Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Speech Communication 42 (2004) 5–23

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Noise adaptive speech recognition based on sequential noise parameter estimation

Kaisheng Yao [a,*], Kuldip K. Paliwal [a,b], Satoshi Nakamura [a]

[a] *ATR Spoken Language Translation Research Labs, Kyoto, Japan*
[b] *School of Microelectronic Engineering, Griffith University, Brisbane, Australia*

Abstract

In this paper, a noise adaptive speech recognition approach is proposed for recognizing speech which is corrupted by additive non-stationary background noise. The approach sequentially estimates noise parameters, through which a nonlinear parametric function adapts mean vectors of acoustic models. In the estimation process, posterior probability of state sequence given observation sequence and the previously estimated noise parameter sequence is approximated by the normalized joint likelihood of active partial paths and observation sequence given the previously estimated noise parameter sequence. The Viterbi process provides the normalized joint-likelihood. The acoustic models are not required to be trained from clean speech and they can be trained from noisy speech. The approach can be applied to perform continuous speech recognition in presence of non-stationary noise. Experiments conducted on speech contaminated by simulated and real non-stationary noise show that when acoustic models are trained from clean speech, the noise adaptive speech recognition system provides improvements in word accuracy as compared to the normal noise compensation system (which assumes the noise to be stationary) in slowly time-varying noise. When the acoustic models are trained from noisy speech, the noise adaptive speech recognition system is found to be helpful to get improved performance in slowly time-varying noise over a system employing multi-conditional training.

Keywords: Noisy speech recognition; Non-stationary noise; Expectation maximization algorithm; Kullback proximal algorithm

1. Introduction

The state-of-the-art speech recognition systems work very well when they are trained and tested under similar acoustic environments. Their performance degrades drastically when there is a mismatch in training and test environments. When a speech recognizer is deployed in a real-life situation, it has to encounter environment distortions, such as channel distortion and background noise, which cause mismatch between pre-trained models and testing data. This mismatch between training and test conditions can be viewed in the signal-space, the feature-space, or the model-space (Sankar and Lee, 1996). A number of methods have been proposed in the literature to improve the robustness of a speech recognizer to overcome this mismatch problem occurring due to channel

[*] Corresponding author. Present address: Institute for Neural Computation, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0523, USA. Tel.: +1-858-822-2720; fax: +1-858-565-7440.

E-mail addresses: kyao@ucsd.edu (K. Yao), k.paliwal@me.gu.edu.au (K.K. Paliwal), satoshi.nakamura@atr.co.jp (S. Nakamura).

0167-6393/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2003.09.002

distortion and additive background noise. These robust methods can be grouped, in general, in four kinds. The first kind of methods are based on front-end signal processing, where speech enhancement techniques are used prior to feature extraction for improving the signal-to-noise ratio (SNR) (Ephraim and Malah, 1984). The second kind of methods are based on robust feature extraction; these methods try to extract features from the speech signal which remain, to some extent, invariant to environment effects, e.g., perceptual linear prediction (PLP) (Hermansky, 1990) and combination of static, dynamic and acceleration features (Hanson and Applebaum, 1990). The third kind of methods are based on missing feature theory (MFT) (Morris et al., 1998), where effect of acoustic environment (background noise) on each component of a feature vector is estimated. The components which are effected (or, corrupted) more are either discarded or given less weight (or, importance) during likelihood computation. The fourth kind of methods is model-based. These methods assume parametric models for representing environment effects on speech features. The environment effects are compensated for either by modifying the hidden Markov model (HMM) parameters in the model-space, e.g., parallel model combination (PMC) (Gales and Young, 1997) and stochastic matching (Sankar and Lee, 1996), or by modifying the input feature vectors, e.g., code-dependent cepstral normalization (CDCN) (Acero, 1990), vector Taylor series (VTS) (Moreno et al., 1996), maximum likelihood signal bias removal (SBR) (Rahim and Juang, 1996), Jacobian adaptation (Sagayama et al., 1997; Cerisara et al., 2001), and frequency-domain ML feature estimation (Zhao, 2000). The model-based methods have been shown promising for compensating noise effects (Vaseghi and Milner, 1997).

Most of the above-mentioned methods can deal with the stationary environment conditions. In this situation, noise parameters (mean vectors of a Gaussian mixture model (GMM) representing statistics of noise) are often estimated before speech recognition from a small set of environment adaptation data for modifying HMM parameter or input features. However, the environmental distortions may be non-stationary which happens most of the time when a speech recognizer is used in a real-life situation. As a result, the environment statistics may vary during recognition and the noise parameters estimated prior to speech recognition are no longer relevant to the subsequent speech signal.

In this paper, we propose a method (Yao et al., 2002) that performs speech recognition in the presence of non-stationary background noise. Noise parameters are estimated here sequentially, i.e., frame-by-frame, which allows this method to handle non-stationary noise. In addition, this method has the advantage that it does not require the acoustic models to be trained from clean speech. [1] The acoustic models can also be trained from noisy speech.

This paper is organized as follows. In Section 2 we briefly review the current methods for speech recognition in non-stationary noise. In Section 3, the model-based noisy speech recognition is reviewed and, in particular, Section 3.2 presents noise parameter estimation as a process that requires both acoustic models and noisy speech observations. The noise parameter estimation process must be carried out sequentially in order to track the time-varying noise parameter. In Section 4, the time-recursive noise parameter estimation is described. The sequential Kullback proximal algorithm (Yao et al., 2001), which is an extension of the sequential EM algorithm, is applied for sequential estimation. Compared to the sequential EM algorithm, the sequential Kullback proximal algorithm gives flexibility in controlling its convergence rate. Section 4.2 justifies the Viterbi approximation of the posterior probabilities of state sequences given observation sequences. Section 5 provides experimental results carried out on TI-Digits and Aurora 2 database (Hirsch and Pearce, 2000) to show the efficacy of the method. Discussions and conclusions are presented in Sections 6 and 7, respectively.

[1] This is different from some of the above-mentioned robust methods (e.g., PMC) which assume the acoustic models to be trained from clean speech.

1.1. Notation

Vectors are denoted by bold-faced lower-case letters and matrices are denoted by bold-faced upper-case letters. Elements of vectors and matrices are not bold-faced. Time index is in the parenthesis of vectors, matrices, or elements. Superscript T denotes transpose. Sequence is denoted by (,). Set is denoted as {,}. Sequence of vectors is denoted by bold-faced uppercased letter. For example, sequence $\boldsymbol{Y}(T) = (\boldsymbol{y}(1), \ldots, \boldsymbol{y}(T))$ consists of vector element $\boldsymbol{y}(t)$ at time t, where its ith element is $y_i(t)$. The distribution of the vector $\boldsymbol{y}(t)$ is $P(\boldsymbol{y}(t))$.

In the rest of the paper, the symbol $\boldsymbol{X}$ (or x) is exclusively used for original speech and $\boldsymbol{Y}$ (or y) is used for noisy speech in testing environments. $\boldsymbol{n}$ is used to denote noise.

In the context of speech recognition, speech model is denoted as Λ_X. The time-varying noise parameter sequence is denoted by $\boldsymbol{\Lambda}_N(T) = (\lambda_N(1), \lambda_N(2), \ldots, \lambda_N(T))$, where $\lambda_N(t)$ is the noise parameter at time t. In this work, $\lambda_N(t)$ is a time-varying mean vector $\mu_n^{\mathrm{l}}(t)$.

By default, observation (or feature) vectors are in cepstral domain. Superscript l explicitly denotes log-spectral domain. For example, speech model Λ_X is trained from speech sequence $\boldsymbol{X}(T) = (\boldsymbol{x}(1), \ldots, \boldsymbol{x}(t), \ldots, \boldsymbol{x}(T))$ in cepstral domain, and its log-spectral domain counterpart is $\boldsymbol{X}^{\mathrm{l}}(T) = (\boldsymbol{x}^{\mathrm{l}}(1), \ldots, \boldsymbol{x}^{\mathrm{l}}(t), \ldots, \boldsymbol{x}^{\mathrm{l}}(T))$.

2. Review of methods for noisy speech recognition in non-stationary noise

The model-based robust speech recognition methods use a number of techniques to combat time-varying environment effects. They can be categorized into two approaches. In the first approach, time-varying environment sources are modeled by HMMs or GMMs that are trained by prior measurement of environments, so that environment compensation is a task of identification of the underlying state sequences of the environment HMMs (Gales and Young, 1997; Varga and Moore, 1990; Takiguchi et al., 2000) by MAP estimation in a batch mode. For example, in (Gales and Young, 1997), an ergodic HMM represents different SNR conditions, so that HMMs that are compositions of speech and the ergodic environment model can have expanded states that possibly represent speech states at different SNR conditions. This approach requires a model representing different conditions of environments (SNRs, types of noise, etc.), so that statistics at some states or mixtures obtained before speech recognition are close to the real testing environments.

In the second approach, parameters of the environment models are assumed to be time varying. The parameters can be estimated based on maximum likelihood, e.g., sequential EM algorithm (Kim, 1998; Zhao et al., 2001; Afify and Siohan, 2001). In (Kim, 1998), the sequential EM algorithm is applied to estimate time-varying parameters in cepstral domain. A batch-mode noise parameter estimation method (Zhao, 2000) has been extended to sequential estimation of time-varying parameters in linear frequency domain (Zhao et al., 2001). The noise parameters can also be estimated by Bayesian methods (Frey et al., 2001; Yao et al., 2002). In (Frey et al., 2001) a Laplace transform is used to approximate the joint distribution of speech, additive noise and channel distortion by vector Taylor series approximation. In (Yao and Nakamura, 2002), sequential Monte Carlo method is used to estimate noise parameters.

The method reported in this paper belongs to the second approach using maximum likelihood estimation. A more detailed discussion about the relation of our method with the above methods is presented in Section 6.

3. Model-based noisy speech recognition

3.1. MAP decision rule for automatic speech recognition

The speech recognition problem can be described as follows. Given a set of trained models $\Lambda_X = \{\lambda_{x_m}\}$ (where λ_{x_m} is the model of mth speech unit trained from $\boldsymbol{X}$) and an observation vector sequence $\boldsymbol{Y}(T) = (\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(T))$, the aim is to recognize the word sequence $\boldsymbol{W} = (\boldsymbol{W}(1), \boldsymbol{W}(2), \ldots, \boldsymbol{W}(L))$ embedded in $\boldsymbol{Y}(T)$. Each speech unit model λ_{x_m} is a Υ-state CDHMM with state transition probability $a_{iq}(0 \leqslant a_{iq} \leqslant 1)$ and each state i is modeled by a mixture of Gaussian probability density functions $\{b_{ik}(\cdot)\}$ with parameter $\{w_{ik}, \mu_{ik}, \boldsymbol{\Sigma}_{ik}\}_{k=1,2,\dots,M}$, where M denotes the number of Gaussian mixture components in each state. $\mu_{ik} \in R^{D\times 1}$ and $\boldsymbol{\Sigma}_{ik} \in R^{D\times D}$ are the mean vector and covariance matrix, respectively, of each Gaussian mixture component. D is the dimensionality of feature space (or number of components in a feature vector). w_{ik} is the mixture weight for state i and mixture k.

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

$$\begin{aligned}\widehat{\boldsymbol{W}} &= \arg\max_{\boldsymbol{W}} P(\boldsymbol{W}|\Lambda_X, \boldsymbol{Y}\ (T)) \\ &= \arg\max_{\boldsymbol{W}} P(\boldsymbol{Y}\ (T)|\Lambda_X, \boldsymbol{W}) P_\Gamma(\boldsymbol{W}) \qquad (1)\end{aligned}$$

where the first term is the likelihood of observation sequence $\boldsymbol{Y}(T)$ given that the word sequence is $\boldsymbol{W}$, and the second term denotes the language model.

3.2. Model-based noisy speech recognition

In the model-based robust speech recognition methods, the effect of environment effects on speech feature vectors is represented in terms of a model. In particular, for MFCC features based front-end, the following function was used in (Gales and Young, 1997; Acero, 1990) to approximate additive noise effects on speech power (See Appendix A for derivation):

$$y_j^1(t) = x_j^1(t) + \log(1 + \exp(n_j^1(t) - x_j^1(t))) \qquad (2)$$

where $y_j^1(t)$ denotes the logarithm of the power of (observed) noisy speech from the jth bin of filter bank (used in MFCC analysis) at time t. Similarly, $x_j^1(t)$ and $n_j^1(t)$ denote the log-powers of clean speech and additive noise from the jth filter-bank bin at time t. J is the number of bins in the filter bank.

In order to illustrate the functional form represented by Eq. (2), we plot in Fig. 1 $y_j^1(t)$ as a function of $n_j^1(t)$ keeping $x_j^1(t)$ fixed ($x_j^1(t) = 1.0$). It can be seen from this figure that the function is smooth and convex. This function approximates the masking effects of $n_j^1(t)$ on $x_j^1(t)$. The function (2) will output either $x_j^1(t)$ or $n_j^1(t)$ depending on whether $x_j^1(t)$ is much larger than $n_j^1(t)$ or $n_j^1(t)$ is much larger than $x_j^1(t)$. When $x_j^1(t) \approx n_j^1(t)$, the observation $y_j^1(t)$ is non-linearly related to $x_j^1(t)$ and $n_j^1(t)$.

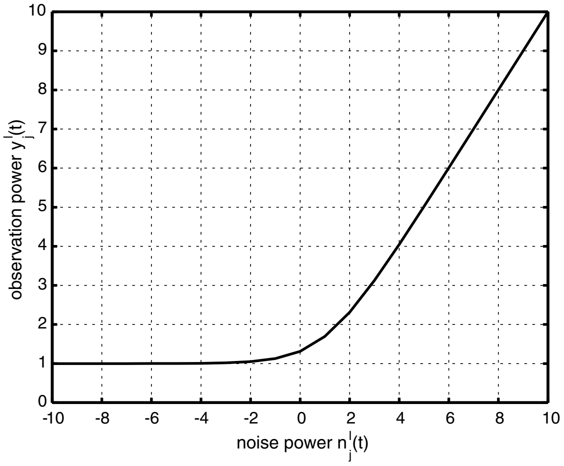

Fig. 1. Plot of function $y_j^1(t) = x_j^1(t) + \log(1 + \exp(n_j^1(t) - x_j^1(t)))$. $x_j^1(t) = 1.0$. $n_j^1(t)$ ranges from −10.0 to 10.0.

Cepstral vectors $\boldsymbol{y}(t)$, $\boldsymbol{x}(t)$ and $\boldsymbol{n}(t)$ are obtained by discrete Cosine transform (DCT) on $\boldsymbol{y}^1(t)$, $\boldsymbol{x}^1(t)$ and $\boldsymbol{n}^1(t)$, respectively. Training data of $\{\boldsymbol{x}(t) : t = 1, \dots, T\}$ is used to train the acoustic model Λ_X.

Based on certain assumptions, some of the above-mentioned robust methods use Eq. (2) to transform parameters of speech models or noisy speech features. For example, when the variances of $x_j^1(t)$ and $n_j^1(t)$ are assumed to be very small (as done in Log-Add method (Gales and Young, 1997)), a non-linear transformation on the mean vector μ_{ik}^1 in mixture k of state i in Λ_X can be derived as follows:

$$\hat{\mu}_{ik}^1 = \mu_{ik}^1 + \log(1 + \exp(\mu_n^1 - \mu_{ik}^1)) \qquad (3)$$

where $\mu_n^1 \in R^{J\times 1}$ is a mean vector for modeling statistics of the noise data $\{\boldsymbol{n}^1(t) : t = 1, \dots, T\}$. We denote the parameters of the noise model, e.g., mean vector and variance of a GMM, of the noise $\{\boldsymbol{n}(t) : t = 1, \dots, T\}$ by $\boldsymbol{\Lambda}_N$.

With the estimated $\boldsymbol{\Lambda}_N$ and certain transformation function (e.g., Eq. (3)), Eq. (1) can be carried out as

$$\widehat{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}} P(\boldsymbol{Y}\ (T)|\Lambda_X, \boldsymbol{\Lambda}_N, \boldsymbol{W}) P_\Gamma(\boldsymbol{W}) \quad (4)$$

This function defines the model-based noisy speech recognition approach in our paper. Note that the likelihood is obtained here given speech model Λ_X, word sequences $\boldsymbol{W}$, and $\boldsymbol{\Lambda}_N$. Compared to Eq. (1), this approach has an extra requirement on estimation of $\boldsymbol{\Lambda}_N$.

3.2.1. Noise compensation by the model-based noisy speech recognition

In practice, we may encounter the situation that $x_j^1(t)$ for training speech models is noisy. Therefore, in order to apply function (2) for model-based noisy speech recognition, it is necessary to consider two situations: one when $x_j^1(t)$ comes from clean speech and the other when it comes from noisy speech.

In the first situation where $x_j^1(t)$ is extracted from clean speech, as shown in Appendix A, function (2) provides physical meaning of $n_j^1(t)$, which denotes noise power at jth filter bank at time t in the log-spectral domain. Assuming that the statistics do not change during recognition process, a model of the statistics of $\{n_j^1(t) : t = 1, \ldots, T\}$ can be estimated from noise along segments. For example, μ_n^1 in Eq. (3) can be estimated as the mean vector of $\boldsymbol{n}^1(t) : t = 1, \ldots, T\}$. This assumption is explored in other methods, e.g., PMC (Gales and Young, 1997).

In the second situation, $x_j^1(t)$ in function (2) is extracted from noisy speech. One way to apply function (2) to this situation is to decompose it by Taylor series as that in Jacobian adaptation (Sagayama et al., 1997; Cerisara et al., 2001). Another way, which is adopted in this paper, treats function (2) as a non-linear regression function between $x_j^1(t)$ and $y_j^1(t)$. In this context, $n_j^1(t)$ is the parameter for non-linear regression between $x_j^1(t)$ and $y_j^1(t)$; i.e., $n_j^1(t)$ is a function of $x_j^1(t)$ and $y_j^1(t)$. To illustrate the idea, the function (2) can be manipulated to derive the relation $n_j^1(t) = x_j^1(t) + \log(\exp(y_j^1(t) - x_j^1(t)) - 1)$. Although this relation is not directly utilized in this paper, it shows that estimation of $n_j^1(t)$ requires both $x_j^1(t)$ and $y_j^1(t)$. Since it is the parameter of $n_j^1(t)$, $\boldsymbol{\Lambda}_N$, that is used in the model-based noisy speech recognition approach, $\boldsymbol{\Lambda}_N$ is estimated given sequences of $x_j^1(t)$ and $y_j^1(t)$.

Thus, in the present paper, we perform noise compensation as a process conducted (iteratively) in two steps: noise parameter estimation step and acoustic model (or feature) adaptation step.

In the noise parameter estimation step, $\boldsymbol{\Lambda}_N$ (parameterizing $n_j^1(t)$) is estimated as the parameter for the non-linear regression between the sequences of $y_j^1(t)$ and $x_j^1(t)$, via certain criterion, e.g., maximum likelihood estimation of $\boldsymbol{\Lambda}_N$ given the sequences. In the acoustic model (or feature) adaptation step, $\boldsymbol{\Lambda}_N$ is substituted back into functional formula, e.g., Eq. (3), which is derived based on the non-linear regression function (2), to transform speech model Λ_X in the model space, so that the transformed model $\widehat{\Lambda}_Y$ is close to $\{\boldsymbol{y}(t) : t = 1, \ldots, T\}$. Similarly, the transformation can be carried out in the feature space to make $\{\boldsymbol{y}(t) : t = 1, \ldots, T\}$ close to Λ_X.

One point that needs to be clarified is that, as shown in Fig. 1, when estimating parameter of $n_j^1(t)$ as a non-linear regression between $x_j^1(t)$ and $y_j^1(t)$, the non-linearity of the function (2) may result in an estimate that is different from the true parameters of the additive noise even though $x_j^1(t)$ is clean speech. In view of this, it is better to see the estimate as parameter in the non-linear function of (2), instead of explicit meaning of the noise parameter. For the consistency in notation with other methods (Gales and Young, 1997), in the sequel, we still refer $\boldsymbol{\Lambda}_N$, the estimated parameter for $\{\boldsymbol{n}^1(t) : t = 1, \ldots, T\}$, as noise parameter.

Normally, a direct observation of $x_j^1(t)$ is not available, so $\boldsymbol{\Lambda}_N$ are estimated from Λ_X (the model of $x_j^1(t)$), and sequences of $y_j^1(t)$ in either a supervised (with correct transcript) or unsupervised (correct transcript is not known) way.

4. Noise adaptive speech recognition

As mentioned earlier, we consider the case when noise conditions change during the recognition process. Therefore, Λ_N (in (4)) has to be estimated sequentially, i.e., frame-by-frame.

We propose here a noise adaptive speech recognition algorithm that carries out sequential estimate of time-varying noise parameter for noisy speech recognition. This algorithm works in the model space; i.e., modifying HMM parameters, and is shown in Fig. 2. At each frame t, the noise adaptive speech recognition carries out noise parameter estimation in the module "Noise parameter estimation" according to objective functions in Section 4.1. With the estimated noise parameter at the current frame, the module "Acoustic model adaptation" adapts mean vectors of the acoustic model Λ_X by a non-linear function (14). The adapted acoustic model, $\widehat{\Lambda}_Y(t)$, is fed into the recognition process, which updates the approximation of the posterior probabilities of state sequences via a Viterbi process, and the approximated posterior probabilities are used in the module "Noise parameter estimation" to update the noise parameter sequence in the next frame. Detailed description of this algorithm is provided in the following sections.

In Section 4.1, the objective function for time-varying noise parameter estimation is defined. The Viterbi approximation of the posterior distribution of state sequences given noisy observation sequences is described in Section 4.2. Section 4.3 provides the detailed implementation.

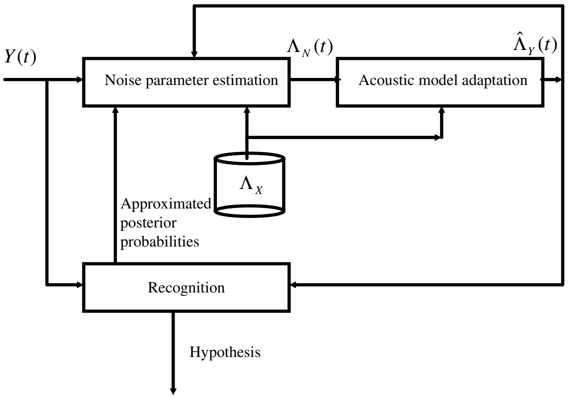
$Y(t)$
$\Lambda_N(t)$
$\hat{\Lambda}_Y(t)$
Noise parameter estimation
Acoustic model adaptation
Λ_X
Approximated posterior probabilities
Recognition
Hypothesis

Fig. 2. Diagram of the noise adaptive speech recognition. Λ_X, $\boldsymbol{\Lambda}_N(t)$ and $\widehat{\Lambda}_Y(t)$ are the original acoustic model, noise parameter sequence at frame t, and adapted acoustic model at frame t, respectively. $\boldsymbol{Y}$ (t) is the input noisy speech observation sequence till frame t. Recognition module provides approximated posterior probabilities of state sequences given noisy observation sequences till frame t to the noise parameter estimation module, which output $\boldsymbol{\Lambda}_N(t)$ to adapt acoustic model Λ_X to $\widehat{\Lambda}_Y(t)$.

4.1. Objective function for time-varying noise parameter estimation

Denote the estimated noise parameter sequence till frame $t-1$ as $\boldsymbol{\Lambda}_N(t-1) = (\hat{\lambda}_N(1), \hat{\lambda}_N(2), \ldots, \hat{\lambda}_N(t-1))$, where $\hat{\lambda}_N(t-1)$ is the parameter estimated in the previous frame. Given the current observation sequence $\boldsymbol{Y}(t) = (\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(t))$ till frame t, the noise parameter estimation procedure will find $\hat{\lambda}_N(t)$ as the current noise parameter estimate, which satisfies

$$l_t(\hat{\lambda}_N(t)) \geqslant l_t(\hat{\lambda}_N(t-1)) \tag{5}$$

where $l_t(\hat{\lambda}_N(t))$ is the log-likelihood of observation sequence $\boldsymbol{Y}(t)$ given speech model Λ_X and noise parameter sequence $(\Lambda_N(t-1), \hat{\lambda}_N(t))$; i.e.,

$$\begin{aligned} l_t(\hat{\lambda}_N(t)) &= \log P(\boldsymbol{Y}(t)|\Lambda_X, (\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t))) \\ &= \log \sum_{S(t)} P(\boldsymbol{Y}(t), S(t)|\Lambda_X, \\ &\quad (\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t))) \end{aligned} \tag{6}$$

and $l_t(\hat{\lambda}_N(t-1))$ is the log-likelihood of observation sequence $\boldsymbol{Y}(t)$ given speech model Λ_X and noise parameter sequence $(\Lambda_N(t-1), \hat{\lambda}_N(t-1))$; i.e.,

$$\begin{aligned} l_t(\hat{\lambda}_N(t-1)) &= \log P(\boldsymbol{Y}(t)|\Lambda_X, \\ &\quad (\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t-1))) \\ &= \log \sum_{S(t)} P(\boldsymbol{Y}(t), S(t)|\Lambda_X, \\ &\quad (\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t-1))) \end{aligned} \tag{7}$$

Here $S(t) = (s(1), s(2), \ldots, s(t))$ is the state sequence till frame t.

Eq. (5) shows that the updated noise parameter sequence $(\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t))$ will not decrease the likelihood of observation sequence $\boldsymbol{Y}(t)$, over that given by the previous estimate of the noise parameter $\hat{\lambda}_N(t-1)$ concatenated with the previously estimated noise parameter sequence $\boldsymbol{\Lambda}_N(t-1)$.

Since $S(t)$ is hidden, at each frame, we iteratively maximize the lower bound of the log-likelihood according to Jensen's inequality; i.e.,

$$
\begin{aligned}
&\log P(\boldsymbol{Y}(t)|\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t)))\\
&\quad =\log\sum_{S(t)}P(\boldsymbol{Y}(t),S(t)|\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t)))\\
&\quad \geqslant \sum_{S(t)}P(S(t)|\boldsymbol{Y}(t),\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))\\
&\qquad \times\log\frac{P(\boldsymbol{Y}(t),S(t)|\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t)))}{P(S(t)|\boldsymbol{Y}(t),\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))}\\
&\quad =\sum_{S(t)}P(S(t)|\boldsymbol{Y}(t),\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))\\
&\qquad \times\log\{P(\boldsymbol{Y}(t),S(t)|\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t)))\}+Z
\end{aligned}
\tag{8}
$$

where $\lambda_N^\star(t)$ is an auxiliary variable, and Z is not a function of $\hat{\lambda}_N(t)$.

Define auxiliary function as

$$
\begin{aligned}
&Q_t(\lambda_N^\star(t);\hat{\lambda}_N(t))\\
&\quad =\sum_{S(t)}P(S(t)|\boldsymbol{Y}(t),\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))\\
&\qquad \times\log\{P(\boldsymbol{Y}(t),S(t)|\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t)))\}
\end{aligned}
\tag{9}
$$

It provides the objective function to be maximized by sequential EM algorithm (Krishnamurthy and Moore, 1993).

The algorithm is carried out by iterations between the procedure to calculate the posterior probability $P(S(t)|\boldsymbol{Y}(t),\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))$, and maximization of the objective function to obtain $\hat{\lambda}_N(t)$. For each iteration, estimated $\hat{\lambda}_N(t-1)$ is for initialization of $\lambda_N^\star(t)$ in the next iteration.

Forgetting factor $\rho(0<\rho\leqslant 1.0)$ can be adopted to improve convergence rate by reducing the effects of past observations relative to the new input, so that the auxiliary function is modified to (Krishnamurthy and Moore, 1993)

$$
\begin{aligned}
&Q_t(\lambda_N^\star(t);\hat{\lambda}_N(t))\\
&\quad =\sum_{\tau=1}^{t}\rho^{t-\tau}\sum_{s(\tau)}P(s(\tau)|\boldsymbol{Y}(\tau),\Lambda_X,(\boldsymbol{\Lambda}_N(\tau-1),\lambda_N^\star(\tau)))\\
&\qquad \times\log\{P(\boldsymbol{Y}(\tau),s(\tau)|\Lambda_X,(\boldsymbol{\Lambda}_N(\tau-1),\hat{\lambda}_N(\tau)))\}
\end{aligned}
\tag{10}
$$

In this above summation, the posterior probability at state $s(\tau)$ is weighted by a factor $\rho^{t-\tau}$, which is diminishing to smaller value when $t-\tau$ gets larger.

The objective function by sequential Kullback proximal algorithm (Yao et al., 2001) is obtained by adding a Kullback–Leibler (K–L) divergence, $I(\hat{\lambda}_N(t-1);\hat{\lambda}_N(t))$, between $P(S(t)|\boldsymbol{Y}(t),\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t-1)))$ and $P(S(t)|\boldsymbol{Y}(t),\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t)))$ into the above objective functions. So the new objective function is given by,

$$
Q_t(\lambda_N^\star(t);\hat{\lambda}_N(t))-(\beta_t-1)I(\hat{\lambda}_N(t-1);\hat{\lambda}_N(t)) \tag{11}
$$

where $\beta_t\in R^+$ works as a relaxation factor, and the K–L divergence is

$$
\begin{aligned}
&I(\hat{\lambda}_N(t-1);\hat{\lambda}_N(t))\\
&\quad =\sum_{S(t)}P(S(t)|\boldsymbol{Y}(t),\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t-1)))\\
&\qquad \times\log\frac{P(S(t)|\boldsymbol{Y}(t),\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t-1)))}{P(S(t)|\boldsymbol{Y}(t),\Lambda_X,(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t)))}
\end{aligned}
\tag{12}
$$

The sequential EM algorithm is a special case of this algorithm and corresponds to setting β_t equal to 1.0 in the algorithm (proofs are shown in Appendix C.). Sequential Kullback proximal algorithm can achieve faster parameter estimation than that by sequential EM algorithm (Yao et al., 2001).

4.2. Approximation of the posterior probability

Normally, time-varying environment parameter estimation is carried out separately from the recognition process, as that in (Kim, 1998; Zhao et al., 2001), by sequential EM algorithm with summation over all state/mixture sequences of a separately trained acoustic model. In fact, the joint likelihood of observation sequence $\boldsymbol{Y}(t)$ and state sequence $S(t)$ can be approximately obtained from the Viterbi process, i.e.,

$$
\begin{aligned}
&P(\boldsymbol{Y}(t),S(t)|\Lambda_X,\boldsymbol{\Lambda}_N(t))\approx a_{s^\star(t-1)s(t)}b_{s(t)}(\boldsymbol{y}(t))\\
&P(\boldsymbol{Y}(t-1),S^\star(t-1)|\Lambda_X,\boldsymbol{\Lambda}_N(t-1))
\end{aligned}
\tag{13}
$$

where the previous state $s^\star(t-1)$ for decision of $S^\star(t-1)$ is given as,

$$s^{\star}(t-1) = \arg\max\nolimits_{s(t-1)} a_{s(t-1)s(t)} \cdot P(\boldsymbol{Y}\ (t-1), \\ S(t-1)|\Lambda_X, \boldsymbol{\Lambda}_N(t-1))$$

By normalizing the joint likelihood with respect to the sum of those from all active partial state sequences in the recognition stage, an approximation of the posterior probability of state sequence can be obtained. Thus in (9) and (12), instead of summing over all state/mixture sequences, the summation is over all *active partial state sequence* (path) till frame t provided by Viterbi process. By Jensen's inequality (8), the summation still provides the lower bound of the log-likelihood. This approximation makes it easy to combine time-varying environment parameter estimation with the Viterbi process. We thus denote this scheme of time-varying environment parameter estimation as noise adaptive speech recognition since the same Viterbi process is shared by the recognition process and the time-varying noise parameter estimation process.

4.3. Implementation

Time-varying noise parameter estimation is carried out in the log-spectral domain. In particular, the mean vector $\mu_n^{\rm l}$ in Eq. (3) is treated as time-varying noise parameter. Thus, Eq. (3) is written as,

$$\hat{\mu}_{ik}^{\rm l}(t) = \mu_{ik}^{\rm l} + \log(1 + \exp(\mu_n^{\rm l}(t) - \mu_{ik}^{\rm l})) \tag{14}$$

By DCT on the above transformed mean vector, cepstral mean vector $\hat{\mu}_{ik}(t) \in R^{D\times 1}$ of the adapted model $\widehat{\Lambda}_Y(t)$ is obtained.

By (14), the likelihood density function is related to noise parameters as that shown in (4). The log-likelihood density function for mixture k in state i is given by

$$\log b_{ik}(\boldsymbol{y}(t)) = -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_{ik}| \\ -\frac{1}{2}(\boldsymbol{y}(t) - \hat{\mu}_{ik}(t))^{\rm T}\boldsymbol{\Sigma}_{ik}^{-1}(\boldsymbol{y}(t) - \hat{\mu}_{ik}(t)) \tag{15}$$

Initialization of the noise parameter is $\hat{\lambda}_N(0)$. $\lambda_N(t)$ is estimated by the sequential Kullback proximal algorithm (derivation is in Appendix D).

Time-varying parameter estimation by the sequential Kullback proximal algorithm is carried out as follows. Given $\boldsymbol{Y}(t)$, the recursive update of $\hat{\lambda}_N(t)$ is given as,

$$\hat{\lambda}_N(t) \leftarrow \hat{\lambda}_N(t-1) \\ - \left.\frac{\frac{\partial Q_t(\hat{\lambda}_N(t-1);\hat{\lambda}_N)}{\partial\hat{\lambda}_N}}{\beta_t\frac{\partial^2 Q_t(\hat{\lambda}_N(t-1);\hat{\lambda}_N)}{\partial\hat{\lambda}_N^2} + (1-\beta_t)\frac{\partial^2 l_t(\hat{\lambda}_N)}{\partial\hat{\lambda}_N^2}}\right|_{\hat{\lambda}_N=\hat{\lambda}_N(t-1)} \tag{16}$$

where $Q_t(\hat{\lambda}_N(t-1);\hat{\lambda}_N)$ is the auxiliary function for the parameter estimation. Its first- and second-order derivative of the auxiliary function with respect to the noise parameter are respectively given as,

$$\frac{\partial Q_t(\hat{\lambda}_N(t-1);\hat{\lambda}_N)}{\partial\hat{\lambda}_N} \\ = \sum_{s(t)}\sum_{k(t)} P(s(t)k(t)|\boldsymbol{Y}(t), \Lambda_X, (\boldsymbol{\Lambda}_N(t-1), \\ \hat{\lambda}_N(t-1)))\frac{\partial \log b_{s(t)k(t)}(\boldsymbol{y}(t))}{\partial\hat{\lambda}_N} \tag{17}$$

$$\frac{\partial^2 Q_t(\hat{\lambda}_N(t-1);\hat{\lambda}_N)}{\partial\hat{\lambda}_N^2} \\ = \rho\cdot\frac{\partial^2 Q_{t-1}(\hat{\lambda}_N(t-2);\hat{\lambda}_N)}{\partial\hat{\lambda}_N^2} \\ + \sum_{s(t)}\sum_{k(t)} P(s(t)k(t)|\boldsymbol{Y}(t), \Lambda_X, (\boldsymbol{\Lambda}_N(t-1), \\ \hat{\lambda}_N(t-1)))\frac{\partial^2 \log b_{s(t)k(t)}(\boldsymbol{y}(t))}{\partial\hat{\lambda}_N^2} \tag{18}$$

The second-order derivative of the log-likelihood $l_t(\hat{\lambda}_N)$ with respect to the noise parameter is

$$\frac{\partial^2 l_t(\hat{\lambda}_N)}{\partial\hat{\lambda}_N^2} = \sum_{s(t)}\sum_{k(t)} P(s(t)k(t)|\boldsymbol{Y}(t), \Lambda_X, (\boldsymbol{\Lambda}_N(t-1), \\ \hat{\lambda}_N(t-1)))\left[\left(\frac{\partial \log b_{s(t)k(t)}(\boldsymbol{y}(t))}{\partial\hat{\lambda}_N}\right)^2 \\ + \frac{\partial^2 \log b_{s(t)k(t)}(\boldsymbol{y}(t))}{\partial\hat{\lambda}_N^2}\right] \\ - \left(\frac{\partial Q_t(\hat{\lambda}_N(t-1);\hat{\lambda}_N)}{\partial\hat{\lambda}_N}\right)^2 \tag{19}$$

Eqs. (16)–(19) are general formulae of the sequential Kullback proximal algorithm, which are applicable to sequential parameter estimation beyond the current work. Note that, in the above formula, in addition to the posterior probabilities of state sequences $P(S(t)|\boldsymbol{Y}(t), \Lambda_X, (\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t-1)))$, only the first- and second-order derivative of the log-likelihood with respect to the parameter in interests, i.e.,

$$\frac{\partial \log b_{s(t)k(t)}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N} \quad \text{and} \quad \frac{\partial^2 \log b_{s(t)k(t)}(\boldsymbol{y}(t))^2}{\partial \hat{\lambda}_N}$$

are required to be specified. In the context of the sequential noise parameter estimation in this work, by (15), they are respectively given as,

$$\frac{\partial \log b_{s(t)k(t)}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N} = \boldsymbol{G}_{\hat{\lambda}_N} \frac{\partial \hat{\mu}^1_{s(t)k(t)}(t)}{\partial \hat{\lambda}_N} \tag{20}$$

$$\begin{aligned}\frac{\partial^2 \log b_{s(t)k(t)}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N^2} &= \boldsymbol{H}_{\hat{\lambda}_N} \left(\frac{\partial \hat{\mu}^1_{s(t)k(t)}(t)}{\partial \hat{\lambda}_N} \right)^2 \\ &\quad + \boldsymbol{G}_{\hat{\lambda}_N} \frac{\partial^2 \hat{\mu}^1_{s(t)k(t)}(t)}{\partial \hat{\lambda}_N^2}\end{aligned} \tag{21}$$

where the jjth element in diagonal matrices $\boldsymbol{G}_{\hat{\lambda}_N} \in R^{J \times J}$ and $\boldsymbol{H}_{\hat{\lambda}_N} \in R^{J \times J}$ are given as

$$G_{\hat{\lambda}_N jj} = \sum_{d=1}^{D} \left[z_{dj} \frac{(y_t(d) - \hat{\mu}_{s(t)k(t)d}(t-1))}{\Sigma^2_{s(t)k(t)d}} \right] \quad \text{and}$$

$$H_{\hat{\lambda}_N jj} = \sum_{d=1}^{D} \left[-\frac{1}{\Sigma^2_{s(t)k(t)d}} z^2_{dj} \right]$$

respectively. z_{dj} is the DCT coefficient.

The posterior probability, $P(s(t)k(t)|\boldsymbol{Y}(t), \Lambda_X, (\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t-1)))$, at state $s(t)$ and mixture $k(t)$ given observation sequence $\boldsymbol{Y}(t)$ and noise parameter sequence $(\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t-1))$ is approximated by Viterbi process as described in Section 4.2.

Since $\hat{\lambda}_N(t)$ represents the time-varying noise mean vector $\hat{\mu}^1_n(t)$, by (14), the first- and second-order derivatives of $\hat{\mu}^1_{s(t)k(t)}(t)$ with respect to the noise parameter $\mu^1_{nj}(t)$ in Eqs. (20) and (21) are respectively given as,

$$\frac{\partial \hat{\mu}^1_{s(t)k(t)}(t)}{\partial \mu^1_{nj}(t)} = \frac{\exp(\mu^1_{nj}(t) - \mu^1_{s(t)k(t)j})}{1 + \exp(\mu^1_{nj}(t) - \mu^1_{s(t)k(t)j})} \tag{22}$$

$$\frac{\partial^2 \hat{\mu}^1_{s(t)k(t)}(t)}{\partial \mu^1_{nj}(t)^2} = \frac{\exp(\mu^1_{nj}(t) - \mu^1_{s(t)k(t)j})}{(1 + \exp(\mu^1_{nj}(t) - \mu^1_{s(t)k(t)j}))^2} \tag{23}$$

Plugging the above estimates, respectively, into $\frac{\partial \hat{\mu}^1_{s(t)k(t)}(t)}{\partial \hat{\lambda}_N}$ and $\frac{\partial^2 \hat{\mu}^1_{s(t)k(t)}(t)}{\partial \hat{\lambda}_N^2}$ can update noise parameter $\hat{\mu}^1_{nj}(t)$ by Eqs. (16)–(19).

5. Experiments

5.1. Experiments on acoustic models trained from clean speech

5.1.1. Experiment setup

In this set of experiments, acoustic models were trained from clean speech. Because, in this situation, some model-based noise compensation methods can be applied, we thus compared three systems. The first was the baseline without noise compensation, denoted as Baseline, and the second was the system with noise compensation based on Eq. (3) while using a stationary noise assumption (henceforth denoted as SNA); i.e., μ^1_n was kept as constant once estimated from noise along segments. The third was the noise adaptive recognition system as given by (16). This system was studied for different values of the relaxation factor β_t. Forgetting factor ρ in (10) and (18) was set to 0.995 empirically. Recognition performances of systems were measured by their word accuracies (WAs) calculated by HTK (Young, 1997) (insertion errors are counted.). These systems were compared in the view of the averaged relative error rate reduction (AERR) in noise, which is calculated as the average of the relative error rate reductions (ERR) in the noise. For example, the ERR of system 2 over system 1 is calculated by

$$\text{ERR} = \frac{\text{WA2} - \text{WA1}}{100\% - \text{WA1}} \tag{24}$$

where WA1 and WA2 each denote the WA achieved by system 1 and system 2.

Experiments were performed on TI-Digits database down-sampled to 16 kHz. Five hundred clean speech utterances from 15 speakers were used for training and 111 utterances unseen in the training set were used for testing.

Each of the 11 digits was represented by a whole word HMM with 10 states and each state modeled by four diagonal Gaussian mixtures. Silence was modeled by a 3-state HMM with four diagonal Gaussian mixtures for each state. The beginning- and ending-state of HMMs were skip-state; i.e., without output densities. The window size was 25 ms with a 10 ms shift. A filter-bank with 26 filters was used in the binning stage.

Four seconds of contaminating noise was used in each experiment to obtain noise mean vector for SNA. It was also used for initialization of $\mu_n^l(0)$ in Eq. (16) in the noise adaptive system. Baseline performance in clean condition was 97.89% word accuracy (WA). Though we could increase amount of training and testing data in the experiments, our main objective was to verify, when acoustic models were trained from clean speech, whether the sequential parameter estimation method can track background noise and whether tracking of the noise evolution can improve system robustness in terms of speech recognition performance.

5.1.2. Speech recognition in simulated non-stationary noise

In this section, we report speech recognition results on noisy speech generated from clean speech by computer-generated simulated non-stationary noise. In order to generate non-stationary noise, we used white noise signal obtained through a Gaussian random number generator and multiplied it by a chirp signal (of fixed shape) in time domain. To illustrate the shape of this chirp signal, we analyze the noise by the filter bank and plot the noise power in the 12th bin of the filter bank as a function of time in Fig. 3 (shown as the dash-dotted curve). From this figure, it can be seen that the noise power changes at an accelerating rate, which may have different value among speech utterances and within an speech utterance. The SNR of noisy speech as a result ranged from 0 to 20.4 dB. In contrast to the assumption of stationary noise, the introduced non-stationarity in the simulated noise signal is significant.

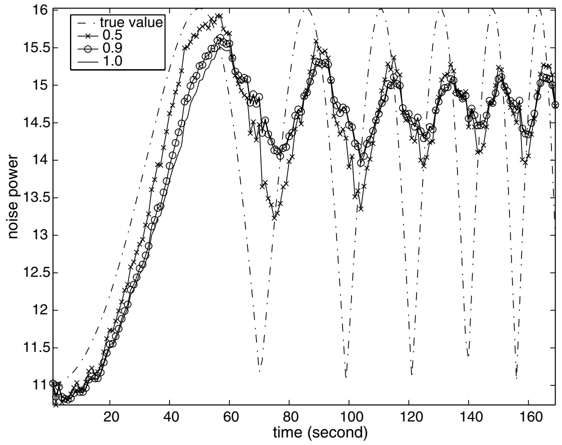

Fig. 3. Estimation of the time-varying parameter $\mu_n^l(t)$ by the noise adaptive systems in the 12th bin of the filter bank. Estimates are labeled according to the relaxation factor β_t. The dash-dotted curve shows evolution of the true noise power in the same filter-bank bin.

We also plot in Fig. 3 the noise power in the 12th bin of the filter bank estimated by the noise adaptive system. We can make the following observations from this figure. First, the noise adaptive system can track the evolution of the true noise power. Second, the results show that the smaller is the relaxation factor β_t, the faster is the convergence rate of the estimation process. For example, estimation with $\beta_t = 0.5$ shows much better tracking performance than that with $\beta_t = 1.0$.

Speech recognition performance of the noise adaptive system (measured in terms of word accuracy) is studied here for different values of β_t. The results are listed in Table 1. For comparison, the word recognition accuracies from the Baseline and SNA systems are also given in this table. It can be seen from this table that the noise adaptive system achieves significant improvement in recognition performance over the Baseline and SNA systems.

5.1.3. Speech recognition in real noise

Here, noisy speech at different SNRs is produced by adding an appropriate amount of Babble

Table 1
Word accuracy (in %) in simulated non-stationary noise achieved by the noise adaptive system as a function of β_t in comparison with Baseline (without noise compensation) and SNA (noise compensation assuming stationary noise) systems

Baseline	SNA	0.5	0.9	1.0
34.3	58.7	95.5	95.5	95.5

Because of the simulated non-stationary noise, SNR ranged from 0 to 20.4 dB.

Table 2
Word accuracy (in %) in Babble noise achieved by the noise adaptive system as a function of β_t in comparison with Baseline (without noise compensation) and SNA (noise compensation assuming stationary noise) systems

SNR (dB)	Baseline	SNA	0.5	0.9	1.0
29.5	96.7	96.7	97.6	97.9	97.9
21.5	34.0	95.2	96.4	96.7	96.7
13.6	25.3	83.1	91.0	91.3	91.3
7.6	16.3	73.2	75.6	75.3	75.3
AERR (in %)			26.9	30.9	30.9

Averaged relative error rate reduction (AERR) with respect to the SNA system is shown as a function of β_t in the last row.

noise to clean speech signals. The noise adaptive system is applied to this noisy speech and the recognition results for different values of β_t are listed in Table 2. Also shown in this table are the results from the Baseline and SNA systems for comparison. From this table, it can be observed that, in all SNR conditions, the noise adaptive system provides better performance than the SNA and Baseline systems. For example, at 21.5 dB SNR, the Baseline system achieved 34.0% WA and the SNA system attained 95.2%. The noise adaptive system with $\beta_t = 1.0$ achieved 96.7% WA. As a whole, the adaptive system with β_t set to 0.5, 0.9, and 1.0, achieved, respectively, 26.9%, 30.9%, and 30.9% averaged relative error rate reduction (AERR) with respect to the SNA system.

Though the noise adaptive system improved the recognition performance with respect to the SNA and Baseline systems for this Babble noise case, this improvement is not as significant as obtained in Section 5.1.2 for the simulated noise case. The reason for this is that amount of non-stationarity in the Babble noise is less than that in the simulated noise used in Section 5.1.2. We then increased the non-stationarity of the Babble noise by multiplying the noise signal with the same chirp signal as used in Section 5.1.2. Results are shown in Table 3. It can be observed that the averaged relative error rate reductions (AERRs) of the noise adaptive system are larger than those in Table 2.

We also tested systems in highly non-stationary Machine-gun noise. Through results shown in

Table 3
Word accuracy (in %) in the chirp-signal-multiplied Babble noise achieved by the noise adaptive system as a function of β_t in comparison with Baseline and SNA systems

SNR (dB)	Baseline	SNA	0.5	0.9	1.0
12.4	28.3	64.1	93.1	92.8	92.2
6.9	17.2	50.0	82.8	82.2	81.9
4.4	16.9	48.5	74.1	72.0	71.7
−1.6	14.8	37.7	47.6	50.0	51.5
AERR (in %)			53.0	52.4	52.3

Averaged relative error rate reduction (AERR) with respect to the SNA system is shown as a function of β_t in the last row.

Table 4
Word accuracy (in %) in Machine-gun noise, achieved by the noise adaptive system as a function of β_t in comparison with baseline without noise compensation (Baseline), and noise compensation assuming stationary noise (SNA)

SNR (dB)	Baseline	SNA	0.5	0.9	1.0
33.3	91.9	93.4	96.7	95.5	97.6
28.8	88.0	90.6	94.3	95.2	94.3
22.8	78.6	81.3	87.1	83.4	82.8
20.9	77.4	79.8	83.7	85.2	76.5
AERR (in %)			34.8	29.7	23.6

Averaged relative error rate reduction (AERR) with respect to the SNA system is shown as a function of β_t in the last row.

Table 4, we can observe that the noise adaptive system can improve recognition performance in the noise when the SNRs are within a certain range; i.e., above 20.9 dB SNR. [2]

Results presented in Fig. 3, Tables 1–3 show that the noise adaptive speech recognition performs well in slowly time-varying noise, e.g., Babble noise.

5.2. Experiments on acoustic models trained from multi-conditional data

5.2.1. Experiment setup

This set of experiments is conducted on Aurora-2 database (Hirsch and Pearce, 2000), which is a modified database from TI-Digits database. Training utterances in the experiments include 8840 utterances containing Subway, Babble, Car and Exhibition hall noise in five different SNR conditions (from 5 dB to clean condition in 5 dB steps). The test set contains noisy utterances where the noise types were the same as in the training set. In each noise, there are 1001 utterances in each SNR condition for SNRs ranging from 0 to 20 dB with 5 dB steps.

Since some model-based noise compensation methods, e.g., PMC (Gales and Young, 1997), require the acoustic models to be trained from clean speech, they cannot be applied to the experiments. A normal way for environment robustness is the multi-conditional training; i.e., the acoustic models are trained from noisy speech utterances in all sorts of noise that are the same in testing environments, which is in fact the approach carried out by the baseline in this paper.

We thus compare two systems in this set of experiments, the noise adaptive speech recognition system (denoted as Adaptive) and the baseline with multi-conditional training (denoted as Baseline).

Features were MFCC + C0 and their first-order derivatives. The feature dimension was 26. Though it was possible to improve performance by increasing the feature dimension, or state and mixture numbers, our major objective was to verify if the noise adaptive speech recognition can yield a gain over the multi-condition training system.

The noise adaptive speech recognition system was set with relaxation factor $\beta_t = 0.9$ and forgetting factor $\rho = 0.995$. At time $t = 0$, $\hat{\lambda}_N(t-1)$ was set to zero vector in order to initialize the parameter estimation by Eq. (16). Performances of systems were measured by WA. The ERR, calculated by Eq. (24), was used to compare system performances in each noise condition and the system performances as a whole were compared as the average of the ERRs (AERRs) between systems.

5.2.2. Experimental results

The recognition performances of the Adaptive and Baseline are shown in Table 5. We can observe that the noise adaptive speech recognition system has better performance than the Baseline system for Subway and Babble noise. In terms of AERR for each noise, the Adaptive system achieved 31.4% and 38.7% AERR with respect to the

[2] Prior information of the contaminating noise can be used as described in our work in (Yao et al., 2002) which formulates noise parameter estimation within the Bayesian framework, so that improvements over the SNA system could be observed in lower SNR conditions.

Table 5
Word accuracy (in %) in the Aurora-2 database, achieved by the noise adaptive speech recognition (denoted as Adaptive) with relaxation factor $\beta_t = 0.9$ and forgetting factor $\rho = 0.995$, in comparison with baseline without noise adaptive speech recognition (denoted as Baseline)

SNR (dB)	Subway		Babble		Car		Exhibit	
	Adaptive	Baseline	Adaptive	Baseline	Adaptive	Baseline	Adaptive	Baseline
20.0	88.0	84.7	92.8	86.1	92.8	92.7	90.4	90.6
15.0	87.0	78.1	89.7	80.6	91.0	90.9	87.0	87.6
10.0	82.8	70.6	84.6	72.5	87.0	87.1	82.9	83.8
5.0	76.1	63.1	75.8	61.8	76.5	75.2	75.0	74.5
0.0	62.1	53.2	58.7	49.6	52.5	53.4	61.6	57.2
AERR (in %)	31.4		38.7		1.0		0.0	

Both of the acoustic models were trained from multi-conditional training set. Average relative error rate reductions (AERR) with respect to the Baseline in each noise are in the last row.

Baseline system for Subway and Babble noise, respectively. It performs as well as the Baseline system for Car and Exhibit noise.

We can make two observations based on the results. First, the noise adaptive speech recognition system has large differences in terms of AERRs when the results for Car and Exhibit noise are compared to those for Subway and Babble noise. For example, whereas there is 31.4% AERR over the Baseline system in Subway noise, the Adaptive system only attains 1.0% AERR over the Baseline system in Car noise. This performance difference is related to the Baseline system's performances in each noise. Note that, in 20 dB SNR, word accuracies attained by the Baseline system in both Subway and Babble noise are below 87%, whereas the word accuracies are above 90% in Car and Exhibit noise. These performance differences show that performances of HMM-based recognition systems are dependent on types of environment noise. The differences can be attributed to many factors, e.g., effects on training accuracies of HMM parameters due to environment noise, which are difficult to analyze.

So far, the results show that this comparatively higher baseline in Car and Exhibit noise leave less room for noise adaptive speech recognition to have performance improvements.

However, the second observation is more interesting. The noise adaptive speech recognition has larger AERR over the Baseline system in Babble noise than that achieved in Subway noise. In Babble noise, the Adaptive system has 38.7% AERR, whereas it is 31.4% in Subway noise. Since the performances by the Baseline system in the above two types of noise are similar (In 20 dB SNR, word accuracies by the Baseline system are 84.7% and 86.1%, respectively, in Subway and Babble noise.), the difference in AERR can be largely attributed to the performances of the estimation process (16) in the Adaptive system when applied to different noise.

In order to compare the two types of noise, we view their histograms in log-spectral domain. [3] An example of the histogram of the Mel-scaled filter-bank log-spectral power is plotted in Fig. 4. It is seen that, in addition to a larger mean value in the 21st filter bank, the Subway noise has wider peak in distribution. This indicates that the Subway noise has larger variance than the Babble noise (Quantitatively, the variance of the Subway noise in the filter bank is 0.97, whereas the Babble noise has variance of 0.89.). Furthermore, the skewness of the Subway noise is larger than the Babble noise, which suggests that it may not be reliable to model the distribution of the Subway noise by a single Gaussian density. Similar observation can be found in other higher indexed Mel-scaled filter banks. The observation of large variance conflicts the assumption in Eq. (14), which assumes that the

[3] Two processes are applied before comparing them in distribution. First, the power of each noise has been normalized. Second, the length of noise sequence is equalized for the two types of noise.

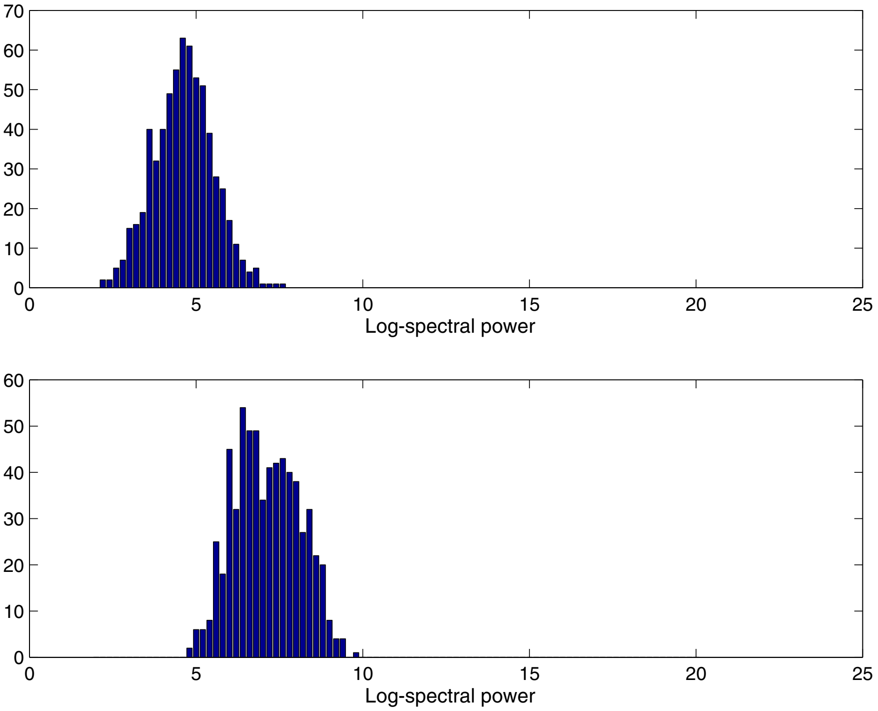

Fig. 4. Histogram of the log-spectral power of the Babble (upper) and Subway (lower) noise in the 21st Mel filter bank.

variance of $n_j^1(t)$ in Eq. (2) is small. In the view of this conflict, Babble noise can be seen as the noise that better fits the assumption in Eq. (14). This may account for the different performances of the Adaptive system in the two types of noise.

6. Discussions

The method presented in this paper treats the noise parameter as *time-varying*, and estimates the noise parameter by an EM-type sequential parameter estimation method. Note that the approach adopted by the method is quite different from some well-known methods, e.g., PMC (Gales and Young, 1997), CDCN (Acero, 1990), and VTS (Moreno et al., 1996), proposed in the literature for robust speech recognition. When applied to non-stationary environments, these methods make use of a *fixed* noise model, which is either HMM or GMM, and the state/mixture is considered to be representative to testing environments. For example, in PMC (Gales and Young, 1997), environment effects are compensated by expanding the original speech model according to the number of mixture/state in the noise model. Accordingly, recognition in non-stationary noise is carried out in expanded state sequences. Since the noise model is fixed, these methods assume that the statistics of the testing noise is represented by the trained noise model; i.e., the testing environment is known.

Some methods (Kim, 1998; Zhao et al., 2001; Afify and Siohan, 2001) follow the same approach adopted in this paper; i.e., estimation of noise parameter sequentially, which relaxes the above assumption and can possibly handle non-stationary and unknown environments. These methods employ sequential EM algorithm for time-varying noise parameter estimation in cepstral domain (Kim, 1998) and in linear spectral domain (Zhao et al., 2001). Afify and Siohan (2001) have considered the effects of changing rate of the noise spectral coefficients on parameter estimation and applied a scheme which adapts the forgetting factor ρ to adjust convergence rates of the estimation process. The forgetting factor is limited to a certain range in order to avoid divergence in estimation. Since the influence of the forgetting factor on parameter estimation is highly non-linear, the adaptation scheme involves manual efforts to set the range of the forgetting factor. Our work makes use of an extension of the sequential EM algorithm, the sequential Kullback proximal algorithm. In this method, the forgetting factor ρ is usually set to a constant smaller than 1.0 (e.g., 0.995). According to (Yao et al., 2001), the relaxation factor β_t provides an alternative way to control the convergence rate. When $\beta_t < 1.0$, the convergence rate of the sequential Kullback proximal algorithm by Eq. (11) is faster than that given by sequential EM algorithm (Yao et al., 2001). When $\beta_t > 1.0$, the estimation by sequential Kullback proximal algorithm can be smoother. Our experiments carried out so far showed that, whereas forgetting factor easily gives divergent estimation by sequential EM algorithm, estimation by sequential Kullback proximal algorithm is robust when varying the relaxation factor β_t. It would be interesting and important to devise an automatic way to control the relaxation factor β_t, which will be investigated in future.

In our work, a parametric model, Eq. (2), is employed for sequential parameter estimation when the original speech model is trained either from clean speech or from noisy speech. Given the objective by Eq. (5), the estimation is consistent and is independent from the parametric model. For example, instead of Eq. (2), the effects of noise can be modeled by a linear combination of bias terms in the cepstral domain (Deng et al., 2001). These bias terms can be estimated in batch way given stereo data (Deng et al., 2001) or sequentially. In that case, the modeling of the noise effects as a summation of bias terms is parametrical, but the parameters of the biases do not have explicit meaning. This is the situation when Eq. (2) is applied for parameter estimation when the speech models were trained from noisy speech, since the parametric model of Eq. (2) only provides explicit meaning of noise effects if the original speech $x_j^1(t)$ is clean. However, this does not prohibit its usage of Eq. (2) when speech models are trained from noisy speech because it is the objective in (5) instead of the explicit meaning of the estimation that is pursued.

The proposed noise adaptive speech recognition method is a general framework for sequential estimation when speech is modeled by GMM or HMMs. Although a particular parametric model (2) is applied in the current work, other parametric models, for example, (Deng et al., 2001; Surendran et al., 1999), can be used within this framework. This provides a guideline for application of the noise adaptive speech recognition to other speech features, for example, LDA based features. For such features, there are interesting questions on the formula of the parametric model for mapping between $x_j^1(t)$ and $y_j^1(t)$, and they deserve further investigations.

7. Conclusions

In this paper, a noise adaptive speech recognition approach is proposed for recognizing speech which is corrupted by additive non-stationary background noise. This approach sequentially estimates noise parameters, through which a non-linear parametric function adapts mean vectors of acoustic models. In the estimation process, posterior probability of state sequence given observation sequence and the previously estimated noise parameter sequence is approximated by the normalized joint likelihood of active partial paths and observation sequence given the previously estimated noise parameter sequence. The Viterbi process provides the normalized joint-likelihood. The acoustic models are not required to be trained from clean speech and they can be trained from noisy speech. The approach can be applied to perform continuous speech recognition in presence of non-stationary noise. Experiments conducted on speech contaminated by simulated and real non-stationary noise have shown that when acoustic models are trained from clean speech, the noise adaptive speech recognition system provides improvements in word accuracy when compared to the normal noise compensation system (which assumes the noise to be stationary) in slowly time-varying noise. When the acoustic models are trained from noisy speech, the noise adaptive speech recognition system has been found to be helpful to get improved performance in slowly time-varying noise over a system employing multi-conditional training. It has been observed that the optimal value of relaxation factor β_t used in the estimation process depends on the type of the contaminating noise. Further improvement in recognition performance can be achieved by incorporating the adaptation for the dynamic features in the present sequential estimation framework and by refinement of the parametric function to model noise effects.

Acknowledgements

The authors thank the three anonymous reviewers whose comments have substantially improved the presentation of the paper. The first author thanks Dr. S. Yamamoto, president of the ATR SLT laboratories, for his support of the work. The research was supported in part by the Telecommunications Advanced Organization of Japan.

Appendix A. Approximation of the environment effects on speech features

Effect of additive noise on speech power at the jth bin of the filter bank can be approximated by (Gales and Young, 1997; Acero, 1990)

$$\sigma_y^2(j) = \sigma_x^2(j) + \sigma_n^2(j) \tag{A.1}$$

where $\sigma_y^2(j)$, $\sigma_x^2(j)$, and $\sigma_n^2(j)$ denote noisy speech power, speech power and additive noise power, respectively, in the filter-bank bin j.

This equation can be written in the log-spectral domain as follows:

$$\begin{aligned}\log(\sigma_x^2(j)+\sigma_n^2(j)) &= \log\sigma_x^2(j)+\log\left(1+\frac{\sigma_n^2(j)}{\sigma_x^2(j)}\right)\\ &= \log\sigma_x^2(j)+\log(1+\exp(\log\sigma_n^2(j)\\ &\quad -\log\sigma_x^2(j)))\end{aligned} \tag{A.2}$$

By substituting $x_j^1 = \log\sigma_x^2(j)$, $n_j^1 = \log\sigma_n^2(j)$ and $y_j^1 = \log\sigma_y^2(j)$, this equation can be written as,

$$y_j^1 = x_j^1 + \log(1+\exp(n_j^1 - x_j^1)) \tag{A.3}$$

Appendix B. The objective function of the sequential Kullback proximal algorithm

The sequential Kullback proximal algorithm (Yao et al., 2001) is a sequential version of the Kullback proximal algorithm (Chrétien and Hero, 2000) for maximum-likelihood estimation. In the sequential Kullback proximal algorithm (Yao et al., 2001), the cost function for the iterative procedure is given as the log-likelihood function (shown in Eq. (6)) regularized by a K–L divergence; i.e.,

$$\begin{aligned}&l_t(\hat{\lambda}_N(t)) - \beta_t I_t(\lambda_N^\star(t);\hat{\lambda}_N(t))\\ &\quad = l_t(\hat{\lambda}_N(t)) - I_t(\lambda_N^\star(t);\hat{\lambda}_N(t))\\ &\qquad -(\beta_t-1)I_t(\lambda_N^\star(t);\hat{\lambda}_N(t))\end{aligned} \tag{B.1}$$

where $I_t(\lambda_N^\star(t);\hat{\lambda}_N(t))$ is the K–L divergence between the posterior distributions $P(S(t)|\boldsymbol{Y}(t),(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))$ and $P(S(t)|\boldsymbol{Y}(t),(\boldsymbol{\Lambda}_N(t-1),\hat{\lambda}_N(t)))$; i.e.,

$$\begin{aligned}&I_t(\lambda_N^\star(t);\hat{\lambda}_N(t))\\ &\quad = \sum_{S(t)} P(S(t)|\boldsymbol{Y}(t),(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))\\ &\qquad \times\log\frac{P(S(t)|\boldsymbol{Y}(t),(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))}{P(S(t)|\boldsymbol{Y}(t),(\boldsymbol{\Lambda}_N(t),\hat{\lambda}_N(t)))}\\ &\quad = l_t(\hat{\lambda}_N(t))\\ &\qquad +\sum_{S(t)} P(S(t)|\boldsymbol{Y}(t),(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))\\ &\qquad \times\log\frac{P(S(t)|\boldsymbol{Y}(t),(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))}{P(\boldsymbol{Y}(t),S(t)|(\boldsymbol{\Lambda}_N(t),\hat{\lambda}_N(t)))}\\ &\quad = -Q_t(\lambda_N^\star(t);\hat{\lambda}_N(t)) + l_t(\hat{\lambda}_N(t))\\ &\qquad +\sum_{S(t)} P(S(t)|\boldsymbol{Y}(t),(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))\\ &\qquad \times\log P(S(t)|\boldsymbol{Y}(t),(\boldsymbol{\Lambda}_N(t-1),\lambda_N^\star(t)))\end{aligned} \tag{B.2}$$

where the auxiliary function $Q_t(\lambda_N^\star(t);\hat{\lambda}_N(t))$ is defined in Eq. (9).

Substituting above equation into (B.1), we obtain,

$$\begin{aligned}&l_t(\hat{\lambda}_N(t)) - \beta_t I_t(\lambda_N^\star(t);\hat{\lambda}_N(t))\\ &\quad = Q_t(\lambda_N^\star(t);\hat{\lambda}_N(t)) - (\beta_t-1)I_t(\lambda_N^\star(t);\hat{\lambda}_N(t)) + Z\end{aligned} \tag{B.3}$$

where Z is a function without relation to $\hat{\lambda}_N(t)$. We thus obtain (11) as the objective function for the sequential parameter estimation.

Appendix C. Properties of the sequential Kullback proximal algorithm

C.1. Sequential EM algorithm is a particular case of the sequential Kullback proximal algorithm

When $\beta_t = 1.0$, according to (B.3), the objective function $l_t(\hat{\lambda}_N(t)) - \beta_t I_t(\lambda_N^\star(t); \hat{\lambda}_N(t))$ to be maximized is equivalent to maximization of $Q_t(\lambda_N^\star(t); \hat{\lambda}_N(t))$, which is the objective function to be maximized by sequential EM algorithm.

C.2. Monotonic likelihood property

According to the objective function defined by the sequential Kullback proximal algorithm, it has

$$\begin{aligned} l_t(\hat{\lambda}_N(t)) - l_t(\hat{\lambda}_N(t-1)) &\geqslant \beta_t I_t(\hat{\lambda}_N(t-1); \hat{\lambda}_N(t)) \\ &\quad - \beta_t I_t(\hat{\lambda}_N(t-1); \hat{\lambda}_N(t-1)) = \beta_t I_t(\hat{\lambda}_N(t-1); \hat{\lambda}_N(t)) \end{aligned} \tag{C.1}$$

Since $\beta_t \in R^+$, $I_t(\hat{\lambda}_N(t-1); \hat{\lambda}_N(t-1)) = 0$ and $I_t(\hat{\lambda}_N(t-1); \hat{\lambda}_N(t)) \geqslant 0$, we prove that the sequential Kullback proximal algorithm can achieve the objective function (5).

Appendix D. Derivation of the sequential Kullback proximal algorithm

The first- and second-order differential of the K–L divergence of (B.2) are given respectively as,

$$\frac{\partial I_t(\lambda_N^\star(t); \hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)} = -\frac{\partial Q_t(\lambda_N^\star(t); \hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)} + \frac{\partial l_t(\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)} \tag{D.1}$$

$$\frac{\partial^2 I_t(\lambda_N^\star(t); \hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)^2} = -\frac{\partial^2 Q_t(\lambda_N^\star(t); \hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)^2} + \frac{\partial^2 l_t(\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)^2} \tag{D.2}$$

Assume that $\left.\frac{\partial I_t(\lambda_N^\star(t); \hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)}\right|_{\hat{\lambda}_N(t)=\lambda_N^\star(t)} = 0$ has been achieved, and it thus holds

$$\left.\frac{\partial l_t(\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)}\right|_{\hat{\lambda}_N(t)=\lambda_N^\star(t)} = \left.\frac{\partial Q_t(\lambda_N^\star(t); \hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)}\right|_{\hat{\lambda}_N(t)=\lambda_N^\star(t)} \tag{D.3}$$

With the second-order Taylor series expansion of the objective function (B.1) at $\lambda_N^\star(t)$, the updating of noise parameter is given as,

$$\hat{\lambda}_N(t) \leftarrow \hat{\lambda}_N(t-1) - \left.\frac{\frac{\partial(l_t(\hat{\lambda}_N(t)) - \beta_t I_t(\lambda_N^\star(t); \hat{\lambda}_N(t)))}{\partial \hat{\lambda}_N(t)}}{\frac{\partial^2(l_t(\hat{\lambda}_N(t)) - \beta_t I_t(\lambda_N^\star(t); \hat{\lambda}_N(t)))}{\partial \hat{\lambda}_N(t)^2}}\right|_{\hat{\lambda}_N(t)=\hat{\lambda}_N(t-1)} \tag{D.4}$$

By (D.2) and (D.3), the updating is given as,

$$\hat{\lambda}_N(t) \leftarrow \hat{\lambda}_N(t-1) - \left.\frac{\frac{\partial Q_t(\lambda_N^\star(t); \hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)}}{\beta_t \frac{\partial^2 Q_t(\lambda_N^\star(t); \hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)^2} + (1-\beta_t) \frac{\partial^2 l_t(\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)^2}}\right|_{\hat{\lambda}_N(t)=\hat{\lambda}_N(t-1)} \tag{D.5}$$

The derivation of the updating formula for the auxiliary function $Q_t(\hat{\lambda}_N(t-1); \hat{\lambda}_N(t))$ can be seen in (Krishnamurthy and Moore, 1993). We briefly describe the derivation in this paper. Since

$$\begin{aligned} &Q_t(\hat{\lambda}_N(t-1); \hat{\lambda}_N(t)) \\ &\quad = \sum_{S(t)} P(S(t) | \boldsymbol{Y}(t), \Lambda_X, (\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t-1))) \\ &\qquad \cdot \log[P(S(t-1) | \boldsymbol{Y}(t-1), \Lambda_X, \boldsymbol{\Lambda}_N(t-1)) \\ &\qquad \cdot a_{s(t-1)s(t)} b_{s(t)}(\boldsymbol{y}(t))] \\ &\quad = \sum_{S(t-1)} P(S(t-1) | \boldsymbol{Y}(t-1), \Lambda_X, \boldsymbol{\Lambda}_N(t-1)) \\ &\qquad \times \log b_{s(t-1)}(\boldsymbol{y}(t-1)) \\ &\qquad + \sum_{s(t)} P(s(t) | \boldsymbol{Y}(t), \Lambda_X, (\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t-1))) \\ &\qquad \times \log b_{s(t)}(\boldsymbol{y}(t)) + Z \end{aligned}$$

Denote $Q_{t-1}(\hat{\lambda}_N(t-2);\hat{\lambda}_N(t)) = \sum_{S(t-1)} P(S(t-1)| \boldsymbol{Y}(t-1), \Lambda_X, \boldsymbol{\Lambda}_N(t-1)) \log b_{s(t-1)}(\boldsymbol{y}(t-1))$. Assume that $\hat{\lambda}_N(t-1)$ has made $\frac{\partial Q_{t-1}(\hat{\lambda}_N(t-2);\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N}|_{\hat{\lambda}_N(t)=\hat{\lambda}_N(t-1)} = 0$. We thus obtain the first- and second-order derivative of the auxiliary function with respect to the noise parameter, which are shown in (17) and (18), respectively.

In order to calculate $\frac{\partial^2 l_t(\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)^2}$, define forward accumulated likelihood at state i and mixture m as $\alpha_t(i,m;\hat{\lambda}_N(t)) = P(\boldsymbol{Y}(t), s(t)=i, \quad k(t)=m|\Lambda_X, (\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t)))$, and accordingly the forward accumulated likelihood at state i, $\alpha_t(i;\hat{\lambda}_N(t)) = \sum_m \alpha_t(i,m;\hat{\lambda}_N(t))$. They are related as shown below.

$$\alpha_t(i,m;\hat{\lambda}_N(t)) = \sum_{l=1}^{\Upsilon} \alpha_{t-1}(l;\hat{\lambda}_N(t-1)) a_{li} w_{im} b_{im}(\boldsymbol{y}(t)) \tag{D.6}$$

Since $l_t(\hat{\lambda}_N(t)) = \log \sum_{im} \alpha_t(i,m;\hat{\lambda}_N(t))$, it has

$$\left.\frac{\partial l_t(\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)}\right|_{\hat{\lambda}_N(t)=\hat{\lambda}_N(t-1)} = \left.\frac{\partial \log \sum_{i=1}^{\Upsilon}\sum_{m=1}^{M} \alpha_t(i,m;\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)}\right|_{\hat{\lambda}_N(t)=\hat{\lambda}_N(t-1)} = \left.\frac{\sum_{i=1}^{\Upsilon}\sum_{m=1}^{M} \frac{\partial \alpha_t(i,m;\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)}}{\sum_{j=1}^{\Upsilon}\sum_{m=1}^{M} \alpha_t(j,m;\hat{\lambda}_N(t))}\right|_{\hat{\lambda}_N(t)=\hat{\lambda}_N(t-1)} \tag{D.7}$$

By (15) and (D.6), it has

$$\frac{\partial \alpha_t(i,m;\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)} = \alpha_t(i,m;\hat{\lambda}_N(t)) \frac{\partial \log b_{im}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N(t)} \tag{D.8}$$

Substituting the above equation into (D.7), we have

$$\left.\frac{\partial l_t(\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)}\right|_{\hat{\lambda}_N(t)=\hat{\lambda}_N(t-1)} = \sum_{i=1}^{\Upsilon}\sum_{m=1}^{M} \gamma_t(i,m;\hat{\lambda}_N(t)) \left.\frac{\partial \log b_{im}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N(t)}\right|_{\hat{\lambda}_N(t)=\hat{\lambda}_N(t-1)}$$

where $\gamma_t(i,m;\hat{\lambda}_N(t)) = \frac{\alpha_t(i,m;\hat{\lambda}_N(t))}{\sum_{lm} \alpha_t(l,m;\hat{\lambda}_N(t))}$ represents the posterior probability at state i and mixture m given observation sequence $\boldsymbol{Y}(t)$ and noise parameter sequence $(\boldsymbol{\Lambda}_N(t-1), \hat{\lambda}_N(t))$. We thus obtain

$$\frac{\partial^2 l_t(\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)^2} = \sum_{i=1}^{\Upsilon}\sum_{m=1}^{M} \frac{\partial \gamma_t(i,m;\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)} \frac{\partial \log b_{im}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N(t)} + \sum_{i=1}^{\Upsilon}\sum_{m=1}^{M} \gamma_t(i,m;\hat{\lambda}_N(t)) \frac{\partial^2 \log b_{im}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N^2(t)} \tag{D.9}$$

Noting that $\gamma_t(i,m;\hat{\lambda}_N(t)) = \frac{\alpha_t(i,m;\hat{\lambda}_N(t))}{\sum_{i=1}^{\Upsilon}\sum_{k=1}^{M} \alpha_t(i,k;\hat{\lambda}_N(t))}$ and referring to (D.8), we have

$$\frac{\partial \gamma_t(i,m;\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)} = \gamma_t(i,m;\hat{\lambda}_N(t)) \left[\frac{\partial \log b_{im}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N(t)} - \sum_{l=1}^{\Upsilon}\sum_{k=1}^{M} \gamma_t(l,k;\hat{\lambda}_N(t)) \frac{\partial \log b_{lk}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N(t)} \right] \tag{D.10}$$

Substituting above equation into (D.9), we have

$$\frac{\partial^2 l_t(\hat{\lambda}_N(t))}{\partial \hat{\lambda}_N(t)^2} = \sum_{i=1}^{\Upsilon}\sum_{m=1}^{M} \gamma_t(i,m;\hat{\lambda}_N(t)) \left[\left(\frac{\partial \log b_{im}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N(t)} \right)^2 + \frac{\partial^2 \log b_{im}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N(t)^2} \right] - \left(\sum_{i=1}^{\Upsilon}\sum_{m=1}^{M} \gamma_t(i,m;\hat{\lambda}_N(t)) \frac{\partial \log b_{im}(\boldsymbol{y}(t))}{\partial \hat{\lambda}_N(t)} \right)^2 \tag{D.11}$$

References

Acero, A., 1990. Acoustical and environmental robustness in automatic speech recognition. Ph.D. Thesis, Carnegie Mellon University.

Afify, M., Siohan, O., 2001. Sequential noise estimation with optimal forgetting for robust speech recognition. In: ICASSP. pp. 229–232.

Cerisara, C., Rigazio, L., Boman, R., Junqua, J.-C., 2001. Environmental adaptation based on first-order approximation. In: ICASSP. pp. 213–216.

Chrétien, S., Hero III, A.O., 2000. Kullback proximal point algorithms for maximum likelihood estimation. IEEE Trans. Informat. Theory 46 (5), 1800–1810.

Deng, L., Acero, A., Jiang, L., Droppo, J., Huang, X.D., 2001. High-performance robust speech recognition using stereo training data. In: ICASSP. pp. 301–304.

Ephraim, Y., Malah, D., 1984. Speech enhancement using a minimum mean square error short-time spectral amplitude estimator. IEEE. Trans. Acoust. Speech Signal Process. 32 (6), 1109–1121.

Frey, B., Deng, L., Acero, A., Kristjansson, T., 2001. ALGONQUIN: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition. In: EUROSPEECH. pp. 901–904.

Gales, M., Young, J., 1997. Robust speech recognition in additive and convolutional noise using parallel model combination. Computer Speech Lang. 9, 289–307.

Hanson, B.A., Applebaum, T.H., 1990. Robust speaker-independent word recognition using static, dynamic and acceleration features: experiments with Lombard and noisy speech. In: ICASSP. pp. 857–860.

Hermansky, H., 1990. Perceptual linear predictive (PLP) analysis for speech. J. Acoustic. Soc. Amer. 87 (4), 1738–1752.

Hirsch, H.G., Pearce, D., 2000. The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In: ISCA ITRW ASR2000.

Kim, N.S., 1998. Non-stationary environment compensation based on sequential estimation. IEEE Signal Process. Lett. 5 (3), 57–59.

Krishnamurthy, V., Moore, J.B., 1993. On-line estimation of hidden Markov model parameters based on the Kullback–Leibler information measure. IEEE Trans. Signal Process. 41 (8), 2557–2573.

Moreno, P.J., Raj, B., Stern, R.M., 1996. A vector Taylor series approach for environment-independent speech recognition. In: ICASSP. pp. 733–736.

Morris, A.C., Cooke, M.P., Green, P.D., 1998. Some solutions to the missing feature theory in data classification, with application to noise robust ASR. In: ICASSP. pp. 737–740.

Rahim, M.G., Juang, B.-H., 1996. Signal bias removal by maximum likelihood estimation for robust telephone speech recognition. IEEE Trans. Speech Audio Process. 4 (1), 19–30.

Sagayama, S., Yamaguchi, Y., Takahashi, S., Takahashi, J., 1997. Jacobian approach to fast acoustic model adaptation. In: ICASSP. pp. 835–838.

Sankar, A., Lee, C.-H., 1996. A maximum-likelihood approach to stochastic matching for robust speech recognition. IEEE Trans. Speech Audio Process. 4 (3), 190–201.

Surendran, A.C., Lee, C.-H., Rahim, M., 1999. Nonlinear compensation for stochastic matching. IEEE Trans. Speech Audio Process. 7 (6), 643–655.

Takiguchi, T., Nakamura, S., Shikano, K., 2000. Speech recognition for a distant moving speaker based on HMM decomposition and separation. In: ICASSP. pp. 1403–1406.

Varga, A., Moore, R.K., 1990. Hidden Markov model decomposition of speech and noise. In: ICASSP. pp. 845–848.

Vaseghi, S.V., Milner, B.P., 1997. Noise compensation methods for hidden Markov model speech recognition in adverse environments. IEEE Trans. Speech Audio Process. 5 (1), 11–21.

Yao, K., Nakamura, S., 2002. Sequential noise compensation by sequential Monte Carlo method. In: Advances in Neural Information Processing Systems. MIT press, pp. 1213–1220.

Yao, K., Paliwal, K.K., Nakamura, S., 2001. Sequential noise compensation by a sequential Kullback proximal algorithm. In: EUROSPEECH. pp. 1139–1142.

Yao, K., Paliwal, K., Nakamura, S., 2002. Noise adaptive speech recognition in time-varying noise based on sequential Kullback proximal algorithm. In: ICASSP. pp. 189–192.

Young, S., 1997. The HTK BOOK. Ver. 2.1. Cambridge University.

Zhao, Y., 2000. Frequency-domain maximum likelihood estimation for automatic speech recognition in additive and convolutive noises. IEEE Trans. Speech Audio Process. 8 (3), 255–266.

Zhao, Y. et al., 2001. Recursive estimation of time-varying environments for robust speech recognition. In: ICASSP. pp. 225–228.