Recursive Estimation of Nonstationary Noise Using Iterative Stochastic Approximation for Robust Speech Recognition

Li Deng, Jasha Droppo, and Alex Acero

***Abstract*—We describe a novel algorithm for recursive estimation of nonstationary acoustic noise which corrupts clean speech, and a successful application of the algorithm in the speech feature enhancement framework of noise-normalized SPLICE for robust speech recognition. The noise estimation algorithm makes use of a nonlinear model of the acoustic environment in the cepstral domain. Central to the algorithm is the innovative iterative stochastic approximation technique that improves piecewise linear approximation to the nonlinearity involved and that subsequently increases the accuracy for noise estimation. We report comprehensive experiments on SPLICE-based, noise-robust speech recognition for the AURORA2 task using the results of iterative stochastic approximation. The effectiveness of the new technique is demonstrated in comparison with a more traditional, MMSE noise estimation algorithm under otherwise identical conditions. The word error rate reduction achieved by iterative stochastic approximation for recursive noise estimation in the framework of noise-normalized SPLICE is 27.9% for the multicondition training mode, and 67.4% for the clean-only training mode, respectively, compared with the results using the standard cepstra with no speech enhancement and using the baseline HMM supplied by AURORA2. These represent the best performance in the clean-training category of the September-2001 AURORA2 evaluation. The relative error rate reduction achieved by using the same noise estimate is increased to 48.40% and 76.86%, respectively, for the two training modes after using a better designed HMM system. The experimental results demonstrated the crucial importance of using the newly introduced iterations in improving the earlier stochastic approximation technique, and showed sensitivity of the noise estimation algorithm's performance to the forgetting factor embedded in the algorithm.**

***Index Terms*—Author, please supply your own keywords or send a blank e-mail to keywords@ieee.org to receive a list of suggested keywords.**

I. Introduction

NOISE-ROBUST speech recognition under the acoustic environment where speech is corrupted by unknown noise, especially by unknown and fast changing nonstationary noise, has been a long-standing and unsolved problem (e.g., [19]). One important class of techniques for noise-robust recognition operate by enhancing speech features via front-end denoising, producing cleaned inputs to speech recognizers for decoding. Such techniques have also been effectively used to enhance either naturally or intentionally corrupted training data, followed by subsequent re-training of the HMM systems to remove the residual mismatch between the training and test sets after speech feature enhancement [4]. Use of denoising or preprocessing in this way is shown to be superior to re-training recognizers under matched noisy conditions with no preprocessing, beating the conventional wisdom that the matched noisy condition sets the upper limit for the performance. Recently, we have successfully developed a class of front-end denoising algorithms based on the use of a limited set of stereo training data [4], [5]. The basic version of the algorithm has been called SPLICE, short for Stereo-based Piecewise Linear Compensation for Environment. For most of the noisy test speech data that have been collected and generated internally at Microsoft, we found that SPLICE has been highly effective.[1]

More recently, we started applying SPLICE to the AURORA2 task [12], which is noisy connected digit recognition used in the September-2001 evaluation participated by about 20 systems. AURORA2 is based on the TIDigits database that is corrupted digitally by passing them through a linear filter and/or by adding different types of realistic, nonstationary noises at a wide range of SNRs. The AURORA2 task has strongly constrained the coverage of the noise conditions in designing the stereo training data. We discovered in our earlier AURORA2 work that when the training set used to obtain the correction vectors in SPLICE are under very different noise environments than the environment for the test data, the performance often becomes undesirably low [8]. One obvious solution to this mismatch problem is to normalize, in an instantaneous-SNR-dependent manner, the test and training environments. Some carefully designed diagnostic experiments have confirmed the crucial importance of the accuracy of noise estimation in successful applications of denoising under seriously mismatched conditions between the SPLICE's training and deployment environments. The nonstationary nature of the noise represents one major source of difficulty for accurate noise estimation, which will be specifically addressed in this work.

Toward solving the problem of accurate noise estimation (especially for the nonstationary noise), we have developed an effective recursive noise estimation method based on a nonlinear model of the acoustic environment. There is vast literature in signal processing on recursive algorithms, also called sequential, online, or adaptive algorithms, and on stochastic

Manuscript received September 13, 2001; revised June 9, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shrikanth Narayanan.

The authors are with the Microsoft Research, Redmond, WA 98052 USA (e-mail: deng@microsoft.com).

Digital Object Identifier 10.1109/TSA.2003.818076

[1]Some of the results were reported in [4], [5], [7].

1063-6676/03$17.00 © 2003 IEEE

IEEE Proof

approximations which provide theoretical foundations for the recursive algorithms (e.g., [3], [10], [15], [18], [20]). One main contribution of the work reported in this paper is to devise a new technique that generalizes these well-studied algorithms so that they can be effectively used to handle the potentially complex nonlinearity involved in the underlying models for generating noisy speech data. Specifically, the novel technique developed for the generalized algorithm makes use of the newly developed iterative stochastic approximation technique to achieve a high degree of accuracy in approximating the nonlinearity via truncated Taylor series expansion. This is accomplished by introducing new auxiliary or nuisance parameters (as the Taylor series expansion points) that are jointly optimized with the desired parameters. This new technique has been successfully applied to improve the accuracy of the nonstationary noise estimate, which is exploited in noise-normalized SPLICE as the basis for enhancing the cepstra of speech embedded in nonstationary noises. Significant performance improvement has been achieved in noise-robust speech recognition using such enhanced speech features. In addition to its successful use in noise-normalized SPLICE, the new technique presented in this paper for high-performance nonstationary noise estimation has applications in other areas of speech processing such as spectral subtraction and voice activity detection (not reported in this paper).

The organization of this paper is as follows. Section II, we outline the noise-normalized SPLICE framework in which the new nonstationary noise estimation technique is exploited in cepstral feature enhancement intended for noise-robust speech recognition. Section III, we first introduced a nonlinear model of the acoustic environment in the cepstral domain, and use it as the basis for developing the recursive-EM algorithm for noise estimation using a linearized version of the nonlinear model. The developed algorithm is built upon some recent noise estimation work in robust speech recognition (e.g., [2], [14]), but it generalizes the earlier work by making the linearization process include auxiliary parameters that are subject to joint optimization with the noise parameters. One effective approach to solving this joint optimization problem is presented in Section IV, using iterative stochastic approximation. Section V, we report comprehensive experimental results that demonstrate the effectiveness of the new noise estimation method for the AURORA2 task using the noise-normalized SPLICE framework for speech feature enhancement. We provide evidence that demonstrates the crucial importance of using the newly introduced iterations in stochastic approximation in terms of the noise estimation accuracy (measured by root mean square error) and of the speech recognition accuracy (measured by error rate reduction). We further report the results on the importance of the forgetting mechanism embedded in the algorithm that enables the algorithm to effectively track the time-varying noise.

II. Noise-Normalized SPLICE for Cepstral Feature Enhancement

In this section, we outline the noise-normalized SPLICE framework, which makes use of the recursively estimated noise to enhance the noisy speech in the Mel-Frequency Cepstral Coefficient (MFCC) domain. This section summarizes and enriches the earlier descriptions of SPLICE, including its various improved versions, which appeared in [4], [5], [8]. While noise estimation, rather than speech feature enhancement, is the primary focus of this paper, the latter serves as the best application area of the former. Hence, we present the framework of enhancement here in order to set up the background for the use of the outputs of the novel recursive noise estimation algorithm, which will be presented following this section.

A. SPLICE Basics

One of the two basic modeling assumptions in SPLICE is that the noisy (corrupted) speech MFCC vector $\mathbf{y}$, under each distinct distortion condition, follows a mixture of Gaussians:

$$p(\mathbf{y}) = \sum_i p(\mathbf{y}|i)p(i), \text{ where}$$
$$p(\mathbf{y}|i) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i).$$

The mixture component i is a discrete random variable. It takes a set of discrete values, one for each cepstral-space partitioning. A piecewise linear approximation between the clean speech MFCC vector $\mathbf{x}$ and its noisy counterpart $\mathbf{y}$ is made for each such partitioning.

The second modeling assumption in SPLICE is that the conditional PDF for clean speech $\mathbf{x}$, given noisy speech $\mathbf{y}$ and the mixture component i, is also a Gaussian. The mean vector is assumed to be a shifted version of noisy speech $\mathbf{y}$. That is,

$$p(\mathbf{x}|\mathbf{y}, i) = \mathcal{N}(\mathbf{x}; \mathbf{y} + \mathbf{r}_i, \Gamma_i) \tag{1}$$

where $\mathbf{r}_i$ are the correction vectors that need to be trained using stereo training data.

The above two basic assumptions in SPLICE give rise to a simple yet rigorous MMSE estimate of clean speech MFCC vectors from their distorted counterparts. The MMSE is the mean of the conditional PDF of $p(\mathbf{x}|\mathbf{y}) = \sum_i p(\mathbf{x}, i|\mathbf{y}) = \sum_i p(i|\mathbf{y})p(\mathbf{x}|\mathbf{y}, i)$. This gives

$$\hat{\mathbf{x}}_{\text{MMSE}} = \int \mathbf{x}p(\mathbf{x}|\mathbf{y})d\mathbf{x} = \sum_i p(i|\mathbf{y}) \int \mathbf{x}p(\mathbf{x}|\mathbf{y}, i)d\mathbf{x}$$
$$= \sum_i p(i|\mathbf{y})(\mathbf{y} + \mathbf{r}_i) = \mathbf{y} + \sum_i p(i|\mathbf{y})\mathbf{r}_i \tag{2}$$

where

$$p(i|\mathbf{y}) = \frac{p(\mathbf{y}|i)p(i)}{\sum_i p(\mathbf{y}|i)p(i)}$$

according to Bayes rule.

A fast implementation of SPLICE is to approximate the weights $p(i|\mathbf{y})$ above according to

$$p(\hat{i}|\mathbf{y}) \approx \begin{cases} 1, & \hat{i} = \arg\max_i \ p(i|\mathbf{y}) \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

This turns the MMSE estimate in (2) into the approximate MAP estimate that was originally formulated in [4]. This approximate MAP estimate consists of two sequential steps of operation: 1) finding optimal "codeword" $\hat{i}$ using the VQ codebook based on

IEEE Proof

the parameters ($\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$) and 2) adding the codeword-dependent vector $\mathbf{r}_i$ to the noisy speech vector.

The distribution parameters in $p(\mathbf{y})$ are trained for each separate noisy condition using noisy speech data.[2] The correction vectors $\mathbf{r}_i$ are trained using stereo training data ($\mathbf{x}_t$ and $\mathbf{y}_t$). Maximum likelihood training is used, and the estimation formula is

$$\mathbf{r}_i = \frac{\sum_t p(i|\mathbf{y}_t)(\mathbf{x}_t - \mathbf{y}_t)}{\sum_t p(i|\mathbf{y}_t)}, \text{ where} \tag{4}$$

$$p(i|\mathbf{y}_t) = \frac{p(\mathbf{y}_t)p(i)}{\sum_i p(\mathbf{y}_t|i)p(i)}. \tag{5}$$

B. Noise-Normalized SPLICE

The above basic version of the SPLICE algorithm for denoising in the MFCC domain has been improved to normalize the difference in noise conditions between the training and test data sets.[3] The improved, noise-normalized SPLICE enhances the basic version as follows. Instead of building codebooks for $\mathbf{y}_1^T = \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T$ from the training set, they are built from $\mathbf{y}_1^T - \hat{\mathbf{n}}_1^T$, where $\hat{\mathbf{n}}_1^T = \hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2, \ldots, \hat{\mathbf{n}}_T$ is the estimated noise vector sequence from $\mathbf{y}_1^T$. Then the correction vectors are estimated from the training set using the noise-normalized stereo data $(\mathbf{y}_1^T - \hat{\mathbf{n}}_1^T)$ and $(\mathbf{x}_1^T - \hat{\mathbf{n}}_1^T)$.[4] For denoising in the test data, the noise-normalized noisy MFCCs $(\mathbf{y}_t - \hat{\mathbf{n}}_t)$ are used to obtain the noise-normalized MMSE estimate via the basic SPLICE enhancement, and then the noise normalization is undone by adding $\hat{\mathbf{n}}_t$ back to the MMSE estimate.

In addition to noise normalization, another improvement over the basic version of SPLICE as used in our experiments is to smooth the correction vectors through time frames using a carefully designed low-pass filter after completing the training of the correction vectors [8].

III. Recursive Estimation of Nonstationary Noise

Given the general outline of the noisy speech enhancement framework presented above, we now describe how the noise vector, $\hat{\mathbf{n}}_t$, which varies in time (nonstationary) and is needed for the noise normalization, is estimated via a recursive-EM algorithm in a frame-by-frame manner. As a basis for this algorithm, we first present a nonlinear environment model for the constraining relationship among the noisy speech, clean speech, and noise in the cepstral domain.

[2]For the AURORA2 experiments [12], which will be reported in Section V, a total of 17 such distributions are used from the Set-A noisy data in the experiments. These 17 distributions correspond to each of the four noise conditions and for each of the four SNRs (5 dB, 10 dB, 15 dB, and 20 dB), in addition to the distribution trained using clean speech data in Set-A.

[3]In the AURORA2 database, the training data come from Set-A and are used to train the 17 codebooks and 17 sets of the codeword dependent correction vectors. The test data consist of all Sets-A, -B, and -C.

[4]The correction vectors trained in this noise-normalized SPLICE will be different from those in the basic version of SPLICE. This is because the codebooks are different (i.e., $p(i|\mathbf{y}_t)$ is changed to $p(i|\mathbf{y}_t - \mathbf{n}_t)$ in (4)), although the term $(\mathbf{x}_t - \mathbf{y}_t)$ in (4) remains the same due to the common $\hat{\mathbf{n}}_1^T$ subtracted from both $\mathbf{x}_1^T$ and $\mathbf{y}_1^T$.

A. Nonlinear Model for Acoustic Environment

Using a linear system model in time domain, one can easily show that the power spectra of distorted or noisy speech $|Y[k]|^2$ are related to those of clean speech $|X[k]|^2$, noise $|N[k]|^2$, and the channel transfer function $|H[k]|^2$ according to

$$|Y[k]|^2 \approx |X[k]|^2|H[k]|^2 + |N[k]|^2 \tag{6}$$

where the approximation is due to omission of the cross term $2|X[k]H[k]||N[k]|\cos\theta_k$ (θ_k represents the random angle between the two complex variables $N[k]$ and $X[k]H[k]$).

Equation (6) in the domain of power spectra can be shown to be equivalent to the following parametric model of the acoustic environment in the cepstral domain ([1], [17])

$$\mathbf{y} \approx \mathbf{h} + \mathbf{x} + \mathbf{C}\ln\big(\mathbf{I} + \exp[\mathbf{C}^T(\mathbf{n} - \mathbf{h} - \mathbf{x})]\big) \tag{7}$$

where $\mathbf{y}$ and $\mathbf{x}$ are distorted and clean speech cepstral vectors, respectively. $\mathbf{n}$ and $\mathbf{h}$ are cepstral vectors for the additive noise and impulse response of convolutional distortion, respectively. $\mathbf{C}$ is the discrete cosine transform matrix. To simplify the notation, we define the vector function $\mathbf{g}(\cdot)$ of

$$\mathbf{g}(\mathbf{z}) = \mathbf{C}\ln\big[\mathbf{I} + \exp[\mathbf{C}^T\mathbf{z}]\big). \tag{8}$$

Then, we can write the model of (7) in short-hand

$$\mathbf{y} \approx \mathbf{h} + \mathbf{x} + \mathbf{g}(\mathbf{n} - \mathbf{h} - \mathbf{x}). \tag{9}$$

Note matrix $\mathbf{C} = [c_{ij}]$ is not a square one, and (9) can be rewritten component-by-component as follows:

$$y_i \approx x_i + h_i + \sum_{j=1}^{\mathcal{J}} c_{ij}\ln\left(1 + \exp\left[\sum_{k=1}^{\mathcal{I}} c_{kj}(n_k - h_k - x_k)\right]\right) \tag{10}$$

where $i = 0, 1, 2, \ldots, \mathcal{I} - 1$; $\mathcal{I}$ is the dimensionality of cepstral vectors and $\mathcal{J}$ is the dimensionality of Mel-frequency log-channel spectra. We set $\mathcal{I} = 13$ and $\mathcal{J} = 24$ in the AURORA2 experiments described in Section V.

The model (9) that relates $\mathbf{x}$, $\mathbf{y}$, $\mathbf{n}$ and $\mathbf{h}$ is nonlinear. Compared with the linear model of (6) in terms of power spectra, the nonlinear model has higher complexity, but is more desirable because the cepstral domain in which the model is expressed is the same domain as that on which most speech recognizers are operating. For developing the recursive estimation algorithm based on the nonlinear model of (9), we make approximation by truncating Taylor series expansion of the nonlinearity, around an updated operating point, up to the linear term. In this way, when $\mathbf{x}$, $\mathbf{n}$, and $\mathbf{h}$ are Gaussian and since $\mathbf{g}(\mathbf{n} - \mathbf{h} - \mathbf{x})$ is linearized, we effectively approximate $\mathbf{y}$ with a Gaussian. In this paper, we consider additive noise only, for which $\mathbf{h} = 0$. Let $\boldsymbol{\mu}_0^x$ and $\mathbf{n}_0$ be the operating points for the first-order Taylor series expansion of $\mathbf{y}$. We then have

$$\begin{aligned}\mathbf{y} \approx \mathbf{x} + \mathbf{g}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x) + \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)(\mathbf{x} - \boldsymbol{\mu}_0^x) \\ + [\mathbf{I} - \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)](\mathbf{n} - \mathbf{n}_0)\end{aligned} \tag{11}$$

IEEE Proof

where the gradient has the close form of

$$\mathbf{G}(\mathbf{z}) = \mathbf{I} - \mathbf{C}\mathrm{diag}\left(\frac{\mathbf{I}}{\mathbf{I}+\exp[\mathbf{C}^T\mathbf{z}]}\right)\mathbf{C}^T$$

and $\mathrm{diag}()$ above denotes conversion of a vector to a diagonal square matrix, with the components of the vector placed on the diagonal positions of the matrix.

B. Prior Models

While the model (9) expressively represents the relationship among clean speech $\mathbf{x}$, noise $\mathbf{n}$, and (observed) distorted speech $\mathbf{y}$, the main difficulty is that both the clean speech and noise are unobserved. Prior information is thus needed to distinguish the two simultaneous unknowns. We now first establish the statistical model, as the prior information, for the clean speech cepstrum ($\mathbf{x}$ as a random vector) to be a mixture of multivariate Gaussians:

$$p(\mathbf{x}) = \sum_{m=1}^{M} c_m \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m^x, \boldsymbol{\Sigma}_m^x). \tag{12}$$

The speech frames $\mathbf{x}_t$'s are assumed to be independent and identically distributed, and hence t is not denoted in (12). The unobserved missing random variable in (12) is the mixture component m.

In the work described in this paper, the noise cepstrum $\mathbf{n}$ is assumed to be a deterministic (rather than random) vector, which is time varying and is the variable to be estimated for each time frame t.

C. Recursive-EM Algorithm for Noise Estimation

Recursive noise parameter estimation can be shown to be the solution to the following recursive-EM optimization problem [2], [14], [15]

$$\mathbf{n}_{t+1} = \arg\max_{\mathbf{n}} Q_{t+1}(\mathbf{n}) \tag{13}$$

where the objective function $Q_{t+1}(\mathbf{n})$ above is the conditional expectation

$$Q_{t+1}(\mathbf{n}) = E\left[\, ln\, p(\mathbf{y}_1^{t+1}, \mathcal{M}_1^{t+1}|\mathbf{n})|\mathbf{y}_1^{t+1}, \mathbf{n}_1^t\right]. \tag{14}$$

In (14), $\mathcal{M}_1^{t+1} = m_1, m_2, \ldots, m_{t+1}$ is the sequence of (hidden) mixture components in the clean speech model up to time $t+1$, and likewise $\mathbf{y}_1^{t+1} = \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{t+1}$. The expectation in (14) is carried out with respect to the conditional distribution $p(\mathcal{M}_1^{t+1}|\mathbf{y}_1^{t+1}, \mathbf{n}_1^t)$. Note that the objective function of (14) in the recursive-EM algorithm differs from the one in the conventional batch-EM. Q_{t+1} in (14) is time indexed, and the observation sequence is used up to that time, as denoted by $\mathbf{y}_1^{t+1}$.

Appendix, we show that the above E-step objective function can be computed by

$$Q_{t+1}(\mathbf{n}) = \sum_{\tau=1}^{t+1}\sum_{m=1}^{M} \gamma_\tau(m) \cdot ln\, p(\mathbf{y}_\tau \mid m, \mathbf{n}) + Const$$

where $\gamma_\tau(m)$ is the posterior probability as defined in (24) in Appendix. We now incorporate an additional forgetting mechanism into the E-step described so far. This is accomplished by modifying the objective function of Q, via the use of the forgetting factor ϵ, to

$$Q_{t+1}(\mathbf{n}) = \sum_{\tau=1}^{t+1} \epsilon^{t+1-\tau} \sum_{m=1}^{M} \gamma_\tau(m) \cdot ln\, p(\mathbf{y}_\tau|\, m, \mathbf{n}) + Const.$$

This, after using (25) (see Appendix), can be expressed as (after ignoring constant term $Const$ and other terms irrelevant to optimizing noise $\mathbf{n}$)

$$\begin{aligned}\tilde{Q}_{t+1}(\mathbf{n}) =& -\sum_{\tau=1}^{t+1} \epsilon^{t+1-\tau} \sum_{m=1}^{M} \gamma_\tau(m) \big[\mathbf{y}_\tau - \boldsymbol{\mu}_m^y(\mathbf{n}_\tau)\big]^T (\boldsymbol{\Sigma}_m^y)^{-1} \\ & \times \big[\mathbf{y}_\tau - \boldsymbol{\mu}_m^y(\mathbf{n}_\tau)\big] \\ =& \epsilon \cdot \tilde{Q}_t(\mathbf{n}) - R_{t+1}(\mathbf{n})\end{aligned} \tag{15}$$

where

$$\begin{aligned}R_{t+1}(\mathbf{n}) = & \sum_{m=1}^{M} \gamma_{t+1}(m) \big[\mathbf{y}_{t+1} - \boldsymbol{\mu}_m^y(\mathbf{n})\big]^T (\boldsymbol{\Sigma}_m^y)^{-1} \\ & \times \big[\mathbf{y}_{t+1} - \boldsymbol{\mu}_m^y(\mathbf{n})\big].\end{aligned}$$

The value of the forgetting factor ϵ determines the tradeoff between the strength of noise tracking ability (ϵ close to zero) and the reliability of noise estimate (ϵ close to one). The effects of the forgetting factor on speech recognition accuracy have been experimentally studied, which will be presented in Section V.

To carry out the M-step, one can use stochastic approximation [3], [15], [18] to sequentially update the noise parameter. Generalizing from [18] (Theorem 3; pg.264–265, where $\epsilon = 1$) and from [15] (Theorem 3.3, pg.2561, where also $\epsilon = 1$), it can be proved that $\tilde{Q}_{t+1}(\mathbf{n})$ in recursion (15) is maximized via the following recursive form for the noise parameter updating (i.e., recursive M-step)

$$\mathbf{n}_{t+1} = \mathbf{n}_t + \mathbf{K}_{t+1}^{-1}\mathbf{s}_{t+1} \tag{16}$$

where

$$\begin{aligned}\mathbf{s}_{t+1} =& \frac{\partial R_{t+1}}{\partial \mathbf{n}}\,|_{\mathbf{n}=\mathbf{n}_t} \\ =& \sum_{m=1}^{M} \gamma_{t+1}(m)[\mathbf{I} - \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)]^T (\boldsymbol{\Sigma}_m^y)^{-1} \\ & \times \big[\mathbf{y}_{t+1} - \boldsymbol{\mu}_m^y(\mathbf{n})\big]\end{aligned} \tag{17}$$

and

$$\begin{aligned}\mathbf{K}_{t+1} =& -\frac{\partial^2 \tilde{Q}_{t+1}}{\partial^2 \mathbf{n}}\,|_{\mathbf{n}=\mathbf{n}_t} \\ =& -\sum_{\tau=1}^{t+1} \epsilon^{t+1-\tau} \sum_{m=1}^{M} \gamma_\tau(m)[\mathbf{I} - \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)]^T \\ & \times (\boldsymbol{\Sigma}_m^y)^{-1}[\mathbf{I} - \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)].\end{aligned} \tag{18}$$

In (17) and (18), $\mathbf{n}_0$ and $\boldsymbol{\mu}_0^x$ are the operating points (auxiliary parameters) for the truncated Taylor series expansion, and $\boldsymbol{\mu}_m^y(\mathbf{n})$ and $\boldsymbol{\Sigma}_m^y$ are defined in Appendix.

IEEE Proof

In the same way as for (15), we rewrite (18) in a recursive form for efficient computation

$$\mathbf{K}_{t+1} = \epsilon \cdot \mathbf{K}_t + \mathbf{L}_{t+1} \tag{19}$$

where

$$\mathbf{L}_{t+1} = -\sum_{m=1}^{M} \gamma_{t+1}(m)[\mathbf{I} - \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)]^T (\boldsymbol{\Sigma}_m^y)^{-1} \times [\mathbf{I} - \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)]. \tag{20}$$

We note that in some published work (e.g., [2]), (19) was expressed in a different form.[5]

IV. Implementation Using Iterative Stochastic Approximation

Equations (16), (17), and (19) constitute a generic recursive-EM algorithm based on the general principle of stochastic approximation and on the approximate nonlinear model of acoustic environment. It sequentially estimates the noise vector for each frame, $\mathbf{n}_{t+1}$, using the information from its previous frames as well as from the current frame. In this section, we will describe practical considerations for implementing this algorithm, where the key technique of iterative stochastic approximation is introduced.

In (17) and (18), the vectors $\mathbf{n}_0$ and $\boldsymbol{\mu}_0^x$ are the operating points for the truncated Taylor series expansion of the nonlinear environmental model, and need to be appropriately determined. For clean speech, $\mathbf{x}$, the operating points can be set naturally at the most appropriate mean vector in the clean mixture speech model.[6]

To determine $\mathbf{n}_0$, we assume that the noise does not change abruptly, and hence when a new frame, at $(t+1)$, of the observation is entered into the algorithm, the most reasonable noise estimate would be the estimate from the immediately preceding frame t. Therefore, we set the operating point of the truncated Taylor series expansion for the noise at $\mathbf{n}_0 = \mathbf{n}_t$ (or a smoothed version of it) in the evaluation of the $\mathbf{g}$ vector function and $\mathbf{G}$ matrix function in (17), (18), (26) and (27).

A final consideration for improving the effectiveness of the recursive EM algorithm is based on the earlier work that the accuracy of linear approximation to the nonlinear environment model is a key factor in speech enhancement performance, and it is possible to improve the accuracy iteratively by using the enhanced speech [11]. Since the goal of our algorithm is to estimate the noise at the current frame at $(t+1)$ according to (16), the operating point of the Taylor series expansion for noise can be likewise iteratively updated after the estimation is completed at the same time frame $(t+1)$. A smoothed version of the previous frame's estimate $\mathbf{n}_t$ is used to initialize this iteration.[7] This generalizes the stochastic approximation described in Section III into the new "iterative stochastic approximation", within the same recursive-EM framework.

Taking into account all the above implementation considerations, we describe the practical algorithm execution steps below. First, train and fix all parameters in the clean speech model: $c_m, \boldsymbol{\mu}_m^x$, and $\boldsymbol{\Sigma}_m^x$. Then, set $\mathbf{n}_1$ at $t = 1$ to be the average noise vector based on a crude speech-noise detector,[8] and initialize $\mathbf{K}_0 = 0$. For each $t = 2, 3, \ldots, T$ in a noisy utterance $\mathbf{y}_t$, set iteration number $j = 1$ and execute the following steps sequentially (i.e., online).

- Step 1: Compute

$$\gamma_{t+1}^j(m) \equiv p\big(m|\mathbf{y}_{t+1}, \mathbf{n}_t^j\big) = \frac{p\big(\mathbf{y}_{t+1}|m, \mathbf{n}_t^j\big) c_m}{\sum_{m=1}^{M} p\big(\mathbf{y}_{t+1}|m, \mathbf{n}_t^j\big) c_m}$$

  where the likelihood $p\big(\mathbf{y}_{t+1}|m, \mathbf{n}_t^j\big)$ is computed from (25).

- Step 2: Compute

$$\mathbf{s}_{t+1}^j = \sum_{m=1}^{M} \gamma_{t+1}^j(m)[\mathbf{I} - \mathbf{G}(\mathbf{n}_t^j - \boldsymbol{\mu}_m^x)]^T (\boldsymbol{\Sigma}_m^y)^{-1} \times \big[\mathbf{y}_{t+1} - \boldsymbol{\mu}_m^x - \mathbf{g}(\mathbf{n}_t^j - \boldsymbol{\mu}_m^x)\big], \text{ and} \tag{21}$$

$$\mathbf{K}_{t+1}^j = \epsilon \cdot \mathbf{K}_t^j - \sum_{m=1}^{M} \gamma_{t+1}^j(m)[\mathbf{I} - \mathbf{G}(\mathbf{n}_t^j - \boldsymbol{\mu}_0^x)]^T (\boldsymbol{\Sigma}_m^y)^{-1} \times [\mathbf{I} - \mathbf{G}(\mathbf{n}_t^j - \boldsymbol{\mu}_0^x)]. \tag{22}$$

- Step 3: Compute

$$\mathbf{n}_{temp} = \mathbf{n}_t^j + \alpha \cdot [\mathbf{K}_{t+1}^j]^{-1} \mathbf{s}_{t+1}^j. \tag{23}$$

- Step 4: If $j < J$ (total number of iterations), then set $\mathbf{n}_t^{(j+1)} = \mathbf{n}_{temp}$ and increment j by one.[9] Then continue the iteration by going to Step 1. If $j = J$, then increment t by one and start the algorithm again by re-setting $j = 1$ to process the next time frame until the end of the utterance $t = T$.

In (23), α is a heuristic parameter that controls the updating rate for noise estimate. In our implementation, α is set to be inversely proportional to a crude estimate of the noise variance for each separate test utterance. α is also a function of J, the iteration number for each frame.

Several approximations have been made in the implementation of the above algorithm to significantly speed up computation. Among these approximations are: 1) a scalar-version implementation to avoid any matrix inversion; 2) approximation

[5]In [2], that different form is $\mathbf{K}_{t+1} = \epsilon \cdot \mathbf{K}_t + (1-\epsilon)\mathbf{L}_{t+1}$, which clearly deviates from the form of (19) required by the M-step as derived in this section. In our comparative experiments conducted on the AURORA2 task, the rigorous form of (19) always gave superior performance.

[6]In the current work, we have not re-estimated the parameters of this clean speech model, which has been pre-trained using clean speech data and then fixed. Hence, the term including $(\boldsymbol{\mu}_m^x - \boldsymbol{\mu}_0^x)$ in (26) (used in (17)) becomes zero.

[7]This smoothing is found to be critical in achieving high speech recognition performance. We carried out the smoothing by interpolating the previous frame's estimate with some crude noise estimate from speech-free frames determined by a silence detector.

[8]In the AURORA2 experiments we simply used the average of the first 20 frames (which are known to be speech free) of each utterance as the $\mathbf{n}_1$.

[9]After the increment of j, the updated noise estimate $\mathbf{n}_t^{(j+1)}$ becomes the new Taylor series expansion point $\mathbf{n}_t^j$ shown on the right hand side of (21) and (22).

IEEE Proof

TABLE I
COMPARISON OF AURORA2 RECOGNITION RATES (%) FOR THE COMMON, AURORA2-SUPPLIED HMM SYSTEM USING DIFFERENT FRONT-ENDS (AND A DIFFERENT HMM IN SCHEME 5): 1) NOISE-NORMALIZED SPLICE USING A BASELINE NUMERICAL-INTEGRATION METHOD FOR MMSE NOISE ESTIMATION; 2) NOISE-NORMALIZED SPLICE USING THE NEW RECURSIVE-EM METHOD WITH ITERATIVE STOCHASTIC APPROXIMATION; 3) AURORA-SUPPLIED STANDARD MFCCS WITH NO DENOISING; 4) BASIC SPLICE WITH NO NOISE NORMALIZATION; AND 5) SAME AS SCHEME 2 EXCEPT USING A NEW, BETTER DESIGNED HMM SYSTEM

Methods	Training-Mode	Set A	Set B	Set C	Overall
1. Numerical Integration	multi-condition	88.97	87.89	87.80	88.30
	clean-only	85.33	85.75	83.74	85.18
2. **Recursive EM**	multi-condition	91.49	89.16	89.62	**90.18**
	clean-only	87.82	87.09	85.08	**86.98**
3. No Denoising	multi-condition	87.82	86.27	83.78	86.39
	clean-only	61.34	55.75	66.14	60.06
4. No Noise Normalization	multi-condition	91.34	84.98	86.05	87.74
	clean-only	87.56	84.07	81.81	85.01
5. Recursive EM(new HMM)	multi-condition	93.34	91.01	92.15	92.17
	clean-only	88.33	87.75	86.33	87.70

of $\gamma_t(m)$ to be either zero or one for each separate frame t; 3) use of Euclidean distance to determine m_0 that gives a single $\gamma_t(m_0) = 1$; and 4) use of the same m_0 for all within-frame iterations $j \leq J$. These approximations speed up the computation by about a factor of 20, incurring virtually no loss of performance.

V. NOISE-ROBUST SPEECH RECOGNITION

The recursive-EM based noise estimation algorithm described so far has been rigorously evaluated in the AURORA2 task [12]. As outlined in Section II, our basic denoising technique is SPLICE [4], [5], exploiting the availability of stereo data (simultaneously pairwise clean and noisy data) in Set-A of the database. The noise estimate is used in an enhanced, noise-normalized version of SPLICE, which effectively handles mismatched distortion conditions between Set-A and Set-B/C in the AURORA2 task.

A. Baseline Noise-Normalized SPLICE System

A baseline noise estimation method used to evaluate the new algorithm is direct computation of the traditional MMSE noise estimate by numerically carrying out the required integral. The MMSE criterion is similar to that used in [9]. In this baseline system and all other recognition systems (with one exception) described in this section, we use the standard HMM built from HTK as specified and supplied by the AURORA2 task [12].

B. Full Recognition Results On the AURORA2 Task

The numerical integration technique in the baseline system produces noise estimates independently for each noisy speech frame and for each Mel-frequency component. The estimated noise is then used in noise-normalized SPLICE outlined in Section II-B to perform denoising for noise-robust speech recognition. This baseline noise-normalized SPLICE system is used to evaluate the effectiveness of the new recursive-EM noise estimation technique under the otherwise identical experimental conditions.

Comparative recognition results are shown in Table I for the full AURORA2 evaluation test data. Sets-A and -B each consists of 1101 digit sequences for each of four noise conditions and for each of the 0 dB, 5 dB, 10 dB, 15 dB, and 20 dB SNRs. The same is for Set-C except there are only two noise conditions. The recognition rates (%) in Table I are the average values over all the noise conditions and over all the five SNRs. All the results in Table I are obtained with the use of cepstral mean normalization (CMN) for all data after applying noise-normalized SPLICE to MFCC enhancement. The use of CMN has substantially improved the recognition rate for Set-C.[10]

The results of Table I have been presented for four different front-end feature enhancement schemes with the same fixed HMM system provided by AURORA2 plus one scheme with a better designed HMM system. For each scheme, the results for both multi-condition training mode and the clean-only training mode are presented. In the multi-condition mode, denoising is applied to the HMM training set as well as to the test sets (Set-A, B, and C). In the clean-only training mode, only the test set is subject to denoising. From Table I, the new recursive-EM method (Scheme 2) with iterative stochastic approximation performs substantially better than the numerical integration method (Scheme 1) for noise estimation, within the same noise-normalized SPLICE for cepstral enhancement. They are both consistently better than the standard MFCCs supplied by the AURORA2 task using no robust preprocessing to enhance speech features (Scheme 3), and better than the earlier version of SPLICE with no noise normalization from training to test sets (Scheme 4). The word error rate reduction using the

[10]Note that we assumed $\mathbf{h} = 0$ in the recursive-EM for noise estimation. This assumption would not be appropriate for Set-C which contains unknown but fixed channel distortion. This deficiency has been, at least partially, offset by the use of CMN.

IEEE Proof

TABLE II
SPEECH RECOGNITION RATES (%) AS A FUNCTION OF THE FORGETTING FACTOR ϵ (15) USED IN THE RECURSIVE-EM ALGORITHM FOR NOISE ESTIMATION. NOISE CONDITION: STREET NOISE; SNR: 5 DB; SET-B RESULTS FOR CLEAN-ONLY TRAINING (HMMS TRAINED WITH CLEAN SPEECH SPECTRA)

Forgetting Factor ϵ	0	0.1	0.2	0.3	0.5	0.8	1.0	No-Denoising
Recog. Accuracy (%)	80.10	80.80	81.65	81.40	81.47	79.08	78.00	38.45

new recursive-EM method is 27.9% for the multicondition training mode, and 67.4% for the clean-only training mode, compared with the results with standard MFCCs with no enhancement. The final row of Table I (Scheme 5) shows the results obtained by using the same noise-normalized SPLICE cepstral enhancement as for Scheme 2 (i.e., recursive EM with iterative stochastic approximation) but using a new, better designed HMM. The relative error rate reduction increases significantly to 48.40% and 76.86%, respectively, for the two training modes. All the results shown in Table I are based on a total of $1101 \times 10 \times 5 = 55\,050$ test utterances for each of the multicondition and clean-only training modes.

C. Effects of the Forgetting Mechanism on Recognition Rate

In this subsection, we provide evidence for the importance of striking a balance between the noise tracking ability and the estimation reliability associated with the recursive-EM algorithm described in Section III. The "forgetting" mechanism implemented by the use of the ϵ parameter in (15) is responsible for the noise tracking ability. In Table II are shown the AURORA2 recognition rates (Set-B with $\mathrm{SNR} = 5$ dB, under the "Street" noise condition, which is nonstationary, and with the clean-only training mode) as a function of the value of ϵ.

We first observed in Table II that for the full range of ϵ, the denoised features using the recursive-EM performs much better than the mismatched case with no denoising (only 38% recognition accuracy). Within the full range of ϵ, the accuracy varies with an about 20% difference in the word error rate. At the one extreme where $\epsilon = 0$, (15) is reduced to

$$Q_{t+1}(\mathbf{n}_{t+1}) = -\sum_{m=1}^{M} \gamma_{t+1}(m)\big[\mathbf{y}_{t+1} - \boldsymbol{\mu}_m^y(\mathbf{n}_{t+1})\big]^T \times (\boldsymbol{\Sigma}_m^y)^{-1}\big[\mathbf{y}_{t+1} - \boldsymbol{\mu}_m^y(\mathbf{n}_{t+1})\big].$$

Thus, the M-step[11] in the EM-algorithm uses only the observation in the current single time frame $\mathbf{y}_{t+1}$ to estimate the noise $\mathbf{n}_{t+1}$ at the same time frame $t+1$. This does not explore the possible temporal coherence between the current and the previous frames of the noise, and is expected to produce a noise estimate not as reliable as compared with the noise estimate when the previous noisy observations are taken into account. Indeed, in Table II, we found that the recognition accuracy for $\epsilon = 0$ is lower than that by most cases of $\epsilon > 0$, which make use of all previous noisy observations in estimating the noise at a given time frame.

Let us analyze the other extreme where $\epsilon = 1$. Equation (15) now becomes

$$Q_{t+1}(\mathbf{n}_{t+1}) = -\sum_{\tau=1}^{t+1}\sum_{m=1}^{M} \gamma_\tau(m)\big[\mathbf{y}_\tau - \boldsymbol{\mu}_m^y(\mathbf{n}_\tau)\big]^T(\boldsymbol{\Sigma}_m^y)^{-1} \times \big[\mathbf{y}_\tau - \boldsymbol{\mu}_m^y(\mathbf{n}_\tau)\big].$$

It is clear that all the noisy observation frames up to the current $(t+1)$ frame, $\mathbf{y}_1^{t+1}$, are used for estimating the current-frame noise vector $\mathbf{n}_{t+1}$. This should take into account the temporal coherence of the noise for enhancing the reliability of the noise estimate.[12] However, all the current $(t+1)$ and its previous frames have the same contribution to the estimate of the current noise vector $\mathbf{n}_{t+1}$. That is, no forgetting mechanism is used to place a greater emphasis on the more recent data than the more distant data in the past. This strategy would be appropriate (and indeed ideal[13]) for the stationary noise, but not for the nonstationary noise present in most of the AURORA2 data. For the nonstationary noise, different observation data segments correspond to different noise parameter values. Hence, it is highly desirable to adaptively track the changing noise parameters by incorporating a forgetting mechanism. Indeed, from Table II, after implementing a simple forgetting mechanism by setting appropriate values of $\epsilon > 0$, greater recognition performance is achieved.

We found that there is a delicate balance between the noise tracking ability and the estimation reliability in the recursive-EM algorithm we have developed. This is represented by the value of the forgetting factor ϵ, chosen empirically in this work. Research on automatic optimization of the forgetting factor appeared recently in the literature [2], which, unfortunately, does not seem to have produced convincingly positive results in speech recognition.[14]

Comparing our results in Table II with those in [2], similar trends emerge while different noisy speech databases are used. Sensitivity of the recognition rate to the values of the forgetting factor is similar. The range of the variation in the recognition rate for the fixed SNR (stationary white noise corruption) in [2] is smaller than ours, and that for the variable SNR is greater than ours. This is expected since in the AURORA2 data, the "fixed" SNR is computed over the entire utterance with a variable instantaneous SNR over time frames.

[11]When $\epsilon = 0$, the M-step can be carried out much more efficiently by solving $\mathbf{s}_{t+1} = 0$ without the need to involve the recursion in $\mathbf{K}_{t+1}$.

[12]In a limiting case, this would approach the estimation performance provided by batch estimation algorithms.

[13]in the sense that it would maximize the estimation reliability for time-invariant parameters.

[14]In Table II of [2], the use of the automatically optimized forgetting factor only reduces the relative recognition error by a small fraction of 3%, at the expense of introducing another empirical parameter ϵ^+, which still need to be tuned.

IEEE PROOF

TABLE III
SPEECH RECOGNITION RATES (%) AS A FUNCTION OF THE WITHIN-FRAME ITERATION NUMBER USED IN THE RECURSIVE-EM ALGORITHM WITH ITERATIVE STOCHASTIC APPROXIMATION FOR NOISE ESTIMATION. NOISE CONDITION: STREET NOISE; RESULTS ARE AVERAGED OVER SNRS OF 0 DB, 5 DB, 10 DB, 15 DB, AND 20 DB; SET-C RESULTS FOR MULTICONDITION TRAINING (HMMS TRAINED WITH DENOISED SPEECH CEPSTRA)

Iteration number J	1	2	4	8	No-Denoising
Recog. Accuracy (%)	85.00	85.87	87.44	88.10	84.31

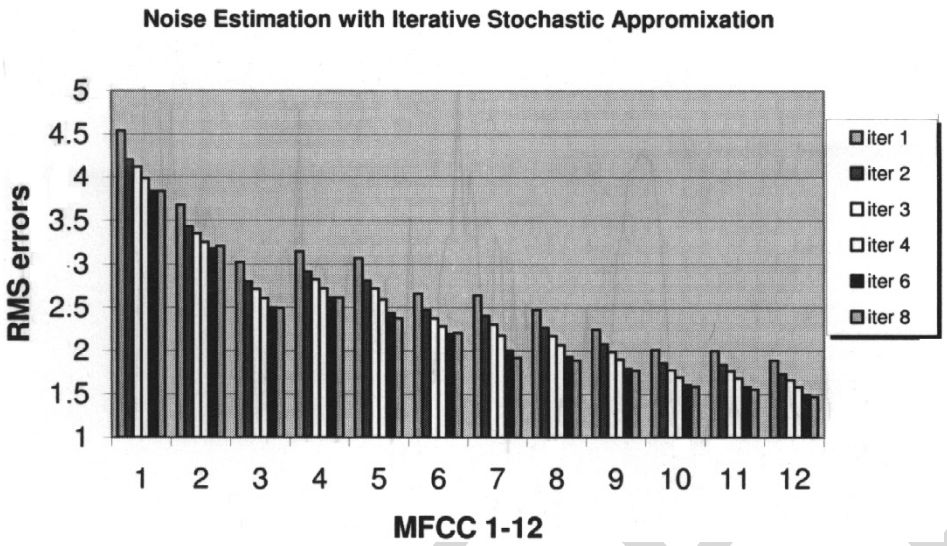

Fig. 1. RMS errors between the estimated noise from noisy speech and true noise. RMS errors are averaged over a large number of files in Set-B under the "Restaurant" noise condition with $\mathrm{SNR} = 0$ dB. The errors are plotted as a function of the MFCC coefficients and of the number of iterations J in the iterative stochastic approximation technique.

D. Effects of Iterations in Stochastic Approximation on Recognition Rate

In this subsection, we provide evidence for the importance of using iterations in stochastic approximation when implementing the recursive-EM algorithm for noise estimation. While we gave theoretical motivations for using iterations in Section IV, we show empirical evidence here now. In Table III are shown the AURORA2 speech recognition rates (Set-C, averaged over the results for $\mathrm{SNR} = 0$ dB, 5 dB, 10 dB, 15 dB, and 20 dB, and under the "Street" noise condition with the multicondition training mode) as a function of the iteration number J in iterative stochastic approximation. Substantial improvement of recognition rates is achieved as the iteration number J increases from one to eight. We note that the "Street" noise used here is not one of the four noises used in training the SPLICE codebooks and correction vectors. This demonstrates that the novel use of iterations in stochastic approximation is highly effective in bridging the mismatched SPLICE training and deployment conditions. The technique of iterative stochastic approximation is also able to improve the prior art in using stochastic approximation, which would correspond to using a single within-frame iteration $J = 1$.

E. Analysis of Noise Estimation and Speech Enhancement

In this subsection, we provide empirical analysis for and examples of the noise estimate obtained from the new algorithm based on iterative stochastic approximation. We also show corresponding examples of enhanced speech after applying the noise-normalized SPLICE procedure making use of the noise estimate.

One analysis we have performed examines how close the estimated noise is to the true noise that was used to corrupt the clean speech in creating the AURORA2 data. Fig. 1 shows the Root Mean Square (RMS) errors computed between the estimated noise and true noise. The noise is estimated using the recursive-EM with iterative stochastic approximation from noisy speech data. The RMS errors are averaged over all the time frames in a large number of files in Set-B under the "Restaurant" noise condition with $\mathrm{SNR} = 0$ dB. The RMS errors are shown for each of the within-frame iterations in iterative stochastic approximation up to eight iterations, and for each of the MFCCs from the first order to the 12th order. The results in Fig. 1 reveal several interesting aspects of the algorithm. First, the RMS errors are, in general, decreasing as a function of the iteration number. However, occasionally, the error decrease is not strictly monotonic; see MFCC-2 results in Fig. 1. This is well understood due to the general nature of stochastic approximation [3], which does not follow the exact EM property. Second, the RMS errors are reduced most significantly from iteration one to iteration two, compared with the error reductions resulting from the subsequent iterations. Third, the drop of the RMS errors tends to saturate for all the MFCCs at about the same iteration number; eight iterations for the results of Fig. 1. Fourth, the RMS errors tend to be lower for higher-order MFCCs, with the exception seen for the MFCC-3 and MFCC-4 in Fig. 1.

Since the RMS errors shown in Fig. 1 are averaged over all time frames, they tend to hide the nonstationary nature of the

IEEE Proof

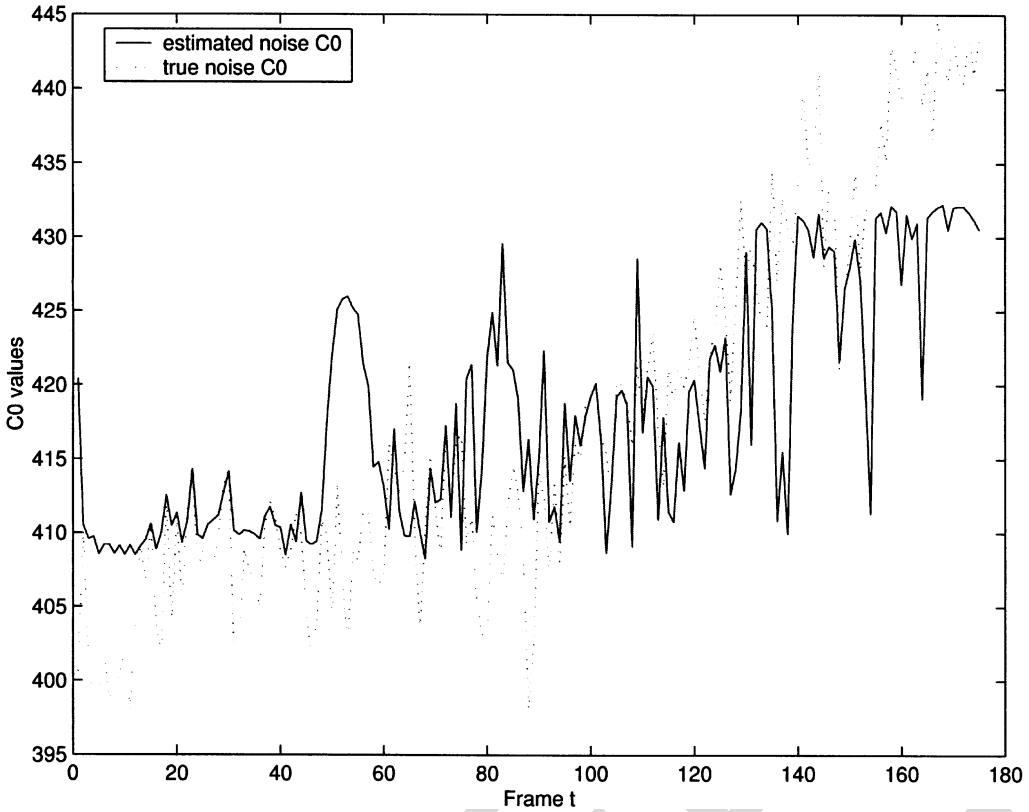

Fig. 2. C0 sequences for the estimated noise (solid) and true noise (dashed). The noisy utterance from which the noise is estimated has the SNR of 5 dB.

noise and of its estimate. To show such nonstationarity, we plot in Figs. 2 and 3 several detailed traces of the MFCCs for the estimated (solid) and true (dotted) noises for one utterance (with $\mathrm{SNR} = 5\ \mathrm{dB}$). From these traces, we observed that the algorithm indeed tracks the changing noise quite closely.

When the MFCC 0–5 shown in Figs. 2 and 3 (and the remaining MFCC 6–12 not shown) are combined as the input to the inverse cosine transform, we recover the Mel-frequency log spectra for this utterance. This is plotted in Fig. 4 in the format of spectrogram. The top panel shows the reference spectrogram of the noisy speech, from which the noise estimate is obtained as plotted in the bottom panel. The noise estimate is generally rather close to the true noise, whose spectrogram is shown in the middle panel. Only during frames of around 50–60 and around 80–90, some speech energies leak through to be part of the incorrect noise estimates. Such incorrect noise estimates can also be seen in the C0 plot in Fig. 2 for the same utterance at around the same frames.

After the estimated noise in Fig. 4 (bottom panel) is used in noise-normalized SPLICE for cepstral enhancement, the resulting denoised (i.e., enhanced) speech is plotted in Fig. 5 (bottom panel) in the same spectrogram format. Compared with the clean speech shown also in Fig. 5 (middle panel), the enhanced speech smears itself slightly during the frames where the noise estimates are relatively poor. For all other frames, the corrupting noises have been effectively removed.

VI. Summary and Conclusions

A novel algorithm for recursive estimation of parameters in a nonlinear model involving incomplete data is presented in this paper. The algorithm is applied specifically to time-varying deterministic parameters of additive noise in a nonlinear model that accounts for the generation of the cepstral data of noisy speech from the cepstral data of the noise and clean speech. For the nonstationary noise that we encounter in robust speech recognition, different observation data segments correspond to different noise parameter values. Hence, recursive estimation algorithms are more desirable than batch algorithms, since they can be designed to adaptively track the changing noise parameters. One such design based on the novel technique of iterative stochastic approximation in the recursive-EM framework is presented and evaluated in this work. We provide the mathematical basis for this new technique in detail, and report a study on the sensitivity of the new noise estimation algorithm's performance to the forgetting factor, the use of which constitutes the essence of any online-adaptive, recursive technique. The forgetting factor is embedded in our noise estimation algorithm, aimed to equip the algorithm with a desired balance of the tracking ability of nonstationary noise and of the noise estimation reliability.

The proposed recursive noise estimation algorithm tracks the time-varying noise parameters while iteratively optimizing the auxiliary parameters employed to piecewise linearly approximate a nonlinear generative model for the observed noisy speech. The accuracy of approximation is shown to improve progressively with more iterations. The key idea of using

IEEE Proof

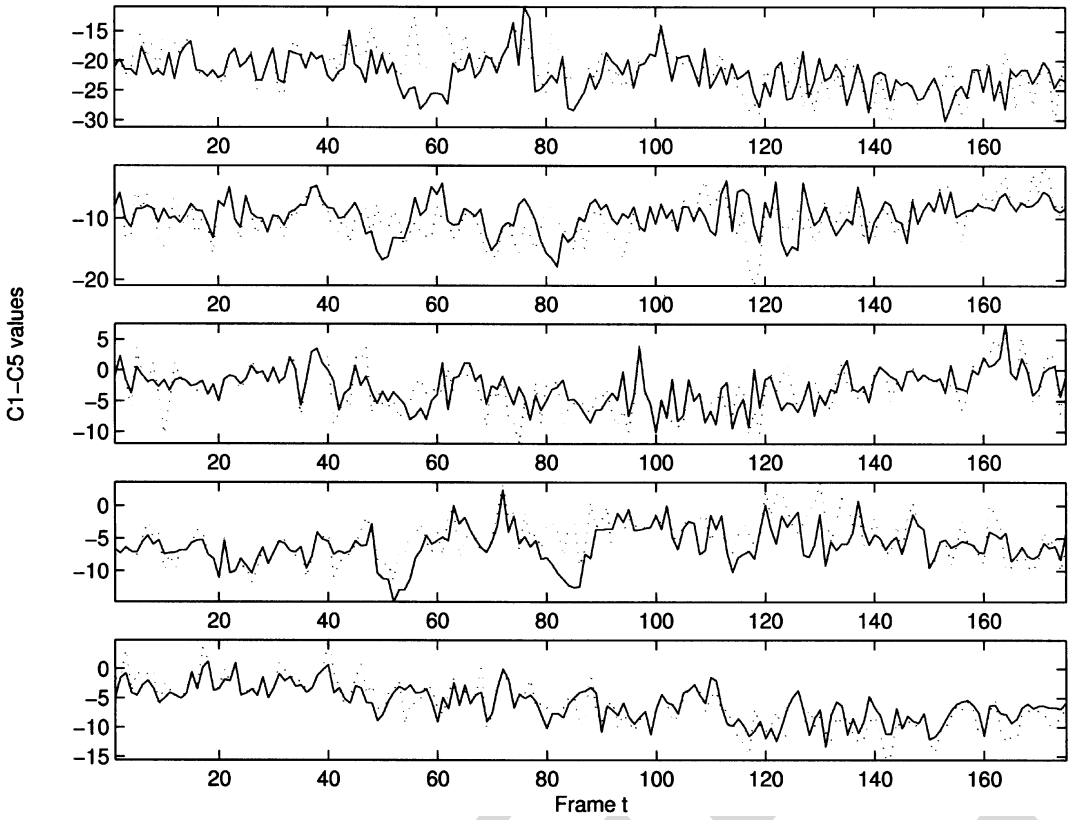

Fig. 3. Estimated noise (solid) and true noise (dashed) MFCC sequences. From top to bottom: C1, C2, C3, C4, and C5. The same utterance as in Fig. 2.

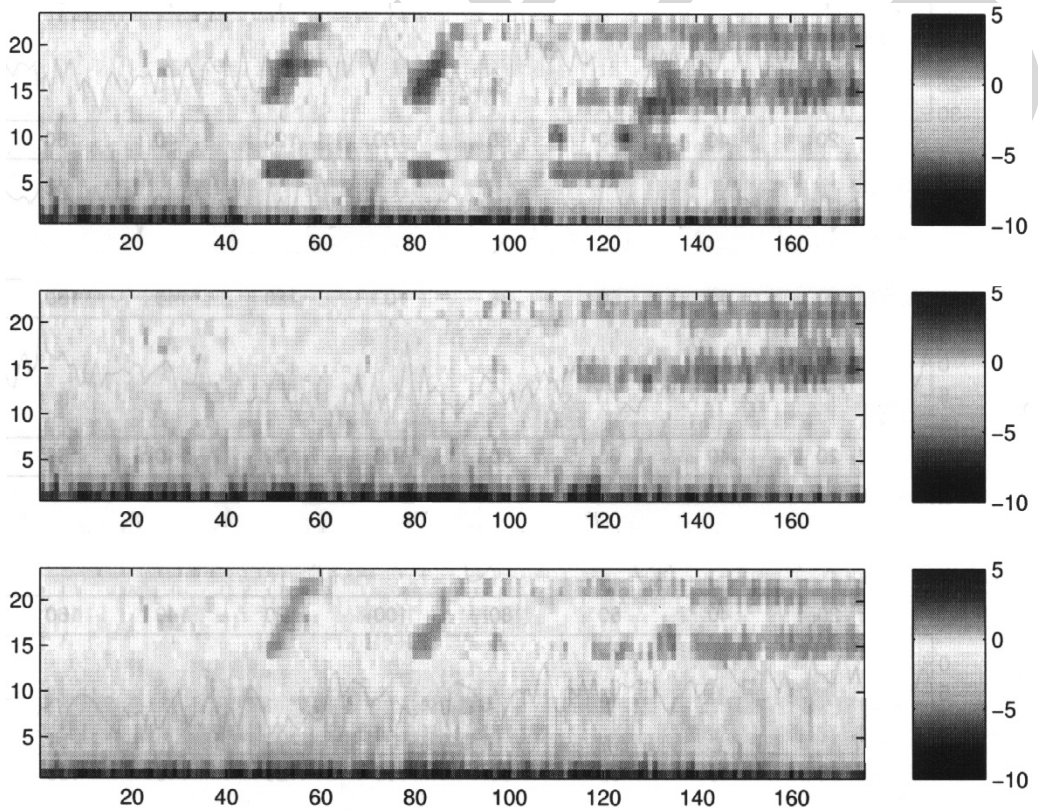

Fig. 4. Comparing Mel-spectra of estimated and true noise. From top to bottom: noisy speech ($\mathrm{SNR} = 5$ dB), true noise, and estimated noise. The same utterance as in Figs. 2 and 3.

iterations to improve estimation algorithms involving nonlin- earity originated from the early work of "iterated extended

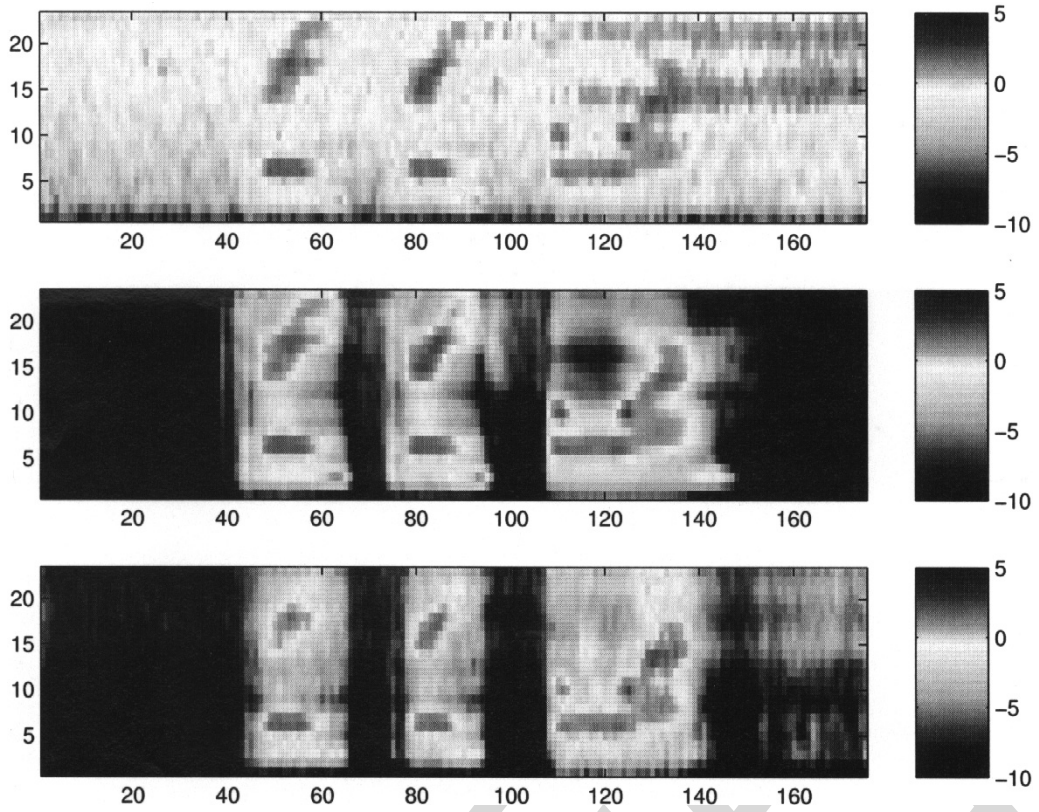

Fig. 5. Comparing Mel-spectra of enhanced speech (after noise-normalized SPLICE) and true speech. From top to bottom: noisy speech (SNR = 5 dB), clean speech, and enhanced speech. The same utterance as in Figs. 2–4.

Kalman filter" [13], [16], and has been used in recent speech recognition research dealing with various forms of nonlinearity in speech production and in acoustic environments [6], [11]. One main contribution of the work described in this paper is successful integration of this idea into the recursive-EM framework, which in combination produces significant results in the robust speech recognition practice. This integration gives rise to the new technique of iterative stochastic approximation presented in detail in Section IV.

While the focus of this paper has been on noise estimation, one key application of the new noise estimate presented is to provide the normalization term for MFCC enhancement in the SPLICE framework for noise-robust speech recognition. (Other possible applications of the new noise estimate, such as perceptual enhancement of speech, have not been dealt with in this paper.) The noise-normalized SPLICE framework is outlined in the earlier part of the paper (Section II) to provide the reader with the right context before describing details of the noise estimation algorithm.

The full speech recognition results for the AURORA2 task are presented that demonstrated the effectiveness of the new recursive noise estimation algorithm in comparison with a more traditional, MMSE noise estimation method under otherwise identical experimental conditions. Future work will extend the algorithm to represent the noise as time-varying random vectors in order to exploit the variance parameter and new prior information. The algorithm will also be extended to include more complex speech models that incorporate dynamic features, and to include more accurate environment models that capture more detail properties of acoustic distortion than the model presented in Section III-A of this paper.

APPENDIX

In this Appendix, we outline the intermediate steps that simplify the objective function $Q_{t+1}(\mathbf{n})$ of (14) into a form that can be easily subject to the M-step optimization. The simplification steps are

$$\begin{aligned}
Q_{t+1}(\mathbf{n}) =& E\big[ln\, p(\mathbf{y}_1^{t+1}|\mathcal{M}_1^{t+1},\mathbf{n}) + ln\, p(\mathcal{M}_1^{t+1}) \\
& \times \mid \mathbf{y}_1^{t+1},\mathbf{n}_1^t\big] \\
=& E\big[\, ln\, p(\mathbf{y}_1^{t+1}|\mathcal{M}_1^{t+1},\mathbf{n}) \mid \mathbf{y}_1^{t+1},\mathbf{n}_1^t\big] + Const. \\
=& \sum_{\tau=1}^{t+1} E\big[\, ln\, p(\mathbf{y}_\tau|m_\tau,\mathbf{n}) \mid \mathbf{y}_1^{t+1},\mathbf{n}_1^t\big] + Const. \\
=& \sum_{\tau=1}^{t+1} E\big[(\sum_{m=1}^{M} ln\, p(\mathbf{y}_\tau|m,\mathbf{n})\delta_{m_\tau,m}) \mid \mathbf{y}_1^{t+1},\mathbf{n}_1^t\big] \\
& + Const. \\
=& \sum_{\tau=1}^{t+1}\sum_{m=1}^{M} E\big[\delta_{m_\tau,m} \mid \mathbf{y}_1^{t+1},\mathbf{n}_1^t\big] ln\, p(\mathbf{y}_\tau|m,\mathbf{n}) \\
& + Const. \\
=& \sum_{\tau=1}^{t+1}\sum_{m=1}^{M} \gamma_\tau(m)\cdot ln\, p(\mathbf{y}_\tau \mid m,\mathbf{n}) + Const.
\end{aligned}$$

where $Const$ is a constant term (independent of noise $\mathbf{n}$ to be estimated), $\delta_{m_\tau,m}$ is the Kronecker delta function (taking values

IEEE Proof

of one if $m_\tau = m$, or zero otherwise), $\gamma_\tau(m)$ is shown below to be the "occupancy" (posterior) probability

$$\gamma_\tau(m) \equiv E\big[\delta_{m_\tau,m} \mid \mathbf{y}_1^{t+1}, \mathbf{n}_1^t\big] = \sum_{\mathcal{M}_1^{t+1}} p(\mathcal{M}_1^{t+1}|\mathbf{y}_1^{t+1}, \mathbf{n}_1^t)\delta_{m_\tau,m} = p\big(m|\mathbf{y}_\tau, \mathbf{n}_{\tau-1}\big).$$

The above $\gamma_\tau(m)$ can be computed using Bayes rule

$$\gamma_\tau(m) = \frac{p\big(\mathbf{y}_\tau|m, \mathbf{n}_{\tau-1}\big)c_m}{\sum_{m=1}^{M} p\big(\mathbf{y}_\tau|m, \mathbf{n}_{\tau-1}\big)c_m} \tag{24}$$

where the likelihood $p\big(\mathbf{y}_\tau|m, \mathbf{n}_{\tau-1}\big)$ is computed using the linearized version of the nonlinear acoustic environment model of (11). This gives

$$p(\mathbf{y}_\tau|m, \mathbf{n}_{\tau-1}) = \mathcal{N}\big[\mathbf{y}_\tau; \boldsymbol{\mu}_m^y(\mathbf{n}_{\tau-1}), \boldsymbol{\Sigma}_m^y\big] \tag{25}$$

where the mean for the given mixture component m is

$$\boldsymbol{\mu}_m^y(\mathbf{n}_{\tau-1}) = \boldsymbol{\mu}_m^x + \mathbf{g}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x) + \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)(\boldsymbol{\mu}_m^x - \boldsymbol{\mu}_0^x) + [\mathbf{I} - \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)](\mathbf{n}_{\tau-1} - \mathbf{n}_0) \tag{26}$$

and the corresponding covariance matrix is

$$\boldsymbol{\Sigma}_m^y = [\mathbf{I} + \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)]\boldsymbol{\Sigma}_m^x[\mathbf{I} + \mathbf{G}^T(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)]^T. \tag{27}$$

Note that (27) becomes clear after rewriting (11) into

$$\mathbf{y} = [\mathbf{I} + \mathbf{G}(\mathbf{n}_0 - \boldsymbol{\mu}_0^x)]\mathbf{x} + \mathbf{d}$$

where $\mathbf{d}$ is a deterministic term not affecting the form of the covariance matrix.

In (26) and (27), $\mathbf{n}_0$ is the operating point for noise in the Taylor series expansion, serving as the auxiliary parameter that will be iteratively optimized with the noise parameter $\mathbf{n}_\tau$ (see Section IV). ($\boldsymbol{\mu}_0^x$ in (26) and (27) is the operating point for clean speech in the Taylor series expansion, which is chosen from one of the mean vectors in the prior mixture-of-Gaussian speech model that best accounts for the observed noisy speech vector.) Given $\mathbf{n}_0$ and given the previously updated noise parameter $\mathbf{n}_{\tau-1}$, $\gamma_\tau(m)$ is computed using Bayes rule (24) with the likelihood computed by (25).

ACKNOWLEDGMENT

The authors thank the reviewers for constructive suggestions which significantly improved the presentation of this paper. They also thank H. Attias for useful discussions and for proofreading the manuscript.

REFERENCES

[1] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proc. ICSLP*, vol. 3, 2000, pp. 869–872.

[2] M. Afify and O. Siohan, "Sequential noise estimation with optimal forgetting for robust speech recognition," in *Proc. ICASSP*, vol. 1, 2001, pp. 229–232.

[3] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations — Applications of Mathematics*. New York: Springer, 1990, vol. 22.

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

[5] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. D. Hung, "High-performance robust speech recognition using stereo training data," in *Proc. ICASSP*, vol. 1, 2001, pp. 301–304.

[6] L. Deng and J. Ma, "Spontaneous speech recognition using a statistical coarticulatory model for the hidden vocal-tract-resonance dynamics," *J. Acoust. Soc. Amer.*, vol. 108, no. 6, pp. 3036–3048, Dec. 2000.

[7] J. Droppo, L. Deng, and A. Acero, "Efficient on-line acoustic environment estimation for FCDCN in a continuous speech recognition system," in *Proc. ICASSP*, vol. 1, April 2001, pp. 209–212.

[8] ——, "Evaluation of the SPLICE algorithm on the AURORA2 database," in *Proc. Eurospeech*, vol. 1, Sept. 2001, pp. 217–220.

[9] Y. Ephraim, "Statistical-model-based speech enhancement systems," *Proc. IEEE*, vol. 80, pp. 1526–1555, Oct. 1992.

[10] L. Frenkel and M. Feder, "Recursive expectation-maximization (EM) algorithms for time-varying parameters with applications to multiple target tracking," in *IEEE Trans. Signal Processing*, vol. 42, 1999, pp. 306–320.

[11] B. Frey, L. Deng, A. Acero, and T. Kristjansson, "ALGONQUIN: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition," in *Proc. Eurospeech*, vol. 2, Sept. 2001, pp. 901–904.

[12] H. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR2000 on Automatic Speech Recognition: Challenges for the Next Millennium*, Paris, France, Sept. 2000.

[13] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic, 1970.

[14] N. S. Kim, "Nonstationary environment compensation based on sequential estimation," *IEEE Signal Processing Lett.*, vol. 5, pp. 57–60, 1998.

[15] V. Krishnamurthy and J. B. Moore, "Online estimation of hidden markov model parameters based on the Kullback-Leibler information mature," *IEEE Trans. Signal Processing*, vol. 41, pp. 2557–2573, 1993.

[16] J. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications and Control (Lesson 24)*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[17] P. Moreno, B. Raj, and R. Stern, "A vector taylor series approach for environment-independent speech recognition," in *Proc. ICASSP*, vol. 1, 1996, pp. 733–736.

[18] D. M. Titterington, "Recursive parameter estimation using incomplete data," *J. R. Statist. Soc. B*, vol. 46, pp. 257–267, 1984.

[19] O. Viikki, *Speech Commun. (Special Issue on Noise Robust ASR)*, O. Viikki, Ed., 2001, vol. 34.

[20] E. Weinstein, M. Feder, and A. Oppenheim, "Sequential algorithms for parameter estimation based on Kullback-Leibler information measure," *IEEE Trans. Signal Processing*, vol. 38, pp. 1652–1654, 1990.

Li Deng received the B.S. degree from University of Science and Technology of China in 1982, the M.S. degree from the University of Wisconsin-Madison in 1984, and the Ph.D. degree from the University of Wisconsin-Madison in 1986.

He worked on large-vocabulary automatic speech recognition in Montreal, QC, Canada, from 1986 to 1989. In 1989, he joined Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, as Assistant Professor; he became Full Professor in 1996. From 1992 to 1993, he conducted sabbatical research at Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, and from 1997 to 1998, at ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan. In 1999, he joined Microsoft Research, Redmond, WA, as Senior Researcher, and is currently a Principal Investigator in the DARPA-EARS Program and Affiliate Professor of electrical engineering at University of Washington, Seattle. His research interests include acoustic–phonetic modeling of speech, speech and speaker recognition, speech synthesis and enhancement, speech production and perception, auditory speech processing, noise robust speech processing, statistical methods and machine learning, nonlinear signal processing, spoken language systems, multimedia signal processing, and multimodal human–computer interaction. In these areas, he has published over 200 technical papers and book chapters, and has given keynote, tutorial, and other invited lectures. He recently completed the book *Speech Processing—A Dynamic and Optimization-Oriented Approach* (New York: Marcel Dekker, 2003).

Dr. Deng served on Education Committee and Speech Processing Technical Committee of the IEEE Signal Processing Society during 1996–2000, and is currently serving as Associate Editor for the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING.

IEEE Proof

Jasha Droppo, photograph and biography not available at time of publication. Please e-mail plain text (ASCII) copy.

Alex Acero, photograph and biography not available at time of publication. Please e-mail plain text (ASCII) copy.

IEEE Proof