximum-likelihood (ML) estimates of the energy contours for the training speech were used to normalize the speech before estimating the HMM parameters. During testing, ML estimates of the clean gain parameters were computed from the noisy speech, which were then used to normalize the parameters of the speech model. A minimax approach to robust speech recognition is presented in [26] where the recognizer is made more robust by allowing its HMM parameters to occupy some neighborhood of the values that were estimated during training. These set of approaches may be viewed as operating in the model-space to deal with the possible distortions in the models as shown in Figure 1.

There is some recent work on speaker and channel adaptation [27, 28, 29] where a fixed bias is estimated that transforms each individual speaker to a reference speaker and then the estimated bias is subtracted from every frame of speech. A similar approach has been used for estimating channel mismatch in speech recognition [30] where the speech is modeled by a vector quantization (VQ) codebook. Another approach to estimate channel mismatch has been proposed in [31, 32], where the estimation is based on the difference between the average log-spectra of the two channels [31, 32].

In this paper, we present an ML approach to stochastic matching for robust speech recognition. The method works by using an ML approach to reduce the mismatch between the observed utterance and the original speech models during recognition of the utterance. This mismatch may be reduced in two ways. First, we may map the distorted features $\boldsymbol{Y}$ to an estimate of the original features $\boldsymbol{X}$ so that the original models Λ_X can be used for recognition. Secondly, we can map the original models Λ_X to the transformed models Λ_Y which better match the observed utterance $\boldsymbol{Y}$. The first mapping operates in the feature-space, whereas the second operates in the model-space (see Figure 1). We denote these mappings as $F_\nu(Y)$ in the feature-space, and $G_\eta(\Lambda_X)$ in the model-space, where ν and η are parameters to be estimated. The functional form of these mappings will depend on our prior knowledge of the nature of the acoustic mismatch. The parameters, ν or η, of these functions are then estimated so as to maximize the likelihood of the observed speech $\boldsymbol{Y}$ given the models Λ_X, thus decreasing the mismatch due to the distortion. It is intuitively appealing to use the HMM's Λ_X as the speech models for estimating the parameters, ν and η, since the goal is to reduce the mismatch *for improved recognition*, and Λ_X are the models used for recognition. The ML parameter estimation is solved using the expectation-maximization (EM) algorithm to iteratively improve the likelihood. The stochastic matching algorithm operates only on the given test utterance and the given set of speech models, and no additional training data is required for the estimation of the mismatch prior to actual testing. The algorithm can also be used effectively when there is a larger amount of adaptation data available.

Experimental results are presented to show the efficacy of the approach to improve the performance of a continuous speech recognition system in the presence of mismatch due to different transducers and channels. This mismatch is modeled both as a fixed bias (in the feature space), and as a random bias (in the model space). The proposed approach reduced the word error rate by about 70% in mismatched conditions and maintained performance under matched conditions. The algorithm was also found to converge fast (within two iterations).

The remainder of this paper is organized as follows. In Section 2, we provide a general framework for the maximum likelihood estimation of the parameters of the transformations $F_\nu(.)$ and $G_\eta(.)$. In Section 3 we consider the case of the feature-space transformation. In particular, we derive expressions for the estimates of the parameters of an inverse distortion function that is linear in the unknown parameters, but non-linear in the observations. As a special case, an additive bias model is considered. In Section 4, the transformation is viewed in the model-space. In particular, we consider the case of a random additive bias distortion. Experimental results showing the efficacy of the method are provided in Section 5. Finally, we summarize our findings in Section 6.

2 The Stochastic Matching Framework

In pattern recognition, we are interested in the following problem. Given a set of trained models $\Lambda_X = \{\lambda_{x_i}\}$ where λ_{x_i} is the model of the ith class, and a set of test data $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_T\}$, we want to recognize the sequence of events $\boldsymbol{W} = \{W_1, W_2, \cdots, W_L\}$ embedded in $\boldsymbol{Y}$. For continuous speech recognition, λ_{x_i} may correspond to the ith subword HMM unit [33, 34], and $\boldsymbol{Y}$ to a particular test utterance. $\boldsymbol{W}$ could be the decoded phone or word sequence. In training the models Λ_X, we are limited to a set of training data. Unfortunately, there may be a mismatch between this training data and the test data $\boldsymbol{Y}$ which results in errors in the recognized sequence $\boldsymbol{W}$. This mismatch can be viewed in the original signal-space, the feature-space or in the model-space as shown in Figure 1. In the figure, the functions $D(.)$ map the original space to the corresponding distorted space. The sources of mismatch can be a distortion in the signal, incomplete characterization of the signal, insufficient amount of training data, or modeling inadequacy and estimation errors. In this paper, we are interested in the problem of *speech recognition* performance degradation due to mismatch in the training and test speech data. This mismatch could be due to microphone and channel mismatch, different environments for training and testing, different speakers and speaking styles or accents, or any combination of these.

In speech recognition, the models Λ_X are used to decode $\boldsymbol{Y}$ using the maximum a posteriori (MAP) decoder

$$\boldsymbol{W}' = \operatorname*{argmax}_{\boldsymbol{W}} p(\boldsymbol{Y}|\boldsymbol{W}, \Lambda_X) P(\boldsymbol{W}) \tag{1}$$

where the first term is the likelihood of observing $\boldsymbol{Y}$ given that the word sequence is $\boldsymbol{W}$, and the second term is the a priori probability of the word sequence $\boldsymbol{W}$. This second term is called the language model [35] which imposes constraints on the set of allowable word sequences. Due to the mismatch between training and testing environments, there is a corresponding mismatch in the likelihood of $\boldsymbol{Y}$ given Λ_X evaluated by Equation 1, thus causing errors in the decoded sequence $\boldsymbol{W}'$. We are motivated to decrease this mismatch, hence improving the recognition error rate.

In the feature-space, let the distortion function map the original utterance $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}$ into the sequence of observations $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_T\}$. If this distortion is invertible, then we may map $\boldsymbol{Y}$ back to the original speech $\boldsymbol{X}$ with an inverse function F_ν, so that

$$\boldsymbol{X} = F_\nu(\boldsymbol{Y}), \tag{2}$$

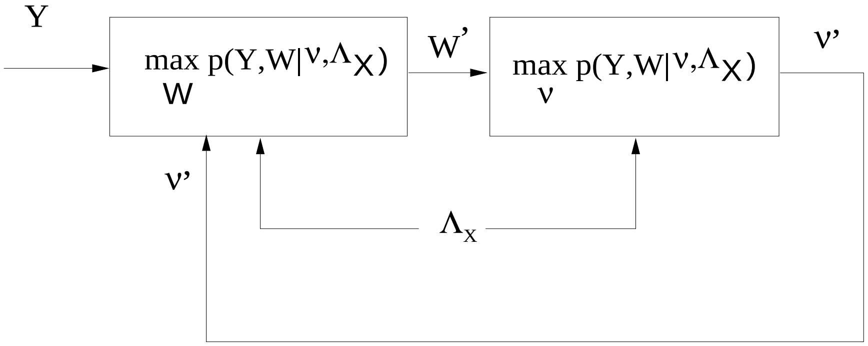

Figure 2: Joint Maximization of Equation 5

where ν are the parameters of the inverse distortion function. Alternately, in the model-space consider the transformation G_η with parameters η that maps Λ_X into the transformed models Λ_Y so that

$$\Lambda_Y = G_\eta(\Lambda_X). \tag{3}$$

One approach to decreasing the mismatch between $\boldsymbol{Y}$ and Λ_X is to find the parameters ν or η, and the word sequence $\boldsymbol{W}$ that maximize the joint likelihood of $\boldsymbol{Y}$ and $\boldsymbol{W}$ in Equation 1 given the models Λ_X. Thus in the feature-space, we need to find ν' such that

$$(\nu', \boldsymbol{W}') = \underset{(\nu, \boldsymbol{W})}{\operatorname{argmax}}\, p(\boldsymbol{Y}, \boldsymbol{W} | \nu, \Lambda_X), \tag{4}$$

which is equivalent to

$$(\nu', \boldsymbol{W}') = \underset{(\nu, \boldsymbol{W})}{\operatorname{argmax}}\, p(\boldsymbol{Y} | \boldsymbol{W}, \nu, \Lambda_X) P(\boldsymbol{W}). \tag{5}$$

Correspondingly, in the model-space, we need to find η' such that

$$(\eta', \boldsymbol{W}') = \underset{(\eta, \boldsymbol{W})}{\operatorname{argmax}}\, p(\boldsymbol{Y}, \boldsymbol{W} | \eta, \Lambda_X), \tag{6}$$

which is equivalent to

$$(\eta', \boldsymbol{W}') = \underset{(\eta, \boldsymbol{W})}{\operatorname{argmax}}\, p(\boldsymbol{Y} | \boldsymbol{W}, \eta, \Lambda_X) P(\boldsymbol{W}). \tag{7}$$

This joint maximization over the variables ν and $\boldsymbol{W}$ in Equation 5 or over η and $\boldsymbol{W}$ in Equation 7 may be done iteratively by keeping ν or η fixed and maximizing over $\boldsymbol{W}$, and then keeping $\boldsymbol{W}$ fixed and maximizing over ν or η. This procedure is conceptually shown in Figure 2 for the feature-space and in Figure 3 for the model-space.

The process of finding $\boldsymbol{W}$ has been treated by many researchers [35, 36, 37, 38]. In this paper, we are interested in the problem of finding the parameters ν and η.

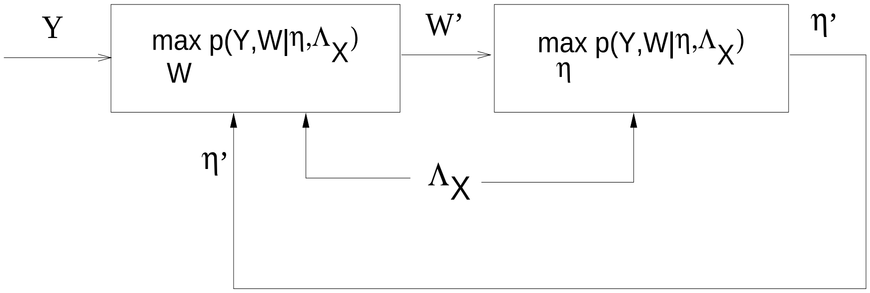

Figure 3: Joint Maximization of Equation 7

To simplify expressions, we remove the dependence on $\boldsymbol{W}$, and write the maximum-likelihood estimation problem corresponding to Equations 5 and 7 as

$$\nu' = \operatorname*{argmax}_{\nu} p(\boldsymbol{Y}|\nu, \Lambda_X), \tag{8}$$

and

$$\eta' = \operatorname*{argmax}_{\eta} p(\boldsymbol{Y}|\eta, \Lambda_X). \tag{9}$$

For this study, we assume that Λ_X is a set of left to right continuous density subword HMMs [33]. The transition probability from state i to j is denoted by $a_{i,j}$ for $i, j = 1, \cdots, N$, and the observation density $p_{\boldsymbol{x}}(\boldsymbol{x}|i)$ for state i is assumed to be a mixture of Gaussians, given by

$$p_{\boldsymbol{x}}(\boldsymbol{x}|i) = \sum_{j=1}^{M} w_{i,j} N(\boldsymbol{x}; \boldsymbol{\mu}_{i,j}, \boldsymbol{C}_{i,j}), \tag{10}$$

where M is the number of mixtures, $w_{i,j}$ is the probability of mixture j in state i, and N is the normal distribution given by

$$N(\boldsymbol{x}; \boldsymbol{\mu}_{i,j}, \boldsymbol{C}_{i,j}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\boldsymbol{C}_{i,j}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_{i,j})^T \boldsymbol{C}_{i,j}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_{i,j})\right), \tag{11}$$

where D is the dimension of the feature vector $\boldsymbol{x}$, and $\boldsymbol{C}_{i,j}$ and $\boldsymbol{\mu}_{i,j}$ are the covariance matrix and mean vector corresponding to mixture j in state i.

Let $S = \{s_1, s_2, \cdots, s_T\}$ be the set of all possible state sequences for the set of models Λ_X and $C = \{c_1, c_2, \cdots, c_T\}$ be the set of all mixture component sequences. Then Equation 8 can be written as

$$\nu' = \operatorname*{argmax}_{\nu} p(\boldsymbol{Y}|\nu, \Lambda_X) = \operatorname*{argmax}_{\nu} \sum_{S} \sum_{C} p(\boldsymbol{Y}, S, C|\nu, \Lambda_X). \tag{12}$$

Similarly, we may write Equation 9 as

$$\eta' = \operatorname*{argmax}_{\eta} \sum_{S} \sum_{C} p(\boldsymbol{Y}, S, C|\eta, \Lambda_X). \tag{13}$$

In general, it is not easy to estimate ν' or η' directly. However, for some F_ν and G_η, we can use the EM algorithm [39] to iteratively improve on a current estimate and obtain a new estimate such that the likelihoods in Equations 12 and 13 increase at each iteration. In the next two sections, we discuss the application of the EM algorithm to find the estimates of the parameters ν of the feature-space transformation F_ν, and the parameters η of the model-space transformation G_η respectively.

3 Estimation of Feature-Space Transformation F_ν

In this section we use the EM algorithm to find the estimates ν' of Equation 8. The EM algorithm is a two-step iterative procedure. In the first step, called the expectation step (E step), we compute the auxiliary function given by

$$\begin{aligned} Q(\nu'|\nu) &= E\{\log p(\boldsymbol{Y}, S, C|\nu', \Lambda_X)|\boldsymbol{Y}, \nu, \Lambda_X\} \\ &= \sum_S \sum_C p(\boldsymbol{Y}, S, C|\nu, \Lambda_X) \log p(\boldsymbol{Y}, S, C|\nu', \Lambda_X). \end{aligned} \tag{14}$$

In the second step, called the maximization step (M step), we find the value of ν' that maximizes $Q(\nu'|\nu)$, i.e.

$$\nu' = \operatorname*{argmax}_{\nu'} Q(\nu'|\nu) \tag{15}$$

It can be shown [39, 40] that if $Q(\nu'|\nu) \geq Q(\nu|\nu)$ then $p(\boldsymbol{Y}|\nu', \Lambda_X) \geq p(\boldsymbol{Y}|\nu, \Lambda_X)$. Thus iteratively applying the E and M steps of Equations 14 and 15 guarantees that the likelihood is nondecreasing. The iterations are continued until the increase in the likelihood is less than some predetermined threshold.

In general, the function $F_\nu(.)$ of Equation 2 can map a block of $\boldsymbol{Y}$ into a block of $\boldsymbol{X}$ of different size. However, for simplicity, we assume that the function is such that it maps each frame of $\boldsymbol{Y}$ onto the corresponding frame of $\boldsymbol{X}$, so that we can write

$$\boldsymbol{x}_t = f_\nu(\boldsymbol{y}_t). \tag{16}$$

With Λ_X given by the continuous density HMMs as described before, we may rewrite the auxiliary function as [41]

$$Q(\nu'|\nu) = \sum_S \sum_C p(\boldsymbol{Y}, S, C|\nu, \Lambda_X) \log \Pi_{t=1}^T a_{s_{t-1},s_t} w_{s_t,c_t} p_{\boldsymbol{y}}(\boldsymbol{y}_t|s_t, c_t, \nu, \Lambda_X) \tag{17}$$

where $p_{\boldsymbol{y}}(\boldsymbol{y}_t|s_t, c_t, \nu, \Lambda_X)$ is the probability density function of the random variable $\boldsymbol{y}_t$. This can be derived from the density function of the random variable $\boldsymbol{x}_t$, given by Equation 11, and the relation $\boldsymbol{x}_t = f_\nu(\boldsymbol{y}_t)$. We may write the density of $\boldsymbol{y}_t$ as

$$p_{\boldsymbol{y}}(\boldsymbol{y}_t|s_t, c_t, \nu, \Lambda_X) = \frac{N(f_\nu(\boldsymbol{y}_t); \boldsymbol{\mu}_{s_t,c_t}, \boldsymbol{C}_{s_t,c_t})}{|J_\nu(\boldsymbol{y}_t)|}, \tag{18}$$

where $J_\nu(\boldsymbol{y}_t)$ is the Jacobian matrix whose (i,j)th entry is given by

$$J_{\nu,i,j} = \frac{\partial y_{t,i}}{\partial f_{\nu,j}(\boldsymbol{y}_t)}, \tag{19}$$

where $y_{t,i}$ is the ith component of $\boldsymbol{y}_t$, and $f_{\nu,j}(\boldsymbol{y}_t)$ is the jth component of $f_\nu(\boldsymbol{y}_t)$. We may now rewrite Equation 17 as

$$\begin{aligned} Q(\nu'|\nu) &= \sum_S \sum_C p(\boldsymbol{Y}, S, C|\nu, \Lambda_X) \\ &\quad \cdot \sum_{t=1}^{T} \left\{ \log a_{s_{t-1},s_t} + \log w_{s_t,c_t} + \log \frac{N(f_{\nu'}(\boldsymbol{y}_t); \boldsymbol{\mu}_{s_t,c_t}, \boldsymbol{C}_{s_t,c_t})}{|J_{\nu'}(\boldsymbol{y}_t)|} \right\} \end{aligned} \tag{20}$$

which may be written as (see also [41])

$$\begin{aligned} Q(\nu'|\nu) &= \sum_{n=1}^{N} p(\boldsymbol{Y}, s_1 = n|\nu, \Lambda_X) \log a_{s_0,n} \\ &\quad + \sum_{t=2}^{T} \sum_{n=1}^{N} \sum_{l=1}^{N} p(\boldsymbol{Y}, s_t = n, s_{t-1} = l|\nu, \Lambda_X) \log a_{l,n} \\ &\quad + \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{M} p(\boldsymbol{Y}, s_t = n, c_t = m|\nu, \Lambda_X) \log w_{n,m} \\ &\quad + \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{M} p(\boldsymbol{Y}, s_t = n, c_t = m|\nu, \Lambda_X) \log N(f_{\nu'}(\boldsymbol{y}_t); \boldsymbol{\mu}_{n,m}, \boldsymbol{C}_{n,m}) \\ &\quad - \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{M} p(\boldsymbol{Y}, s_t = n, c_t = m|\nu, \Lambda_X) \log |J_{\nu'}(\boldsymbol{y}_t)|. \end{aligned} \tag{21}$$

Here $a_{s_0,n}$ is the initial probability for state n. In computing the auxiliary function of Equation 21, we are only interested in the terms involving ν'. Thus, using Equation 11, we may write the auxiliary function as

$$Q(\nu'|\nu) = \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_t(n,m) \left[-\frac{1}{2} (f_{\nu'}(\boldsymbol{y}_t) - \boldsymbol{\mu}_{n,m})^T \boldsymbol{C}_{n,m}^{-1} (f_{\nu'}(\boldsymbol{y}_t) - \boldsymbol{\mu}_{n,m}) - \log |J_{\nu'}(\boldsymbol{y}_t)| \right] \tag{22}$$

where $\gamma_t(n,m) = p(\boldsymbol{Y}, s_t = n, c_t = m|\nu, \Lambda_X)$ is the joint likelihood of $\boldsymbol{Y}$ and mixture m from state n producing the observation $\boldsymbol{y}_t$. We may compute the probability $\gamma_t(n,m)$ using the forward-backward algorithm [34, 41] as

$$\gamma_t(n,m) = \alpha_t(n)\beta_t(n) \frac{w_{n,m} N(f_\nu(\boldsymbol{y}_t); \boldsymbol{\mu}_{n,m}, \boldsymbol{C}_{n,m})}{\sum_{j=1}^{M} w_{n,j} N(f_\nu(\boldsymbol{y}_t); \boldsymbol{\mu}_{n,j}, \boldsymbol{C}_{n,j})}, \tag{23}$$

where

$$\begin{aligned} \alpha_t(n) &= p(\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_t, s_t = n|\nu, \Lambda_X), & (24) \\ \beta_t(n) &= p(\boldsymbol{y}_{t+1}, \boldsymbol{y}_{t+2}, \cdots, \boldsymbol{y}_T|\nu, s_t = n, \Lambda_X). & (25) \end{aligned}$$

The forward-backward algorithm can be used to iteratively compute $\alpha_t(n)$ and $\beta_t(n)$ as detailed in [34, 41].

In order to find the maximum of $Q(\nu'|\nu)$ with respect to ν', we may use any hill climbing technique such as the gradient ascent algorithm. However, for some cases, an explicit solution can be derived by differentiating the right hand side of Equation 22 with respect to ν' and solving for its zeros, i.e., we find ν' such that

$$\frac{\partial}{\partial \nu'} \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_t(n,m) \left[-\frac{1}{2} (f_{\nu'}(\boldsymbol{y}_t) - \boldsymbol{\mu}_{n,m})^T \boldsymbol{C}_{n,m}^{-1} (f_{\nu'}(\boldsymbol{y}_t) - \boldsymbol{\mu}_{n,m}) - \log |J_{\nu'}(\boldsymbol{y}_t)| \right] = 0. \tag{26}$$

The E.M. algorithm to maximize the likelihood in Equation 8 proceeds by computing $Q(\nu'|\nu)$ from Equation 22 (the E step), and then finding ν' from Equation 26 (the M step). This value is then substituted for ν in Equation 22, and the algorithm proceeds iteratively.

In analogy with the segmental k-means algorithm [42], we may use a segmental ML approach to maximize the joint likelihood of the observations and the state sequence S, $p(\boldsymbol{Y}, S|\nu, \Lambda_X)$ instead of directly maximizing the likelihood $p(\boldsymbol{Y}|\nu, \Lambda_X)$ in Equation 12. The iterative estimation procedure now becomes

$$\begin{aligned} S^l &= \operatorname*{argmax}_S p(\boldsymbol{Y}, S|\nu^l, \Lambda_X) && (27) \\ \nu^{l+1} &= \operatorname*{argmax}_\nu p(\boldsymbol{Y}, S^l|\nu, \Lambda_X). && (28) \end{aligned}$$

Thus we first find the most likely state sequence S^l, and then find ν^{l+1} to maximize the likelihood of the utterance $\boldsymbol{Y}$ conditioned on this state sequence. The Viterbi algorithm [43] may be used to find the optimal state sequence S^l, and the EM algorithm may be used to find ν^{l+1}. It is easy to show that the EM procedure described above still holds, except that now $\gamma_t(n,m)$ is defined by

$$\gamma_t(n,m) = \begin{cases} \frac{w_{n,m} N(f_\nu(\boldsymbol{y}_t); \boldsymbol{\mu}_{n,m}, \boldsymbol{C}_{n,m})}{\sum_{j=1}^{M} w_{n,j} N(f_\nu(\boldsymbol{y}_t); \boldsymbol{\mu}_{n,j}, \boldsymbol{C}_{n,j})} & \text{if } s_t^l = n \\ 0 & \text{otherwise.} \end{cases} \tag{29}$$

We make the simplifying assumptions that $f_\nu(\boldsymbol{y}_t)$ operates separately on each component, i.e., $x_{t,i} = f_{\nu,i}(y_{t,i})$, and that the covariance matrices $C_{n,m}$ are diagonal, i.e., $C_{n,m} = \text{diag}(\boldsymbol{\sigma}_{n,m}^2)$. In what follows, for ease of the expressions, we drop the reference to the subscript i denoting the ith component of the vectors. We consider functions of the form

$$f_\nu(y_t) = a\text{g}(y_t) + b, \tag{30}$$

where $\text{g}(y_t)$ is a known (possibly non-linear) differentiable function of y_t, and $\nu = \{a, b\}$ is the set of unknown parameters. The auxiliary function of Equation 22 can now be written as

$$Q(a', b'|a, b) = \sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n, m)\left[-\frac{1}{2}\frac{(a'\mathrm{g}(y_t) + b' - \mu_{n,m})^2}{\sigma^2_{n,m}} + \log a'\right]. \tag{31}$$

Taking the derivative of Equation 31 with respect to a' and b' respectively and setting to zero, we get

$$\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n, m)\left[\frac{1}{a'} - \frac{(a'\mathrm{g}(y_t) + b' - \mu_{n,m})\,\mathrm{g}(y_t)}{\sigma^2_{n,m}}\right] = 0, \tag{32}$$

and

$$\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n, m)\left[(a'\mathrm{g}(y_t) + b' - \mu_{n,m})\,/\sigma^2_{n,m}\right] = 0. \tag{33}$$

We can solve equation 32 and 33 explicitly for the estimates a' and b'.

We now consider a specific case of Equation 30 corresponding to an additive bias $\boldsymbol{b}_t$ so that

$$\boldsymbol{x}_t = \boldsymbol{y}_t - \boldsymbol{b}_t. \tag{34}$$

We can see that if $a = 1$, $\mathrm{g}(y_t) = y_t$, and $b = -b_t$, for each component, then Equation 34 is equivalent to Equation 30. If the observations are in the spectral domain, then $\boldsymbol{b}_t$ can be interpreted as an additive noise spectrum, whereas if the observations are in the cepstral or log-energy domains then $\boldsymbol{b}_t$ corresponds to a convolutional filtering effect due to, for example, transducers or channels.

We may model the bias $\boldsymbol{b}_t$ as either fixed for an utterance or varying with time. Some examples of a time-varying bias are a piecewise-constant bias, or a signal state-dependent bias. Alternatively we may model the bias stochastically in which case the distortion is viewed in the model-space as detailed in Section 4. In this section we address the cases of a state-dependent bias and a fixed bias.

We consider first the state-dependent case, in which the bias is varying from one HMM state to another. Suppose each speech state n has associated with it a specific bias term $\boldsymbol{b}_n$. Then we may write the auxiliary function of Equation 22 as

$$Q(\boldsymbol{b}'|\boldsymbol{b}) = \sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n, m)\left[-\sum_{i=1}^{D}\frac{(y_{t,i} - b'_{n,i} - \mu_{n,m,i})^2}{2\sigma^2_{n,m,i}}\right], \tag{35}$$

where $\boldsymbol{b} = \{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n, \cdots, \boldsymbol{b}_N\}$. The reestimation procedure of Equation 26 requires computing the derivative of Equation 35 with respect to $b'_{n,i}$ and equating to zero. This results in

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_t(n, m)(y_{t,i} - b'_{n,i} - \mu_{n,m,i})/\sigma^2_{n,m,i} = 0, \tag{36}$$

which gives

$$b'_{n,i} = \frac{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(n,m)(y_{t,i} - \mu_{n,m,i})/\sigma^2_{n,m,i}}{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(n,m)/\sigma^2_{n,m,i}}. \tag{37}$$

For the case of a single fixed bias $\boldsymbol{b}$, we may similarly show that

$$b'_{i} = \frac{\sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_t(n,m)(y_{t,i} - \mu_{n,m,i})/\sigma^2_{n,m,i}}{\sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_t(n,m)/\sigma^2_{n,m,i}}. \tag{38}$$

We can see from Equation 37 that estimation problems may arise due to small sample effects if there are many state-dependent bias terms to be estimated. However in some situations a state-dependent bias is meaningful. As an example, an additive cepstral bias model for linear filtering is only valid for high signal to noise ratios (SNR). When the SNR is low, the noise dominates, and the additive bias model for channel filtering is inaccurate. One way to handle this is to assume that the bias is SNR dependent as in [32], and estimate a different bias according to different SNR ranges. One implementation of such an approach that we have considered is to estimate a separate bias for speech and background segments. In our experiments, this was found to be useful for the case where part of the mismatch was caused by a telephone channel. This is probably due to additive noise present in the telephone channel. Details of the results will be discussed in Section 5.

4 Estimation of Model-Space Transformation G_η

In the previous section, the distorted speech was assumed to be a fixed function of the original speech. Our treatment in this section assumes that the observed utterance is a function of the original speech and the distortion, both of which are stochastic processes. The probability densities for the observed speech are then derived from those of the original speech and the distortion. A more general treatment of the model-space case may be found in [26] where no assumption is made about the underlying feature-space transformation.

Let the observation sequence $\boldsymbol{Y} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T\}$ be related to the original utterance $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$ and the distortion sequence $\boldsymbol{B} = \{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_T\}$ by

$$\boldsymbol{y}_t = f(\boldsymbol{x}_t, \boldsymbol{b}_t), \tag{39}$$

Then, if $\boldsymbol{x}_t$ and $\boldsymbol{b}_t$ are independent, we can write the probability density function (pdf) of $\boldsymbol{y}_t$ as

$$p(\boldsymbol{y}_t) = \int \int_{H_t} p(\boldsymbol{x}_t) p(\boldsymbol{b}_t) d\boldsymbol{x}_t d\boldsymbol{b}_t. \tag{40}$$

where H_t is the contour given by Equation 39.

As before, $\boldsymbol{X}$ is modeled by the set of HMMs Λ_X. Let the statistical model of $\boldsymbol{B}$ be given by Λ_B. Λ_B could be an HMM as in [20, 21] or a mixture Gaussian density as in [24]. In this study, we assume that Λ_B is a single Gaussian density with a diagonal covariance matrix, i.e.,

$$p(\boldsymbol{b}_t) = N(\boldsymbol{b}_t; \boldsymbol{\mu}_b, \boldsymbol{\sigma}_b^2). \tag{41}$$

Furthermore, as in Equation 34, we consider the specific case of an additive bias, so that the contour H_t is given by

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{b}_t. \tag{42}$$

Under these assumptions, the structure of Λ_Y remains the same as that of Λ_X. The means and variances of each mixture component in Λ_Y are derived by adding the mean $\boldsymbol{\mu}_b$ and the variance $\boldsymbol{\sigma}_b^2$ to the means and variances of the corresponding mixture components in Λ_X, i.e.,

$$\begin{aligned} \boldsymbol{\mu}_y &= \boldsymbol{\mu}_x + \boldsymbol{\mu}_b & (43)\\ \boldsymbol{\sigma}_y^2 &= \boldsymbol{\sigma}_x^2 + \boldsymbol{\sigma}_b^2. & (44) \end{aligned}$$

Equations 43 and 44 define the model transformation $G_\eta(.)$ of Equation 3, with the parameters η being given by $\boldsymbol{\mu}_b$ and $\boldsymbol{\sigma}_b^2$. If Λ_B is more complex such as an HMM or a mixture Gaussian density, then the structure of Λ_Y will be different from that of Λ_X in that the number of states and mixture components will be different.

We may now write the parameter estimation problem of Equation 13 as

$$\begin{aligned} \eta' &= (\boldsymbol{\mu}_b', \boldsymbol{\sigma}_b^{2\prime}) = \operatorname*{argmax}_{\boldsymbol{\mu}_b, \boldsymbol{\sigma}_b^2} \sum_S \sum_C p(\boldsymbol{Y}, S, C | \eta, \Lambda_X) \\ &= \operatorname*{argmax}_{\boldsymbol{\mu}_b, \boldsymbol{\sigma}_b^2} \sum_S \sum_C \Pi_{t=1}^T a_{s_{t-1}, s_t} w_{s_t, c_t} N(\boldsymbol{y}_t; \boldsymbol{\mu}_{s_t, c_t} + \boldsymbol{\mu}_b, \boldsymbol{\sigma}_{s_t, c_t}^2 + \boldsymbol{\sigma}_b^2). \end{aligned} \tag{45}$$

Again, the EM algorithm can be used to iteratively estimate $\boldsymbol{\mu}_b$ and $\boldsymbol{\sigma}_b^2$. It is easy to show that the auxiliary function corresponding to Equation 21 can be written as

$$Q(\eta'|\eta) = -\sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_t(n,m) \sum_{i=1}^{D} \left[\frac{1}{2} \log(\sigma_{n,m,i}^2 + \sigma_{b_i}^2{}') + \frac{(y_{t,i} - \mu_{b_i}' - \mu_{n,m,i})^2}{2(\sigma_{n,m,i}^2 + \sigma_{b_i}^2{}')} \right]. \tag{46}$$

Maximizing this function with respect to $\eta' = (\boldsymbol{\mu}_b', \boldsymbol{\sigma}_b^{2\prime})$ does not result in a closed form expression for $\sigma_{b_i}^2{}'$ though a similar expression as in Equation 38 can be found for μ_{b_i}' so that

$$\mu_{b_i}' = \frac{\sum_{t=1}^T \sum_{n=1}^N \sum_{m=1}^M \gamma_t(n,m)(y_{t,i} - \mu_{n,m,i})/(\sigma_{n,m,i}^2 + \sigma_{b_i}^2{}')}{\sum_{t=1}^T \sum_{n=1}^N \sum_{m=1}^M \gamma_t(n,m)/(\sigma_{n,m,i}^2 + \sigma_{b_i}^2{}')}. \tag{47}$$

One approach to the problem of estimating $\sigma_{b_i}^2$ is to assume that the variances σ_b^2 are signal state dependent and are related to the signal variances by

$$\sigma_{b,n,m,i}^2 = \alpha_i \sigma_{n,m,i}^2, \tag{48}$$

where α_i is a scaling factor for the ith component of the variance. Substituting Equation 48 in Equation 46 and maximizing with respect to α_i, we get

$$1+\alpha_i = \frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n,m)\frac{(y_{t,i}-\mu_{n,m,i}-\mu'_{b_i})^2}{\sigma^2_{n,m,i}}. \tag{49}$$

We note that the assumption of Equation 48 violates our previous assumption of independence between $\boldsymbol{x}_t$ and $\boldsymbol{b}_t$. However, in many real situations, the signal and distortion signal may, in fact, be related. In addition, the assumption of Equation 48 allows for a simple closed-form estimate of the parameters $\boldsymbol{\mu}_b$ and $\boldsymbol{\sigma}_b^2$ as shown in Equations 47 and 49. We also see that $\alpha_i > -1$ allows for both a variance expansion ($\alpha > 0$) and a variance contraction ($\alpha < 0$).

An alternate approach consistent with Equation 42 is to write the likelihood $p(\boldsymbol{Y}|\eta,\Lambda_X)$ as

$$p(\boldsymbol{Y}|\eta,\Lambda_X) = \sum_S\sum_C\int\int_H p(\boldsymbol{X},\boldsymbol{B},S,C|\eta,\Lambda_X)d\boldsymbol{X}d\boldsymbol{B}. \tag{50}$$

where $\int\int_H$ is the T-fold integral given by

$$\int\int_H d\boldsymbol{X}d\boldsymbol{B} = \Pi_{t=1}^{T}\int\int_{H_t} d\boldsymbol{x}_t d\boldsymbol{b}_t. \tag{51}$$

Correspondingly, we can define a new auxiliary function [24] as

$$Q(\eta'|\eta) = \sum_S\sum_C\int\int_H p(\boldsymbol{X},\boldsymbol{B},S,C|\eta,\Lambda_X)\log p(\boldsymbol{X},\boldsymbol{B},S,C|\eta',\Lambda_X)d\boldsymbol{X}d\boldsymbol{B}. \tag{52}$$

This approach has been taken to derive general expressions for the estimates of the parameters of an original speech model given noisy observations in [24], where both the speech and distortion were modeled by a mixture Gaussian distribution. The problem addressed in this section is a reverse one of finding the parameters of the distortion model given the distorted speech. In this paper, the speech model Λ_X is an HMM and the additive distortion is modeled as a single Gaussian density as in Equation 41. Under these conditions, we can use the derivations of [24] to get the reestimation formulas

$$\begin{aligned}
\mu'_{b_i} &= \frac{\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n,m)E\left(b_{t,i}|y_{t,i},s_t=n,c_t=m,\eta,\Lambda_X\right)}{\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n,m)} && (53)\\
\sigma^2_{b_i}{}' &= \frac{\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n,m)E\left(b^2_{t,i}|y_{t,i},s_t=n,c_t=m,\eta,\Lambda_X\right)}{\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n,m)} - \mu'_{b_i}{}^2. && (54)
\end{aligned}$$

As before, $\gamma_t(n,m) = p(\boldsymbol{Y},s_t=n,c_t=m|\eta,\Lambda_X)$ is the joint likelihood of $\boldsymbol{Y}$ and the mth mixture component of the nth state in the *transformed model* $\Lambda_Y = G_\eta(\Lambda_X)$ producing the observation $\boldsymbol{y}_t$. The conditional expectations in Equation 53 and Equation 54 can be evaluated as [24]

$$
\begin{aligned}
E\left(b_{t,i}|y_{t,i}, s_t = n, c_t = m, \eta, \Lambda_X\right) &= \mu_{b_i} + \frac{\sigma_{b_i}^2}{\sigma_{n,m,i}^2 + \sigma_{b_i}^2}\left(y_{t,i} - \mu_{n,m,i} - \mu_{b_i}\right), \quad (55) \\
E\left(b_{t,i}^2|y_{t,i}, s_t = n, c_t = m, \eta, \Lambda_X\right) &= \frac{\sigma_{b_i}^2 \sigma_{n,m,i}^2}{\sigma_{b_i}^2 + \sigma_{n,m,i}^2} + \quad (56) \\
& \left\{E\left(b_{t,i}|y_{t,i}, s_t = n, c_t = m, \eta, \Lambda_X\right)\right\}^2.
\end{aligned}
$$

Examining Equation 55, we may make some observations as to the convergence of the EM algorithm. We note that if $\sigma_{b_i}^2$ is small, then the convergence is slow. This was found to be the case in our experiments, described in Section 5, as the variance in the mismatch due to the different transducers and transmission channels was small. In the limiting case of a deterministic bias ($\sigma_{b_i}^2 = 0$), the estimate does not change at all. This may be remedied by using Equation 47 to estimate $\boldsymbol{\mu}_b$ and Equation 54 to estimate $\boldsymbol{\sigma}_b^2$.

We have shown how to estimate the bias parameters in the feature space and in the model space under the additive model of Equation 34. However the additive bias model was only applied to the cepstral features. In our experiments in addition to the cepstral features, we have used delta and delta-delta cepstral features and delta and delta-delta log energy features. In our stochastic matching algorithm, we do not transform the delta and delta-delta log energy features. However, the effect of the mismatch on the delta cepstrum and the delta-delta cepstrum is taken into account. For the feature space bias model, we assume that the delta and delta-delta cepstral features are unaffected by the mismatch, i.e., the delta and delta-delta bias vectors are zero. This is a meaningful assumption if we assume that the bias in the cepstrum is fixed for the entire utterance. Similarly, for the model space, we assume that the delta and delta-delta mean vectors are zero. However, this is not the case for the delta and delta-delta variances. These variance vectors may be estimated as follows. We use the fact that the delta cepstrum is computed according to

$$
\Delta C_{l,m} = \sum_{k=-K}^{K} G k C_{l-k,m}, \quad (57)
$$

where $\Delta C_{l,m}$ and $C_{l,m}$ are the mth delta cepstral and the mth cepstral coefficients, respectively, for the lth time frame. G is a gain term fixed at 0.375, and $K = 2$. The delta-delta cepstrum is computed according to

$$
\Delta^2 C_{l,m} = \sum_{n=-N}^{N} G n \Delta C_{l-n,m}, \quad (58)
$$

where $\Delta^2 C_{l,m}$ is the mth delta-delta cepstral coefficient for the lth time frame. We choose $G = 0.375$ and $N = 1$. Assuming that the cepstral coefficients for different frames are independent, we may write the variance of the delta cepstrum in terms of the variance of the cepstrum as follows:

$$\sigma^2_{\Delta C_{l,m}} = \sum_{k=-K}^{K} G^2 k^2 \sigma^2_{C_{l-k,m}} \tag{59}$$

where $\sigma^2_{\Delta C_{l,m}}$ and $\sigma^2_{C_{l,m}}$ are the variances of the mth component of delta cepstrum and cepstrum of the lth time frames. Similarly we derive the variances of the delta-delta cepstrum as

$$\sigma^2_{\Delta^2 C_{l,m}} = \sum_{n=-N}^{N} \sum_{k=-K}^{K} G^4 n^2 k^2 \sigma^2_{C_{l-k-n,m}} \tag{60}$$

We are interested in estimating the variances of the delta and delta-delta bias terms. Since we have assumed that the bias is i.i.d. Gaussian, with a variance of $\boldsymbol{\sigma}^2_b$, hence we may estimate the variance of the ith component of the delta bias using Equations 59 as

$$\sigma^2_{\Delta b_i} = \sum_{k=-K}^{K} G^2 k^2 \sigma^2_{b_i}. \tag{61}$$

Similarly, we estimate the variance of the ith component of the delta-delta bias as

$$\sigma^2_{\Delta^2 b_i} = \sum_{n=-N}^{N} \sum_{k=-K}^{K} G^4 n^2 k^2 \sigma^2_{b_i}. \tag{62}$$

We conclude this section by observing that the statistical model used for the distortion was a simple Gaussian density. One example of the case of mixture Gaussian noise densities and also more general interactions between the speech and distortion is treated in [24]. The same stochastic matching algorithms discussed above can also be applied to these more general model transformation cases.

5 Experimental Results

We experimented with the 991-word DARPA resource management (RM) task [44]. New simultaneous recordings of two non-native male speakers were collected through two channels: 1) a close talking microphone (MIC), and 2) a telephone handset over a dial-up line (TEL). The data consisted of 300 utterances for training and adaptation purposes from each speaker (A and B) in each of the two channels (MIC and TEL). For testing, we collected 75 utterances from speaker A, and 67 utterances from speaker B for each of the channels (MIC and TEL).

For each frame, a 38-dimensional feature vector was extracted based on 10th order LPC analysis. The speech was first downsampled from 16 kHz to 8 kHz, and the analysis frames were 30 ms wide with 20 ms overlap. The features correspond to a 12-dimensional cepstrum vector, a 12-dimensional delta-cepstrum vector, a 12-dimensional delta-delta-cepstrum vector, a delta log energy feature, and a delta-delta log energy feature [45]. For recognition we used 1769 context dependent (CD) subword unit models, with a maximum of 16 mixture components per state [45]. For all our experiments, we used the RM word pair grammar which gives a perplexity of about 60.

In the following sections, we present an experimental study of the stochastic matching algorithm. In Section 5.1, several baseline experiments were run for different speaker (A and B) and channel (MIC and TEL) training and testing conditions. The results of this section show the need for performance improvement under mismatched channel conditions. In Section 5.2, the stochastic matching algorithm is used to improve the performance of the mismatched channel conditions. It is shown that an average word error rate reduction of about 70% is achieved in mismatched conditions, while under matched conditions the performance is well maintained.

5.1 Baseline Performance

We first conducted baseline experiments to study the effect of channel mismatch between training and testing conditions. Four different HMM sets were used. The first set consisted of speaker-independent male models. These models were created as follows. First speaker independent models were trained using the NIST/RM SI-109 training set consisting of 3990 utterances from 109 native American talkers (31 females and 78 males), each providing 30 or 40 utterances. These models were then adapted using a maximum a posteriori (MAP) adaptation algorithm [46], with the data from the 78 male talkers, to create speaker-independent male models. These models are denoted below as SI-M. Three speaker-adaptive models were created for each of the two non-native speakers using the MAP adaptation algorithm. For MAP adaptation, the SI-M models were used as seed models. The first two speaker-adaptive models for each speaker were obtained using the 300 training utterances collected from the microphone and telephone channels. These models are labeled MIC and TEL respectively. The third model was created using the pooled 600 utterances recorded simultaneously over both the MIC and TEL channels, i.e., 300 utterances from each channel. This model was labeled MIC+TEL. Table 1 shows the word error rate including insertion, deletion, and substitution errors, when the above four models were used for recognition on the test utterances from each channel (MIC and TEL) for each speaker (A and B).

As expected, the SI-M models perform poorly because the test speech is from non-native speakers. In previous experiments on speaker independent recognition for native speakers on this task, it was found that the typical word error rate was around 4% [47].

	SI-M	MIC	TEL	MIC+TEL
A-MIC	26.8	2.1	14.1	2.7
A-TEL	65.7	24.3	2.7	2.5
B-MIC	42.9	6.3	25.8	6.2
B-TEL	80.3	24.1	7.0	5.3

Table 1: Baseline Word Error Rate (percent) Performance

Clearly, the non-native speaker performance shown in Table 1 is far from this speaker independent performance. In the MAP speaker adaptation experiments reported in [48], it was shown that using 600 utterances for adaptation, the word error rate for native speakers could be brought down to about 1.5%. We can compare this to the performance of the MAP adaptation algorithm for the non-native speakers A and B using 300 utterances for adaptation. When the MAP speaker adaptation is done for each channel (MIC and TEL), and the test speech is from the corresponding channel, the error rate is drastically reduced, showing the efficacy of the MAP speaker adaptation algorithm [46]. We see that for speaker A, the word error rate is close to 2%, i.e., similar to the native speaker adaptation results reported in [48]. For speaker B the matched channel performance is worse (around 6.5%).

Table 1 also shows that channel mismatch causes severe degradation. For example, if the test speech is from the telephone channel (A-TEL), and the models are speaker-adaptive models for the microphone (MIC), then there is a significant degradation in performance (word error rate of 24.3%) due to a channel and microphone mismatch, although the speaker mismatch has been significantly reduced. In this paper, we are interested in reducing this mismatch, hence improving the error rate. One approach is to use the speaker-adaptive models created from the pooled telephone and microphone data (MIC+TEL). As can be seen from the right most column of Table 1, the word error rate for this case is close to the matched channel conditions. However, the problem with this approach is that it is necessary to collect enough data from each possible channel to adapt the pooled models. In particular, if the test utterance is from a new environment that is not covered in the collected data, then there will still be a degradation in performance due to channel mismatch. In addition, the model size needs to be increased in order to cover the large diversity in all training environments. The stochastic matching algorithm proposed in this paper provides a solution to this problem as it operates entirely on the test utterance, and does not require training data from the various possible environmental conditions. We now present an experimental study of this algorithm.

5.2 Performance of the Stochastic Matching Algorithm

We assume that the mismatch between the microphone (MIC) and telephone (TEL) channels can be modeled by a multiplicative spectrum in the spectral domain, and hence by an additive bias in the cepstral domain, as shown by Equation 34. We may treat this bias as fixed or random corresponding to the feature space and model space descriptions of Section 3 and 4 respectively. Furthermore, we may estimate different bias vectors for different signal states, both in the feature and model space. For example, Equation 37 gives the estimate of a state dependent bias vector in the feature space, whereas Equation 38 assumes a single bias for the entire utterance. It is easy to modify Equations 53 and 54 to get the state dependent model space estimates of the mean and variance vectors for a random bias. As we noted before, at the end of Section 3, a signal-dependent bias estimate would be useful, for example, when part of the distortion is due to additive noise. When noise dominates, such as in the silence regions of the utterance, the additive bias model for channel mismatch is inaccurate. We were thus motivated to estimate separate bias parameters for speech and silence frames, both in the signal and model space. An additional silence model in our HMM set was used to make the decision as to whether a particular frame was speech or silence.

In our experiments, the bias parameters were estimated on a per-utterance basis. In the feature space, the bias vector was initialized to zero. In the model space, the mean of the bias was initialized to zero, whereas the variance was initialized to a small positive number. As shown in Figures 2 and 3, we first decode the word string W, and then estimate the bias parameters conditioned on this word string. It is important to note that the recognition hypothesis W guides the algorithm and, hence, a very poor hypothesis can result in sub-optimal performance, especially if the number of signal state-dependent bias vectors is large. In practice, we can get around this problem by performing many iterations of the two-step procedure shown in Figures 2 and 3. In the beginning we use only a small number of state-dependent bias parameters, and increase the number of parameters with the number of iterations. This has the effect of averaging out the recognition errors for a smaller number of state-dependent parameters. As the iterations proceed in this way, the recognition hypothesis will hopefully improve, allowing for an increased number of state-dependent parameters.

The approach described above was used in our experiments. In the case of a single bias estimate for the entire utterance, only one cycle was used. However, for a separate speech and silence bias vector estimate, we proceeded as follows. In the first cycle of the process shown in Figures 2 and 3, we estimated *a single* bias parameter. In the *second cycle*, we estimated a separate speech and silence bias parameter. We used the segmental ML approach given in Equation 27 and 28 to estimate the bias parameters. Again, Equation 27 and 28 can be iterated many times. However, only one iteration of the segmental ML algorithm was used.

Before studying the performance of the stochastic matching algorithm on the database described above, we present the results of a simple experiment designed to study the estimation properties of the algorithm. We added a fixed bias vector to each frame of the 75 test utterances of speaker A recorded through the MIC channel to create 75 mismatched utterances. We then used the stochastic matching algorithm to estimate this bias for each of the 75 mismatched utterances using the MIC speaker adaptive models for speaker A. The initial estimates of the bias were fixed to be zero. Figure 4 shows the fixed bias that was added and the bias estimated by the algorithm averaged over all 75 utterances. The cepstrum index is plotted on the x axis and the value of the cepstrum is plotted on the y axis. It can be seen from the figure that the average estimated bias, indicated by the crosses, closely approximates the assumed fixed bias, indicated by the circles, for almost all cepstral indices. The estimation error is due to the fact that the models used to estimate the bias vector are obtained from a separate set of 300 utterances which do not exactly characterize the 75 test utterances. We also show in Figure 5 the convergence speed of the algorithm. In this figure, we plot the average log-likelihood versus the iteration number of the EM algorithm. The EM algorithm iterations to estimate the bias parameters are conditioned on the recognized word string as shown in Figures 2 and 3. It is seen from Figure 5 that the EM algorithm converges within two iterations.

Table 2 gives the percentage word error rates for speaker A and B under mismatched conditions after processing with two sets of *feature space* bias estimation approaches: 1) a *single* bias vector is estimated for the entire utterance (FS1), and 2) a *separate* vector is estimated for speech and silence frames (FS2). For reference, we also reproduce from Table 1 the mismatched performance (MIS), and the matched performance (MAT).

The results show that the estimation of a single bias in the feature space (FS1) significantly reduces the word error rate for both speakers. In addition, we see that estimating a separate bias for speech and silence frames (FS2) improves the performance for telephone speech (TEL) for both speakers. However no additional improvement was obtained for microphone speech (MIC). This result can be attributed to the fact that whereas the microphone speech is relatively noise-free, there is noise present in the telephone lines. Thus, we expect separate signal/silence bias estimation to result in larger performance improvement for telephone speech.

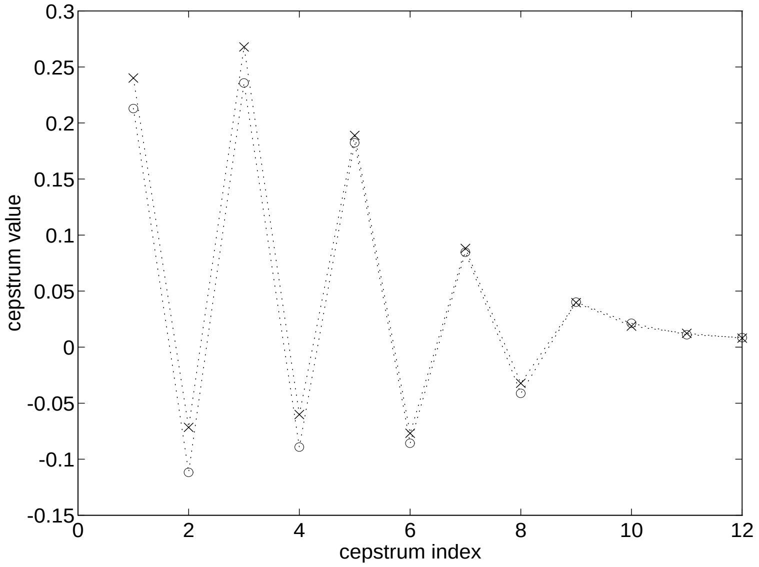

Figure 4: Estimation of a fixed bias. The circles indicate the values of the added bias, while the crosses indicate the estimated bias.

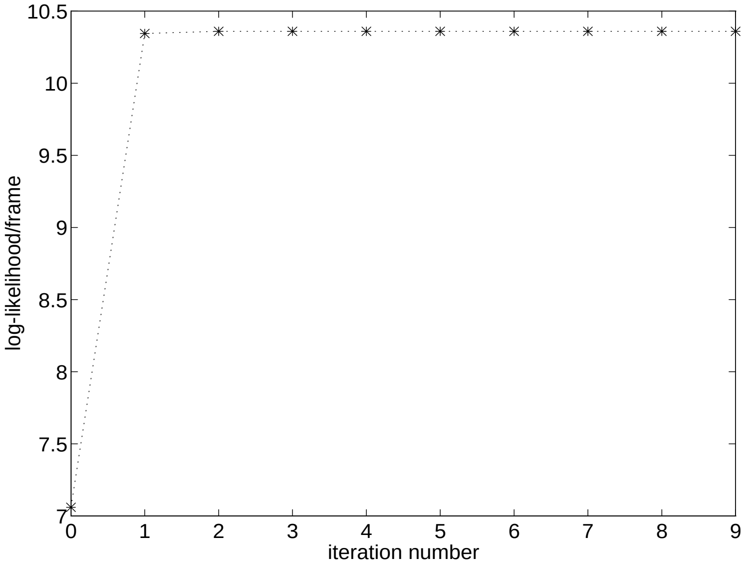

Figure 5: Convergence of Stochastic Matching Algorithm

	MIS	FS1	FS2	MAT
A-MIC	14.1	4.7	4.6	2.1
A-TEL	24.3	14.6	10.7	2.7
B-MIC	25.8	7.7	7.4	6.3
B-TEL	24.1	13.7	10.8	7.0

Table 2: Word Error Rate (percent) with Feature Space Bias Estimation

	MIS	FS1	FS2	MS1	MS2	MAT
A-MIC	14.1	4.7	4.6	4.1	4.1	2.1
A-TEL	24.3	14.6	10.7	9.6	7.1	2.7
B-MIC	25.8	7.7	7.4	6.8	6.3	6.3
B-TEL	24.1	13.7	10.8	8.9	7.4	7.0

Table 3: Word Error Rate (percent) with Feature and Model Space Bias Estimation

In Table 3, we reproduce the results of Table 2, and also give the results for two sets of model space bias estimation procedures: 1) a single mean and variance vector is estimated for the entire utterance (MS1), and 2) a separate mean and variance vector is estimated for speech and silence frames (MS2).

As in the feature space results (Table 2), the results of Table 3 show that, in the model space too, estimating separate speech and silence bias parameters (MS2) improves the performance for both speakers in the telephone speech (TEL), when compared to estimating one set of bias parameters (MS1). Again, for the microphone speech (MIC), separate speech and silence bias parameter estimates did not result in additional improvement. Table 3 also shows that model space bias estimation (MS1 and MS2) consistently improves performance over feature space estimation (FS1 and FS2). The best performance (MS2) corresponds to about a 70% reduction in the word error rate, when compared to the mismatched conditions, for all possible combinations of speakers and channels. Furthermore, we see that, for speaker B, the best stochastic matching performance (MS2), is almost as good as the matched conditions (MAT). However, for speaker A, there is still a performance gap.

A commonly used approach for improving ASR performance in mismatched chan-

	MIS	FS1	FS2	MS1	MS2	MAT
A-MIC	5.0	5.2	3.7	4.1	3.7	3.0
A-TEL	12.0	12.6	8.3	7.7	6.1	3.1
B-MIC	9.9	9.7	8.4	8.0	7.9	5.0
B-TEL	8.9	9.1	7.0	7.2	7.7	6.3

Table 4: Word Error Rate (percent) with Stochastic Matching After CMS Processing

nel conditions is cepstral mean subtraction (CMS), where the average cepstrum of the frames over the entire utterance is subtracted from each frame of the utterance. The stochastic matching approach can also be used to improve the performance *after CMS processing*. Table 4 shows the performance of the stochastic matching algorithm after processing with CMS. Note that the models used were also created with CMS speech. Thus the mismatched (MIS) and matched (MAT) performance were not the same as in the previous tables.

The mismatched (MIS) performance in Table 4 shows the word error rate under mismatched conditions, but after CMS processing. It is clear that CMS processing gives better performance under mismatched conditions compared to the case where no processing is performed (see the MIS column in Table 3). Comparisons between the single feature space bias estimate (FS1 in Table 3) and CMS (MIS in Table 4) do not clearly show the superiority of one over the other. However, the CMS results (MIS column in Table 4) were not as good compared to the stochastic matching algorithm results shown in Table 3 for the cases of two feature space bias estimates (FS2) and the model space approaches (MS1 and MS2).

As noted above, we can also apply the stochastic matching algorithm *after CMS processing*. These results are shown in Table 4. It can be seen that a single feature space bias estimate (FS1) results in similar performance to the mismatched case (MIS). This is not surprising, as the CMS processing has caused both the training and testing utterances to be zero mean. However, a separate speech and silence bias vector estimate (FS2) results in an additional performance improvement. Furthermore, the results show that the model space bias parameter estimation procedures (MS1 and MS2) also decrease the word error rate. When compared with the model space approaches without CMS processing (Table 3), there is no clear improvement shown in the MS1 and MS2 results listed in Table 4.

The above results show that the stochastic matching algorithm results in significant

	Without CMS		With CMS	
	MAT	MS1	MAT	MS1
A-MIC	2.1	2.2	3.0	3.1
A-TEL	2.7	2.5	3.1	3.1
B-MIC	6.3	5.5	5.0	5.3
B-TEL	7.0	6.5	6.3	5.0

Table 5: Word Error Rate (percent) with Stochastic Matching Under Matched Conditions

improvement under mismatched conditions. The best performance corresponds to about a 70% improvement over the mismatched conditions for all combinations of speakers and channels. Since, in a real application, we may not know whether the test utterance is from a mismatched or matched environment, we also tested the performance of the algorithm under matched conditions. Table 5 shows the word error rates under matched conditions (MAT), and after estimating a single mean and variance vector in the model space for the entire utterance (MS1), both with and without CMS processing. The results indicate that the stochastic matching algorithm maintains the performance under matched conditions. Similar results were also obtained for the other stochastic matching algorithms, i.e., FS1, FS2, and MS2.

We also experimented with the functional form given in Equation 30, by setting

$$\mathrm{g}(y_t) = y_t,$$

and using the stochastic matching algorithm to estimate the parameters a and b. We found that the likelihoods for the test utterances, using these two parameters, were always better than if only one parameter was estimated as in the fixed bias case. However, no additional recognition performance improvement over the fixed additive bias estimation approaches was observed on our database.

6 Summary and Conclusion

We have presented a novel approach to robust speech recognition based on an ML approach to stochastic matching. The algorithm operates entirely on the test utterance and does not require the use of a training database to estimate the mismatch. The mismatch can be modeled both deterministically (in the feature space) and probabilistically (in the model space). Experimental studies on a database where the mismatch was due to different transducers and channels showed that the algorithm provides a reduction of about 70% in the word error rate when compared to the performance under mismatched conditions. The algorithm also improved the performance after the speech was processed with cepstral mean subtraction (CMS). Under matched conditions, the performance was also well maintained. Finally, the convergence of the algorithm was found to be fast.

In our experimental results, we have studied additive bias distortions both in the feature and model space. We have shown how separate bias estimates for speech and silence can result in a performance improvement over the case in which only a single bias is used. One area of future research is to extend this idea to estimating distortion parameters based on broad speech classes. Another area of future research is to study non-linear distortion models and how the stochastic matching algorithm can be used to improve the recognition performance under such broad distortion conditions.

Acknowledgements

We thank C. Pao and five of our colleagues for helping with the recording of the non-native speaker RM data. We also thank Drs. B.S. Atal, B.-H. Juang, V. Nagesha, R. Rose, and M.M. Sondhi for useful discussions.

References

[1] B.-H. Juang, “Speech Recognition in Adverse Environments,” *Computer Speech and Language*, vol. 5, pp. 275–294, 1991.

[2] B.-H. Juang, L. R. Rabiner, and J. G. Wilpon, “On the Use of Bandpass Liftering in Speech Recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-35, pp. 947–954, July 1987.

[3] B. Atal, “Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification,” *Journal of the Acoustic Society of America*, vol. 55, pp. 1304–1312, June 1974.

[4] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, “Compensation For The Effect Of The Communication Channel In Auditory-Like Analysis Of Speech (RASTA-PLP),” in *Proceedings of EUROSPEECH*, pp. 1367–1370, 1991.

[5] O. Ghitza, "Auditory Nerve Representation as a Basis for Speech Processing," in *Advances in Speech Signal Processing* (S. Furui and M. Sondhi, eds.), pp. 453–485, Marcel Dekker, 1991.

[6] S. Seneff, "A Joint Synchrony/Mean-Rate Model of Auditory Speech Processing," *Journal of Phonetics*, vol. 16, pp. 55–76, 1990.

[7] C.-H. Lee and C.-H. Lin, "On The Use Of A Family Of Signal Limiters For Recognition Of Noisy Speech," *Speech Communication*, vol. 12, pp. 383–392, 1993.

[8] S. Das, R. Bakis, A. Nadas, D. Nahamoo, and M. Picheny, "Influence of Background Noise and Microphone on the Performance of the IBM TANGORA Speech Recognition System," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II–71–II–74, 1993.

[9] D. Van Compernolle, "Spectral Estimation Using a Log-Distance Error Criterion Applied To Speech Recognition," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 258–261, 1989.

[10] D. Van Compernolle, "Noise Adaptation in a Hidden Markov Model Speech Recognition System," *Computer Speech and Language*, vol. 3, pp. 151–167, 1989.

[11] J. E. Porter and S. F. Boll, "Optimal Estimators for Spectral Restoration of Noisy Speech," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 18.A.2.1–18.A.2.4, 1984.

[12] Y. Ephraim and D. Malah, "Speech Enhancement Using a Minimum Mean Square Error Short-Time Spectral Amplitude Estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, pp. 1109–1121, December 1984.

[13] Y. Ephraim and D. Malah, "Speech Enhancement Using a Minimum Mean Square Error Log-Spectral Amplitude Estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, pp. 443–445, April 1985.

[14] A. Erell and M. Weintraub, "Estimation Using Log-Spectral-Distance Criterion For Noise-Robust Speech Recognition," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 853–856, 1990.

[15] Y. Ephraim, "A Minimum Mean Square Error Approach For Speech Enhancement," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 829–832, 1990.

[16] A. Erell and M. Weintraub, "Filterbank-Energy Estimation Using Mixture and Markov Models for Recognition of Noisy Speech," *IEEE Transactions on Speech and Audio*, vol. 1, pp. 68–76, January 1993.

[17] D. Klatt, "A Digital Filterbank for Spectral Matching," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 573–576, 1976.

[18] J. Holmes and N. Sedgwick, "Noise Compensation for Speech Recognition Using Probabilistic Models," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 14.12.1–14.12.4, 1986.

[19] A. Nadas, D. Nahamoo, and M. Picheny, "Speech Recognition Using Noise-Adaptive Prototypes," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 517–520, 1988.

[20] A. Varga, R. Moore, J. Bridle, K. Ponting, and M. Russell, "Noise Compensation Algorithms For Use With Hidden Markov Model Based Speech Recognition," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 481–484, 1988.

[21] A. Varga and R. Moore, "Hidden Markov Model Decomposition of Speech And Noise," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 845–848, 1990.

[22] M. Wang and S. Young, "Speech Recognition Using Hidden Markov Model Decomposition And a General Background Speech Model," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I–253–I–256, 1992.

[23] M. Gales and S. Young, "An Improved Approach To The Hidden Markov Model Decomposition of Speech And Noise," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I–233–I–236, 1992.

[24] R. Rose, "Integrated Models of Speech and Background with Application to Speaker Identification in Noise," *IEEE Transactions on Speech and Audio*, vol. 2, pp. 245–257, April 1994.

[25] Y. Ephraim, "Gain-Adapted Hidden Markov Models for Recognition of Clean and Noisy Speech," *IEEE Transactions on Singal Processing*, vol. 40, pp. 1303–1316, June 1992.

[26] N. Merhav and C.-H. Lee, “A Minimax Classification Approach with Application to Robust Speech Recognition,” *IEEE Transactions on Speech and Audio*, vol. 1, pp. 90–100, January 1993.

[27] S. Cox and J. Bridle, “Unsupervised Speaker Adaptation by Probabilistic Spectrum Fitting,” in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 294–297, 1989.

[28] S. Cox and J. Bridle, “Simultaneous Speaker Normalisation and Utterance Labelling Using Bayesian/Neural Net Techniques,” in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 161–164, 1990.

[29] Y. Zhao, “A New Speaker Adaptation Technique Using Very Short Calibration Speech,” in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II–562–II–565, 1993.

[30] M. Rahim and B.-H. Juang, “Signal Bias Removal for Robust Telephone Speech Recognition in Adverse Environments,” in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1994. to appear.

[31] A. Acero and R. Stern, “Environmental Robustness In Automatic Speech Recognition,” in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 849–852, 1990.

[32] A. Acero, *Acoustical and Environmental Robustness In Automatic Speech Recognition.* Kluwer Academic Publishers, 1992.

[33] C.-H. Lee, L. R. Rabiner, R. Pieraccini, and J. Wilpon, “Acoustic Modeling for Large Vocabulary Speech Recognition,” *Computer Speech and Language*, vol. 4, pp. 127–165, January 1990.

[34] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE*, vol. 77, pp. 257–286, February 1989.

[35] L. Bahl, F. Jelinek, and R. Mercer, “A Maximum Likelihood Approach to Continuous Speech Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, pp. 179–190, March 1983.

[36] H. Sakoe, “Two-Level DP-Matching–A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition,” *IEEE Transactions on*

Acoustics, Speech, and Signal Processing, vol. ASSP-27, pp. 588–595, December 1979.

[37] C. Myers and L. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, pp. 284–297, April 1981.

[38] C.-H. Lee and L. Rabiner, "A Frame-Synchronous Network Search Algorithm for Connected Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1649–1658, November 1989.

[39] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[40] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[41] B.-H. Juang, "Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains," *AT&T Technical Journal*, vol. 64, no. 6, pp. 1235–1249, 1985.

[42] L. R. Rabiner, J. Wilpon, and B.-H. Juang, "A Segmental K-Means Training Procedure for Connected Word Recognition," *AT&T Technical Journal*, vol. 64, pp. 21–40, May 1986.

[43] G. Forney, "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, March 1973.

[44] P. Price, W. Fisher, J. Bernstein, and D. Pallett, "A database for continuous speech recognition in a 1000-word domain," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 651–654, 1988.

[45] C.-H. Lee, E. Giachin, L. Rabiner, R. Pieraccini, and A. Rosenberg, "Improved acoustic modeling for large vocabulary continuous speech recognition," *Computer Speech and Language*, vol. 6, pp. 103–127, 1992.

[46] J. Gauvain and C.-H. Lee, "Maximum *a posteriori* Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE Transactions on Speech and Audio*, vol. 2, pp. 291–298, April 1994.

[47] C.-H. Lee, J.-L. Gauvain, R. Pieraccini, and L. Rabiner, "Large Vocabulary Speech Recognition using Subword Units," *Speech Communication*, vol. 13, pp. 263–279, 1993.

[48] C.-H. Lee and J.-L. Gauvain, "Speaker Adaptation Based on MAP Estimation of HMM Parameters," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II–558–II–561, 1993.