# Noise Adaptive Training for Robust Automatic Speech Recognition

Ozlem Kalinli, *Member, IEEE*, Michael L. Seltzer, *Senior Member, IEEE*, Jasha Droppo, *Senior Member, IEEE*, and Alex Acero, *Fellow, IEEE*

***Abstract*—In traditional methods for noise robust automatic speech recognition, the acoustic models are typically trained using clean speech or using multi-condition data that is processed by the same feature enhancement algorithm expected to be used in decoding. In this paper, we propose a noise adaptive training (NAT) algorithm that can be applied to all training data that normalizes the environmental distortion as part of the model training. In contrast to feature enhancement methods, NAT estimates the underlying "pseudo-clean" model parameters directly without relying on point estimates of the clean speech features as an intermediate step. The pseudo-clean model parameters learned with NAT are later used with vector Taylor series (VTS) model adaptation for decoding noisy utterances at test time. Experiments performed on the Aurora 2 and Aurora 3 tasks demonstrate that the proposed NAT method obtain relative improvements of 18.83% and 32.02%, respectively, over VTS model adaptation.**

***Index Terms*—Model adaptation, noise adaptive training, robust speech recognition, vector Taylor series (VTS).**

## I. Introduction

DESPITE years of research, automatic speech recognition (ASR) in noisy environments remains a challenging problem since there are many possible types of environmental distortion, and it is difficult to compensate for all of these distortions accurately. The primary reason for poor recognition performance in noise is the mismatch between training and test conditions. Many methods have been proposed in the literature to reduce this mismatch and improve performance. These methods can be grouped under two main categories: feature enhancement methods and model adaptation methods.

Feature enhancement techniques operate by denoising the feature vectors received at test time so that they better match the recognizer's acoustic models, typically trained from clean speech. These methods are attractive because they are typically simpler computationally than model domain techniques and can be implemented independently from the recognizer. Some techniques operate at the spectrum level, e.g., spectral subtraction [1], while others, such as Cepstral-MMSE [2] operate on the features directly. There is also a class of techniques that use a prior speech model, typically in the form of a Gaussian mixture model (GMM) to aid the enhancement process. For example, SPLICE uses stereo recordings of clean and noisy speech to learn a piecewise linear mapping from noisy to clean speech using a Gaussian mixture model (GMM) [3]. While front-end methods have shown improved performance on several tasks, they all, by definition, make point-estimates of the clean speech features. Errors in these estimates can cause further mismatch between the features and the acoustic model, resulting in degraded performance.

Model adaptation techniques avoid this problem by compensating the probability distributions of the recognizer directly. Several model adaptation techniques have been proposed in the literature. Some, such as MLLR [4] and MAP adaptation [5], are data driven methods that do not make any assumptions about the nature of the corrupting process. In situations where there is limited adaptation data, reduced-parameter methods such as CMLLR [6], [7] and Regularized FMLLR [8] have been proposed.

While these methods can improve recognition accuracy in noisy conditions [9], [10], better performance is generally obtained by methods that exploit the known relationship between clean and noisy speech, such as parallel model combination (PMC) [11] and vector Taylor series (VTS) adaptation [12]–[15]. These methods are generally more complex than generic adaptation techniques but require very little adaptation data. In fact, most only need a reliable estimate of the noise and channel distributions. In [15], VTS adaptation produced state of the art performance on the Aurora 2 task for maximum likelihood systems that do not use discriminative training.

In spite of such success, methods like this have two significant drawbacks: 1) they require acoustic models trained from clean data, which means performance will be suboptimal for tasks for which such data does not exist, and 2) the adaptation algorithms make approximations that may cause residual mismatch between the adapted models and the observed data, e.g., in VTS, a truncated Taylor series expansion is used to approximate the relationship between clean and noisy speech in the cepstral domain and the inverse DCT used to convert the features to the log mel filterbank domain is a pseudo-inverse.

In this paper, we propose a new training algorithm called noise adaptive training (NAT) to overcome these two problems in the context of the VTS adaptation algorithm in [15]. It is motivated by the success of the feature-based noise adaptive training

Manuscript received February 03, 2009; revised November 25, 2009. Date of publication March 29, 2010; date of current version September 01, 2010. This work was done while O. Kalinli was an intern at Microsoft Research, Redmond, WA 98052 USA. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mark Hasegawa-Johnson.

O. Kalinli is with the R&D Group, Sony Computer Entertainment of America, Foster City, CA 94404 USA (e-mail: ozlem_kalinli@playstation.sony.com).

M. L. Seltzer, J. Droppo, and A. Acero are with the Microsoft Research, Redmond, WA 98052 USA (e-mail: mseltzer@microsoft.com; jdroppo@microsoft.com; alexac@microsoft.com).

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

Digital Object Identifier 10.1109/TASL.2010.2040522

algorithm in [3], the speaker adaptive training (SAT) algorithm in [16], and the more recently proposed adaptive training algorithms for noisy speech recognition in [17] and [18].

The proposed NAT algorithm uses multi-condition data and transforms a multi-condition trained model into a "pseudo-clean" model to reduce the environment specific variations. NAT integrates this environmental distortion normalization into HMM training using a new formulation of the expectation–maximization (EM) algorithm that incorporates the identical VTS approximation used by the model adaptation. The pseudo-clean model parameters learned by NAT are later used in conjunction with VTS adaptation for decoding noisy utterances at runtime. As an analogy, the proposed NAT algorithm has the same relationship to VTS model adaptation as SAT has to MLLR adaptation. This relationship will be discussed in more detail in Section IV.

The NAT algorithm is closely related to two other recently published adaptive training algorithms. Like NAT, these methods also seek to learn a model from multi-style training data that is better matched to a model adaptation scheme designed for environmental robustness. The first technique is based on the idea of "irrelevant variability normalization" (IVN) [17] and uses the VTS model adaptation in [13] as the basis for its approach. The second technique, called Joint adaptive training (JAT), was proposed in [18] as a companion training scheme to a recently proposed model adaptation scheme called Joint Uncertainty Decoding (JUD) [19]. JUD performs adaptation to a noisy environment using a set of regression classes, in a manner similar to multi-class MLLR, rather than adapting each Gaussian individually, as is typically done in VTS or PMC.

Some salient aspects of the proposed NAT method are listed as follows.

- In contrast to the feature-based adaptive training algorithm in [3], NAT jointly estimates the underlying pseudo-clean model parameters and the environmental distortion parameters without relying on a point estimate of the clean speech features.
- The NAT algorithm uses multi-condition data for acoustic model training whereas the standard model adaptation techniques such as PMC and VTS require clean trained models.
- In NAT, the same VTS scheme used for adaptation in the recognition stage is applied in HMM training, which further reduces the mismatch between training and testing.
- NAT estimates static, delta, and delta-delta model parameters during training while IVN estimates only the static parameters. Also, IVN is based on the VTS approach in [13], while NAT is based on the approach in [15]. As a result, two schemes use slightly different auxiliary functions.
- JAT performs adaptation using regression classes whereas NAT adapts each Gaussian individually. In JAT, the means and variances are updated jointly using an iterative gradient-based approach in the M-step. In NAT, an iterative approach is only required for estimating the variances. This distinction is also true for the distortion model parameters (noise and channel).

The rest of the paper is organized as follows. In Section II, we review HMM adaptation using a VTS approximation. The proposed NAT algorithm for VTS model adaptation is detailed in Section III. In Section IV, we compare and contrast the proposed NAT algorithm to SAT, IVN, and JAT. We then describe a series of experiments that illustrate the performance of NAT in Section V and finally, offer some concluding remarks in Section VI.

## II. HMM Adaptation Using VTS

We assume an acoustic environment in which clean speech is corrupted by stationary additive noise and linear filtering. In the cepstral domain, the relationship between clean and distorted speech can be expressed as [20]

$$\mathbf{y} = \mathbf{x} + \mathbf{h} + \mathbf{g}(\mathbf{n} - \mathbf{x} - \mathbf{h}) \tag{1}$$

where $\mathbf{y}$, $\mathbf{x}$, $\mathbf{h}$, $\mathbf{n}$ are the cepstrum vectors corresponding to distorted speech, clean speech, channel, and noise, respectively. In (1), the nonlinear function $\mathbf{g}(\mathbf{z})$ is

$$\mathbf{g}(\mathbf{z}) = \mathbf{C}\log(1 + \exp(\mathbf{C}^{\dagger}(\mathbf{z})) \tag{2}$$

where $\mathbf{C}$ is the discrete cosine transform (DCT) matrix and $\mathbf{C}^{\dagger}$ is its pseudo-inverse.

It can be shown that the Jacobian of (1) with respect to $\mathbf{x}$ and $\mathbf{h}$ evaluated at a fixed point $(\boldsymbol{\mu}_{\mathbf{x},0}, \boldsymbol{\mu}_{\mathbf{h},0}, \boldsymbol{\mu}_{\mathbf{n},0})$ is

$$\mathbf{G} = \mathbf{C} \cdot \mathrm{diag}\left(\frac{1}{1 + \exp(\mathbf{C}^{\dagger}(\boldsymbol{\mu}_{\mathbf{n},0} - \boldsymbol{\mu}_{\mathbf{x},0} - \boldsymbol{\mu}_{\mathbf{h},0}))}\right) \cdot \mathbf{C}^{\dagger} \tag{3}$$

where $\mathrm{diag}(.)$ represents the diagonal matrix whose elements equal to the value of the vector in the argument. Similarly, the Jacobian of (1) with respect to $\mathbf{n}$ can be expressed as $\mathbf{F} = \mathbf{I} - \mathbf{G}$. Then, the nonlinear relationship between the distorted speech, clean speech and environment parameters (noise and channel) in (1) can be approximated by using a first order VTS expansion around the point $(\boldsymbol{\mu}_{\mathbf{x},0}, \boldsymbol{\mu}_{\mathbf{h},0}, \boldsymbol{\mu}_{\mathbf{n},0})$ as

$$\begin{aligned}\mathbf{y} \approx\ & \boldsymbol{\mu}_{\mathbf{x},0} + \boldsymbol{\mu}_{\mathbf{h},0} + \mathbf{g}_0 \\ & + \mathbf{G}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x},0}) + \mathbf{G}(\mathbf{h} - \boldsymbol{\mu}_{\mathbf{h},0}) + \mathbf{F}(\mathbf{n} - \boldsymbol{\mu}_{\mathbf{n},0})\end{aligned} \tag{4}$$

where

$$\mathbf{g}_0 = \mathbf{C}\log(1 + \exp(\mathbf{C}^{\dagger}(\boldsymbol{\mu}_{\mathbf{n},0} - \boldsymbol{\mu}_{\mathbf{x},0} - \boldsymbol{\mu}_{\mathbf{h},0}))). \tag{5}$$

The goal of the traditional VTS model adaptation, e.g., [13]–[15], is to adapt the parameters of the HMM trained using clean data to the environment conditions of a test utterance. Let $\Lambda_X = \{\boldsymbol{\mu}_{sm}, \boldsymbol{\Sigma}_{sm}\}$ denote the set of Gaussian parameters for the clean speech HMMs where $\boldsymbol{\mu}_{sm}$ and $\boldsymbol{\Sigma}_{sm}$ denote the mean vector and the diagonal covariance matrix of the $m$th Gaussian component in the $s$th state, respectively. We assume that additive noise is Gaussian with mean $\boldsymbol{\mu}_{\mathbf{n}}$ and covariance $\boldsymbol{\Sigma}_{\mathbf{n}}$, and that the channel $\mathbf{h}$ has a probability density of the Kronecker delta function $\delta(\mathbf{h} - \boldsymbol{\mu}_{\mathbf{h}})$.

It is assumed that the environment distortion does not change the alignment between speech frame and the corresponding Gaussian component of the HMM. As a result, only the mean vector and covariance matrix for each Gaussian of the HMM will be affected. Under the VTS approximation, $\mathbf{y}$ is a linear

function of $\mathbf{x}$, $\mathbf{n}$, and $\mathbf{h}$. Therefore, the mean vector $\boldsymbol{\nu}_{sm}$ of the adapted model $\Lambda_Y = \{\boldsymbol{\nu}_{sm}, \bar{\boldsymbol{\Psi}}_{sm}\}$ can be estimated by taking the expected value of the terms in (4), as follows:

$$\boldsymbol{\nu}_{sm} \approx \boldsymbol{\mu}_{sm,0} + \boldsymbol{\mu}_{\mathbf{h},0} + \mathbf{g}_{sm,0} + \mathbf{G}_{sm}(\boldsymbol{\mu}_{sm} - \boldsymbol{\mu}_{sm,0}) + \mathbf{G}_{sm}(\boldsymbol{\mu}_{\mathbf{h}} - \boldsymbol{\mu}_{\mathbf{h},0}) + \mathbf{F}_{sm}(\boldsymbol{\mu}_{\mathbf{n}} - \boldsymbol{\mu}_{\mathbf{n},0}). \quad (6)$$

Similarly the adapted covariance matrix can be estimated with

$$\boldsymbol{\Psi}_{sm} \approx \mathbf{G}_{sm}\boldsymbol{\Sigma}_{sm}\mathbf{G}_{sm}^T + \mathbf{F}_{sm}\boldsymbol{\Sigma}_{\mathbf{n}}\mathbf{F}_{sm}^T. \quad (7)$$

In (6) and (7), $\mathbf{G}$ $\mathbf{F}$ and $\mathbf{g}_0$ are functions of $\boldsymbol{\mu}_{sm,0}$, the mean of the $m$th Gaussian in the $s$th state of the clean-speech HMM, and hence carry subscript of $sm$. It can be concluded from (7) that even if $\boldsymbol{\Sigma}_{sm}$ and $\boldsymbol{\Sigma}_{\mathbf{n}}$ are diagonal, $\boldsymbol{\Psi}_{sm}$ is no longer diagonal. However, $\bar{\boldsymbol{\Psi}}_{sm}$ is assumed to be diagonal so that it can be used with a traditional ASR decoder that has been optimized for diagonal covariance matrices.

To compute the means and variances of the delta features, we use the continuous-time approximation proposed in [21]. This results in the following mean adaptation formula

$$\boldsymbol{\nu}_{\Delta sm} \approx \mathbf{G}_{sm}\boldsymbol{\mu}_{\Delta sm} \quad (8)$$

where we assume that the noise is stationary, hence $\boldsymbol{\mu}_{\Delta\mathbf{n}} = 0$, for all utterances. Similarly, the covariance matrices for the delta features are adapted according to

$$\boldsymbol{\Psi}_{\Delta sm} \approx \mathbf{G}_{sm}\boldsymbol{\Sigma}_{\Delta sm}\mathbf{G}_{sm}^T + \mathbf{F}_{sm}\boldsymbol{\Sigma}_{\Delta\mathbf{n}}\mathbf{F}_{sm}^T. \quad (9)$$

The means and covariance matrices of the delta-delta features are computed in a similar way to (8) and (9) by replacing delta ($\Delta$) parameters with delta-delta ($\Delta^2$) parameters.

To adapt the clean-speech model parameters using (6)–(9), we need environment distortion (noise and channel) parameters. These parameters are hidden variables; hence, they are estimated for each test utterance using an iterative EM [22] algorithm using the VTS approximation. After model adaptation, the utterance is re-decoded with the new models. In this paper, we use the VTS adaptation algorithm in [15] as the basis of our approach. This implementation adapts the means and variances of the static, delta, and delta-delta parameters using a generalized EM approach.

The traditional VTS model adaptation requires the original HMM be trained from clean speech; otherwise the generative model used for adaptation is not valid. This prevents the use of multi-condition data for acoustic model training. This problem is solved by the proposed NAT algorithm which is discussed in the next section.

## III. Noise Adaptive Training

Let us assume that there are $I$ utterances in the multi-condition training set $\mathcal{Y} = \{Y^{(i)}\}_{i=1}^I$, and $Y^{(i)}$ is a sequence of $T_i$ observations corresponding to $i$th utterance. In traditional ML HMM training, the parameters are estimated such that the resulting generic model $\Lambda_Y$ maximizes the likelihood of the multi-condition training data.

In NAT, we assume that each utterance in the training set has an associated distortion model $\phi^{(i)} = \{\boldsymbol{\mu}_{\mathbf{n}}^{(i)}, \boldsymbol{\Sigma}_{\mathbf{n}}^{(i)}, \boldsymbol{\mu}_{\mathbf{h}}^{(i)}\}$ that describes the additive noise and the channel. The NAT algorithm seeks to find the distortion model parameters for all utterances $\Phi = \{\phi^{(i)}\}_{i=1}^I$, and the underlying "pseudo-clean" model parameters $\Lambda_X$ that jointly maximize the likelihood of the multi-condition data when the model $\Lambda_X$ is transformed to the adapted HMM of $\Lambda_Y^{(i)}$. This can be written in the ML sense as

$$(\Lambda_X, \Phi) = \arg\max_{(\overline{\Lambda}_X, \bar{\Phi})} \prod_{i=1}^{I} \mathcal{L}(Y^{(i)}; \Lambda_Y^{(i)}) \quad (10)$$

where

$$\Lambda_Y^{(i)} = VTS(\overline{\Lambda}_X, \bar{\phi}^{(i)}) \quad (11)$$

is the adapted HMM using the VTS approach (6)–(9) as detailed in Section II. In (10), $(\overline{\Phi},\ \overline{\Lambda}_X)$ and $(\Phi,\ \Lambda_X)$ are the old and new parameters set, respectively. The term "pseudo-clean" is used to indicate that the model defined by $\Lambda_X$ is not necessarily equivalent to models trained with clean speech, but rather the model that maximizes the likelihood of the multi-condition training data when processed by the same VTS adaptation scheme that will be used at runtime. In NAT, we use a new EM algorithm that learns the distortion model parameters and the pseudo-clean speech model parameters iteratively. Thus, we start with the following EM auxiliary function:

$$Q(\Phi, \Lambda_X, \bar{\Phi}, \bar{\Lambda}_X) = \sum_{i=1}^{I} \sum_{t,s,m} \gamma_{tsm}^{(i)} \log(p(\mathbf{y}_t^{(i)} | s, m, \Lambda_X, \Phi)) \quad (12)$$

where $i$ is utterance index, $\sum_{t,s,m}$ represents summation over frames, states, and Gaussians, and $\gamma_{tsm}^{(i)}$ is the posterior probability of the $m$th Gaussian in the $s$th state of the HMM for frame $t$ of the $i$th utterance

$$\gamma_{tsm}^{(i)} = p(s_t = s, m_t = m | Y^{(i)}, \bar{\Lambda}_X, \bar{\Phi}) \quad (13)$$

and computed as

$$\gamma_{tsm}^{(i)} = \frac{\alpha_{ts}^{(i)}\beta_{ts}^{(i)}}{\sum_{s'} \alpha_{ts'}^{(i)}\beta_{ts'}^{(i)}} \frac{c_{sm} p(\mathbf{y}_t^{(i)} | s, m, \bar{\Lambda}_X, \bar{\Phi})}{\sum_{m'} c_{sm'} p(\mathbf{y}_t^{(i)} | s, m', \bar{\Lambda}_X, \bar{\Phi})} \quad (14)$$

where $\alpha_{ts}^{(i)}$ and $\beta_{ts}^{(i)}$ are the conventional forward and backward variables used in Baum–Welch training algorithm [23], $c_{sm}$ is the mixture weight of the $m$th Gaussian in state $s$. In (12) and (14)

$$p(\mathbf{y}_t^{(i)} | s, m, \bar{\Lambda}_X, \bar{\Phi}) \sim \mathcal{N}(\mathbf{y}_t^{(i)}; \boldsymbol{\nu}_{sm}^{(i)}, \boldsymbol{\Psi}_{sm}^{(i)}) \quad (15)$$

where $\boldsymbol{\nu}_{sm}^{(i)}$, $\boldsymbol{\Psi}_{sm}^{(i)}$ are computed using VTS model adaptation as in (6)–(7), and they are actually utterance-dependent since they are functions of distortion parameters for that utterance $\phi^{(i)}$. Similarly, $\mathbf{G}_{sm}$, $\mathbf{F}_{sm}$, and $\mathbf{g}_{sm,0}$ terms in (6)–(9) are also computed for each utterance since they are functions of distortion parameters $\phi^{(i)}$ (ref. Equation (3), (5)), hence they carry the superscript $(i)$ in the remainder of the paper.

To update the noise mean in the M-step of the EM algorithm, we take the derivative of the $Q$ function with respect to

$\boldsymbol{\mu}_{\mathbf{n}}^{(i)}$ and set the result to zero. After following the derivation in Appendix A, we can express the update formula for noise mean as

$$\boldsymbol{\mu}_{\mathbf{n}}^{(i)} = \boldsymbol{\mu}_{\mathbf{n},0}^{(i)} + \left\{ \sum_{t,s,m} \gamma_{tsm}^{(i)} (\mathbf{F}_{sm}^{(i)})^T (\boldsymbol{\Psi}_{sm}^{(i)})^{-1} \mathbf{F}_{sm}^{(i)} \right\}^{-1} \times \left\{ \sum_{t,s,m} \gamma_{tsm}^{(i)} (\mathbf{F}_{sm}^{(i)})^T (\boldsymbol{\Psi}_{sm}^{(i)})^{-1} (\mathbf{y}_t^{(i)} - \boldsymbol{\nu}_{sm,0}^{(i)}) \right\} \quad (16)$$

where $\boldsymbol{\nu}_{sm,0}^{(i)}$ is the value of $\boldsymbol{\nu}_{sm}^{(i)}$ at the VTS expansion point $\boldsymbol{\mu}_{sm,0} = \boldsymbol{\mu}_{sm}$, $\boldsymbol{\mu}_{\mathbf{n},0}^{(i)} = \boldsymbol{\mu}_{\mathbf{n}}^{(i)}$, $\boldsymbol{\mu}_{\mathbf{h},0}^{(i)} = \boldsymbol{\mu}_{\mathbf{h}}^{(i)}$, and it is equal to

$$\boldsymbol{\nu}_{sm,0}^{(i)} \triangleq \boldsymbol{\mu}_{sm,0} + \boldsymbol{\mu}_{\mathbf{h},0}^{(i)} + \mathbf{g}_{sm,0}^{(i)}. \quad (17)$$

The channel mean is also found in a similar way by taking the derivative of the $Q$ function with respect to $\boldsymbol{\mu}_{\mathbf{h}}^{(i)}$ and setting the result equal to zero. Then, the following update formula is obtained for the channel mean:

$$\boldsymbol{\mu}_{\mathbf{h}}^{(i)} = \boldsymbol{\mu}_{\mathbf{h},0}^{(i)} + \left\{ \sum_{t,s,m} \gamma_{tsm}^{(i)} (\mathbf{G}_{sm}^{(i)})^T (\boldsymbol{\Psi}_{sm}^{(i)})^{-1} \mathbf{G}_{sm}^{(i)} \right\}^{-1} \times \left\{ \sum_{t,s,m} \gamma_{tsm}^{(i)} (\mathbf{G}_{sm}^{(i)})^T (\boldsymbol{\Psi}_{sm}^{(i)})^{-1} (\mathbf{y}_t^{(i)} - \boldsymbol{\nu}_{sm,0}^{(i)}) \right\}. \quad (18)$$

It is assumed that the noise is stationary; hence, means corresponding to the delta and delta-delta features are assumed to be zero i.e., $\boldsymbol{\mu}_{\Delta\mathbf{n}}^{(i)} = 0$, $\boldsymbol{\mu}_{\Delta^2\mathbf{n}}^{(i)} = 0$ for all utterances.

There is no closed-form solution for the noise covariance matrices, so they are optimized iteratively using Newton's method according to the following update equation:

$$\boldsymbol{\Sigma}_{\mathbf{n}}^{(i)} = \boldsymbol{\Sigma}_{\mathbf{n},0}^{(i)} - \left[ \left( \frac{\partial^2 Q}{\partial^2 \boldsymbol{\Sigma}_{\mathbf{n}}^{(i)}} \right)^{-1} \left( \frac{\partial Q}{\partial \boldsymbol{\Sigma}_{\mathbf{n}}^{(i)}} \right) \right]_{\boldsymbol{\Sigma}_{\mathbf{n}}^{(i)} = \boldsymbol{\Sigma}_{\mathbf{n},0}^{(i)}}. \quad (19)$$

The derivation for the terms in (19) is shown in Appendix A. The noise covariance matrices for dynamic features $\boldsymbol{\Sigma}_{\Delta\mathbf{n}}^{(i)}, \boldsymbol{\Sigma}_{\Delta^2\mathbf{n}}^{(i)}$, are computed in a similar way to (19) by replacing the static parameters and features with dynamic parameters and features. It is assumed that noise covariance matrices $\boldsymbol{\Sigma}_{\mathbf{n}}^{(i)}, \boldsymbol{\Sigma}_{\Delta\mathbf{n}}^{(i)}, \boldsymbol{\Sigma}_{\Delta^2\mathbf{n}}^{(i)}$ are all diagonal.

The pseudo-clean model parameters $\Lambda_X$ are updated in a similar way to the distortion parameters except that they are computed based on all utterances. We take the derivative of the $Q$ function with respect to $\boldsymbol{\mu}_{sm}$ for static features and set the result to zero. After following the derivation in Appendix B, the update formula for model mean $\boldsymbol{\mu}_{sm}$ is computed as

$$\boldsymbol{\mu}_{sm} = \boldsymbol{\mu}_{sm,0} + \left\{ \sum_{i,t} \gamma_{tsm}^{(i)} (\mathbf{G}_{sm}^{(i)})^T (\boldsymbol{\Psi}_{sm}^{(i)})^{-1} \mathbf{G}_{sm}^{(i)} \right\}^{-1} \times \left\{ \sum_{i,t} \gamma_{tsm}^{(i)} (\mathbf{G}_{sm}^{(i)})^T (\boldsymbol{\Psi}_{sm}^{(i)})^{-1} \left( \mathbf{y}_t^{(i)} - \boldsymbol{\nu}_{sm,0}^{(i)} \right) \right\}. \quad (20)$$

Equation (20) is only applied to update the mean parameters corresponding to static features. For the delta portions of features, the mean update formula is shown in (21) at the bottom of the page. The update equation for the delta-delta mean values is the same as (21) substituting $\Delta^2$ parameters for the $\Delta$ parameters.

As with the noise covariance, there is no closed-form solution for computing the covariance matrices of the HMM distributions. In a similar way as we did for $\boldsymbol{\Sigma}_{\mathbf{n}}$, Newton's method is used to estimate them iteratively as follows:

$$\boldsymbol{\Sigma}_{sm} = \boldsymbol{\Sigma}_{sm,0} - \left[ \left( \frac{\partial^2 Q}{\partial^2 \boldsymbol{\Sigma}_{sm}} \right)^{-1} \left( \frac{\partial Q}{\partial \boldsymbol{\Sigma}_{sm}} \right) \right]_{\boldsymbol{\Sigma}_{sm} = \boldsymbol{\Sigma}_{sm,0}}. \quad (22)$$

The derivation for the terms in (22) is shown in Appendix B. The covariance matrices for dynamic features of the pseudo-clean model $\boldsymbol{\Sigma}_{\Delta sm}, \boldsymbol{\Sigma}_{\Delta^2 sm}$ are computed in a similar way to (22) by replacing the static parameters and features with the dynamic parameters and features.

The transition probabilities, the initial probabilities, and the mixture weights for the pseudo-clean model are computed in the same way as traditional ML training of the HMMs but using the new posterior probability as defined in (13). The NAT algorithm is summarized in the next section.

### A. NAT Algorithm

A block diagram of the noise adaptive training algorithm is shown in Fig. 1. Using multi-condition data, an HMM is trained using the conventional Baum–Welch algorithm to initialize $\bar{\Lambda}_X$. The distortion parameters for each utterance ($\bar{\phi}^{(i)}$ for $i = 1, \ldots, I$) are initialized such that the channel mean is set to zero and the noise mean and covariance are estimated from the first and last $N$ frames (non-speech frames) of the utterance. The distortion parameters for each utterance $\phi^{(i)}$ are kept in a separate file. After the model and distortion parameters are initialized, NAT training is performed iteratively as shown in Fig. 1.

NAT first updates the environment distortion parameters $\Phi^{(i)}$ given the old model parameters $\bar{\Lambda}_X$. Then, the new distortion parameters are used to accumulate the sufficient statistics for

$$\boldsymbol{\mu}_{\Delta sm} = \boldsymbol{\mu}_{\Delta sm,0} + \left\{ \sum_{i,t} \gamma_{tsm}^{(i)} (\mathbf{G}_{sm}^{(i)})^T (\boldsymbol{\Psi}_{\Delta sm}^{(i)})^{-1} \mathbf{G}_{sm}^{(i)} \right\}^{-1} \times \left\{ \sum_{i,t} \gamma_{tsm}^{(i)} (\mathbf{G}_{sm}^{(i)})^T (\boldsymbol{\Psi}_{\Delta sm}^{(i)})^{-1} \left( \Delta \mathbf{y}_t^{(i)} - \mathbf{G}_{sm}^{(i)} \boldsymbol{\mu}_{\Delta sm,0} \right) \right\}. \quad (21)$$

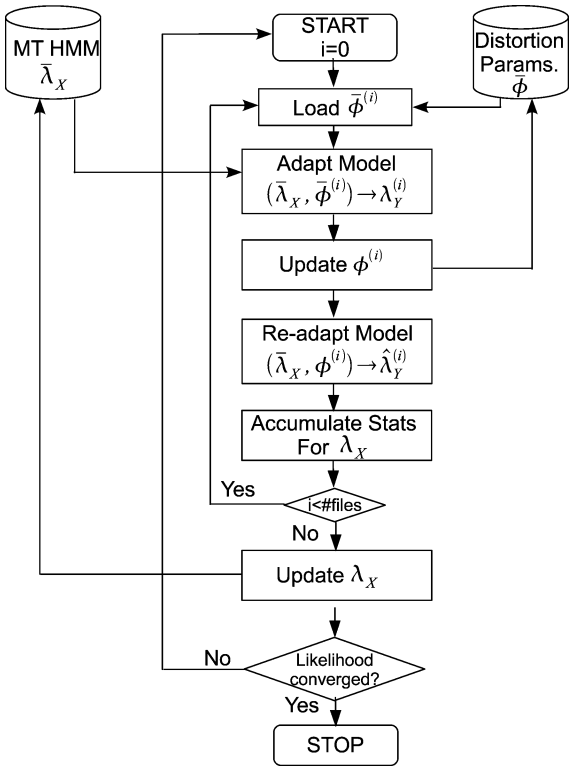



Fig. 1. Flowchart of the NAT training. $\Lambda_X$ denotes pseudo-clean model and initialized with HMM trained using multi-condition data. $\phi^{(i)}$ and $\Lambda_Y^{(i)}$ denote distortion parameters and VTS adapted model for the $i$th utterance, respectively.

the model parameter estimation. After all the utterances are processed and the sufficient statistics are accumulated, the model parameters are updated. This whole process is considered as a single iteration. Multiple iterations of this whole process are performed until the likelihood converges. Once the pseudo-clean model parameters are learned, the distortion parameters $\Phi$ are discarded and the HMM parameters $\Lambda_X$ are ready to be used with VTS adaptation at test time.

### *B. Implementation Details*

The model variances are optimized iteratively using Newton's method as given in (22) since there is no closed form solution. To ensure that the variance remains positive, a common trick is used such that the variance is transformed as $\tilde{\boldsymbol{\Sigma}}_{sm} = \log(\boldsymbol{\Sigma}_{sm})$. After estimating $\tilde{\boldsymbol{\Sigma}}_{sm}$, the exponential function is applied to obtain the actual covariance $\boldsymbol{\Sigma}_{sm} = \exp(\tilde{\boldsymbol{\Sigma}}_{sm})$. Hence, the derivatives in (22) are actually computed with respect to $\tilde{\boldsymbol{\Sigma}}_{sm}$ as given in Appendix B.

There are some well-known numerical issues with Newtons method. If the Hessian matrix is close to singular, its inverse may be unstable. Also, to ensure that the updates converge to a local maximum, the Hessian matrix must be negative definite. A diagonal loading technique [24] is used to fulfill these constraints as

$$\tilde{\boldsymbol{\Sigma}}_{sm} = \tilde{\boldsymbol{\Sigma}}_{sm,0} - \left[\left(\frac{\partial^2 Q}{\partial^2 \tilde{\boldsymbol{\Sigma}}_{sm}} - \varepsilon I\right)^{-1}\left(\frac{\partial Q}{\partial \tilde{\boldsymbol{\Sigma}}_{sm}}\right)\right]_{\tilde{\boldsymbol{\Sigma}}_{sm}=\tilde{\boldsymbol{\Sigma}}_{sm,0}} \tag{23}$$

where $\varepsilon = 1$ was empirically found to be useful to stabilize the optimization. Also, to ensure the stability, a similar approach to [18] is used such that the change of variance was limited as

$$\tilde{\boldsymbol{\Sigma}}_{sm} = \min\left(\max\left(\tilde{\boldsymbol{\Sigma}}_{sm}, \tilde{\boldsymbol{\Sigma}}_{sm,0} - \varsigma\right), \tilde{\boldsymbol{\Sigma}}_{sm,0} + \varsigma\right) \tag{24}$$

which in turn limits the change of the original variance $\boldsymbol{\Sigma}_{sm}$ by a factor of $\exp(\varsigma)$. In the experiments, $\varsigma$ was set to 1. The noise covariance matrix was also optimized iteratively in the same way.

One drawback of our implementation is that the diagonal loading parameter is not optimized, which can hinder the rate of convergence. One possible solution is to choose the smallest $\varepsilon$ that will make the Hessian matrix negative definite and well conditioned. This can be done via eigenvalue decomposition of the Hessian by choosing $\varepsilon$ to be slightly larger than the most positive eigenvalue. An alternative to this is to use a diagonal matrix instead of $\varepsilon I$ in (23) in which only the elements corresponding to positive eigenvalues of the Hessian matrix are replaced by a small constant value.

## IV. Discussion

In this section, we compare NAT to several related algorithms, beginning with SAT [16]. The problem formulation of NAT and SAT are quite similar with the following main difference: the SAT algorithm searches for a compact model $\Lambda_c$ that will maximize the expected likelihood of the data from multiple speakers after performing MLLR transformation on $\Lambda_c$, whereas the NAT algorithm seeks the pseudo-clean model $\Lambda_x$ that will maximize the expected likelihood of the multi-condition data after VTS adaptation. The variances are not updated in the SAT; hence, we only focus on the comparison of the mean update equations here. The mean adaptation formula given in (6) can be written in the form of MLLR transformation as follows:

$$\boldsymbol{\nu}_{sm}^{(i)} = W_{sm}^{(i)} * \boldsymbol{\mu}_{sm} + \beta_{sm}^{(i)} \tag{25}$$

where

$$W_{sm}^{(i)} = \mathbf{G}_{sm}^{(i)} \tag{26}$$

and

$$\beta_{sm}^{(i)} = \boldsymbol{\mu}_{sm,0} + \boldsymbol{\mu}_{\mathbf{h},0}^{(i)} + \mathbf{g}_{sm,0}^{(i)} - \mathbf{G}_{sm}^{(i)}\boldsymbol{\mu}_{sm,0} \tag{27}$$

when the VTS expansion point is $\boldsymbol{\mu}_{\mathbf{h},0}^{(i)} = \boldsymbol{\mu}_{\mathbf{h}}^{(i)}$ and $\boldsymbol{\mu}_{\mathbf{n},0}^{(i)} = \boldsymbol{\mu}_{\mathbf{n}}^{(i)}$. Then, the model mean update equations for the SAT and NAT algorithms are in the same form of

$$\boldsymbol{\mu}_{sm} = \left\{\sum_{i=1}^{I}\sum_{t=1}^{T_i}\gamma_{tsm}^{(i)}(W_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}W_{sm}^{(i)}\right\}^{-1} \times \left\{\sum_{i=1}^{I}\sum_{t=1}^{T_i}\gamma_{tsm}^{(i)}(W_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{y}_t^{(i)} - \beta_{sm}^{(i)})\right\} \tag{28}$$

with the following key exception: whereas SAT utilizes an unconstrained transformation matrix per speaker, NAT uses a matrix $W_{sm}^{(i)}$ that is specific to each Gaussian that is highly constrained by the utterance-specific distortion parameters.

We now compare NAT to the other two adaptive training techniques described in Section I, IVN [17] and JAT [18]. Both IVN and NAT have the same goal of creating a pseudo-clean model from multi-condition training data that is suitable for VTS adaptation at runtime. The most significant difference between NAT and IVN is that in IVN, only the HMM model parameters for the static coefficients are optimized using VTS-based environmental normalization during training. The update formulae for the delta and delta-delta parameters are the same as conventional maximum likelihood training, except that the posterior computation in the E-step is altered to account for the adaptation of the static parameters.

There is a second, more subtle difference between IVN and NAT that is rooted in the specific implementation of VTS used by each algorithm. IVN is based on the VTS adaptation algorithm proposed in [13], while NAT uses VTS adaptation as performed in [14] and [15]. In [13] (and thus in IVN), the log probability of the complete data used in the EM auxiliary function includes the observed noisy speech and hidden variables for clean speech, channel, noise and model component index (cf. Equation (12) in [13]). In contrast, in the VTS approach we followed, these variables are first marginalized out of the complete data distribution and thus the only remaining hidden variables in the auxiliary function are the state and Gaussian component, as shown in (12). Training with this objective function generates parameters which maximize the likelihood of the noisy training data against the adapted HMMs, regardless of whether or not the model parameters learned represent the true distributions of the hidden variables they represent, e.g., clean speech, noise, and channel. We note, however, that by using the auxiliary function in [13], IVN does have the advantage of closed-form update equations for both the means and variances, whereas an iterative approach is required for the variances in NAT.

The JAT algorithm also shares the goal of IVN and NAT to create of a more compact "canonical" model from noisy training data that is appropriate for model adaptation during decoding. As described in Section I, JAT is the companion training algorithm to JUD [19]. JUD operates by explicitly modeling the joint probability of clean and noisy speech in order to transform clean speech models to the current noise conditions. Unless stereo data is available, this joint distribution is obtained using an existing model adaptation scheme, e.g., VTS or DPMC. What differentiates JUD from methods like VTS and PMC is that JUD computes the transformations for a small set of regression classes rather than for each Gaussian individually. This results in a significant computational savings, typically at a small decrease in performance. JAT is the training procedure that estimates the set of canonical HMM parameters for a given JUD setup, i.e., a particular adaptation algorithm and set of regression classes. Thus, the key difference between JAT and NAT is the same as the difference between JUD and VTS adaptation, namely the presence or absence of regression classes.

In JAT [18], the mean and variance parameters are concatenated into a supervector, and updated jointly using a generalized EM approach. In this case, there is no closed-form solution to update the model parameters; hence both the means and the variances are updated iteratively using Newton's method. As a result, the Hessian matrix used in the second-order update has terms that represent the second derivative of the auxiliary function with respect to the mean and variance components as well as heterogeneous terms that involve the mean and variance components. However, in NAT, the means and variances are updated sequentially, i.e., the means are updated assuming the variances are fixed and vice-versa. This enables a closed-form solution for the mean because under the VTS approximation, the adapted mean is a linear function of the mean of the "pseudo-clean" Gaussian. Computing the mean in closed form in the M-step is more efficient and potentially more accurate than using a gradient-based approach. The variances in NAT are updated using Newton's method because a closed form solution is still not possible under the VTS model. Thus, the Hessian matrix used in NAT is simpler than the one used in JAT since it only contains terms related to the variance. This distinction between the mean and variance updates between NAT and JAT applies to both the HMM parameters and the environmental distortion parameters.

It was noted in [25] that if JUD is performed using VTS and *no* clustering is performed, i.e., each Gaussian is its own regression class, then JUD and VTS adaptation are identical. The results reported using JUD indicate that speech recognition accuracy improves as the number of regression classes increases [19]. Thus, as the number of regression classes approaches the number of Gaussian components, the performance of JUD will approach, but not exceed, that of VTS. Thus, it can also be surmised that if no clustering is performed in JAT, and VTS was used by JAT in learning the parameters of the canonical model, then JAT and NAT would converge to approximately the same set of model parameters. However, to the best of the authors' knowledge, the JAT experiments reported in the literature have used fairly aggressive clustering (a small number of regression classes compared to the number of Gaussians in the system). Although these results demonstrate the performance and efficiency of the combination of JAT and JUD, no notion of the upper bound in performance is known. Thus, in the same way that VTS represents an upper bound on performance for JUD, one can consider the performance of the NAT algorithm proposed in this paper as an upper bound on the performance of a JAT approach that uses transformations based on regression classes.

## V. Experiments and Results

To verify the effectiveness of the proposed NAT method, a series of experiments were conducted on the Aurora 2 and Aurora 3 connected digit recognition corpora using the HTK speech recognition system [26]. The Aurora 2 consists of data degraded with eight types of noise artificially added at signal-to-noise ratios (SNRs) varying from $-5$ dB to 20 dB and channel distortion [27]. Three test sets provided with the task are contaminated with noise types seen in the training data (Set A), unseen in the training data (Set B), and additive noise plus channel distortion (Set C). The acoustic models were trained using the standard "complex back end" Aurora 2 recipe [28]. An HMM with 16 states per digit and 20 Gaussian mixtures per state is created for each digit as a whole word. In addition, a three-state silence model with 36 Gaussian mixtures per state and a one state short

pause model which is tied to the middle stage of silence model are used.

Aurora 2 consists of data generated by digitally adding noise to clean speech. We also performed experiments using the Aurora 3 corpus in order to evaluate the algorithm's performance on real data actually collected in a noisy environment. Aurora 3 consists of connected digit strings recorded in realistic car environments [29]. Each utterance is recorded using either a close-talk or hands-free far field microphone and labeled as coming from either a high, medium, or low noise condition. There are four languages: Finnish, Spanish, German, and Danish and three experiment conditions: well-matched, medium-matched, and highly-mismatched. The acoustic models were trained using the standard "simple back end" scripts [27]. An HMM with 16 states per digit and three Gaussian mixtures per state is created for each digit as a whole word. A three state silence model with six Gaussian mixtures per state and a one state short pause model which is tied to the middle stage of silence model are included.

For both Aurora 2 and 3, 39-dimensional MFCC features consisting of 13 cepstral features plus delta and delta-delta features are used in the experiments. The cepstral coefficient of order zero (C0) is used instead of log energy. The cepstra are computed based on the spectral magnitudes, as in the standard Aurora front end [27]. Also, the first and last $N = 20$ frames of each utterance are used to initialize noise mean and variance in the NAT algorithm.

In the experiments, we compared performance obtained by the proposed method (denoted as NAT in tables and figures), and that of standard VTS model adaptation (denoted as VTS in tables and figures). As mentioned earlier, the NAT and the VTS perform the identical adaptation at test time and only differ in how the HMM parameters are trained. The HMMs are trained using the standard ML training for the VTS results, and using the proposed NAT algorithm described in Section III for the NAT results. In Table I, the baseline results obtained with no compensation are presented along with the results obtained with NAT and VTS model adaptation methods when multi-condition data is used for training in the Aurora 2 task. The baseline multi-condition data training (MT) obtains 90.35% average word recognition accuracy, whereas the VTS model adaptation improves the results and achieves 92.30% average word recognition accuracy. However, NAT further improves the performance, and achieves 93.75% average word recognition accuracy. In addition, we present the relative improvement of NAT and VTS model adaptation methods over the MT baseline at each SNR level in Fig. 2. From Fig. 2 and Table I, it can be concluded that NAT significantly outperforms both the baseline and the VTS model adaptation for all SNR conditions when multi-condition data is used for training in the Aurora 2 task.

We also compare the results obtained by several well-known algorithms including cepstral mean normalization (CMN), cepstral mean and variance normalization (CMVN), and the ETSI advanced front-end (AFE) [30]. The AFE is a good representation of state of the art in the feature enhancement style of processing on these tasks. In Table II, we present word accuracy results for Aurora 2 using multi-condition training data for a set of methods for each test set. The proposed NAT method outperforms all other methods, and provides 11.97% relative improvement over CMN, 3.85% relative improvement over CMVN, 7.54% relative improvement over AFE, and 18.83% relative improvement over the VTS method. The detailed results obtained with NAT using multi-condition training data for Aurora 2 are presented in Table III.

We also applied NAT to the ML trained acoustic models using clean data to check whether the results could be improved. The set of results obtained using clean training data is presented in Table IV for Aurora 2. NAT provides a small improvement over the VTS model adaptation (92.75% versus 92.94%) showing that the clean models may not be considered completely noise-free due to potential microphone differences and speaker differences, and the distortion model still has approximations which we can model in NAT. Also, when the acoustic models are

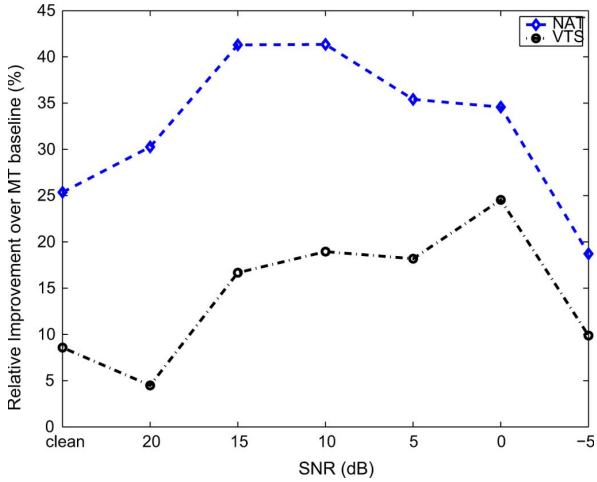



Fig. 2. Relative improvement (%) of NAT and VTS methods over the baseline using multi-condition training data.

TABLE I
WORD ACCURACY FOR AURORA 2 AT EACH SNR LEVEL USING MULTI-CONDITION TRAINING DATA

| SNR | Baseline | VTS | NAT |
|---|---|---|---|
| Clean | 99.46 | 99.51 | 99.60 |
| 20 dB | 98.95 | 99.00 | 99.27 |
| 15 dB | 98.09 | 98.41 | 98.88 |
| 10 dB | 96.03 | 96.78 | 97.67 |
| 5 dB | 90.02 | 91.83 | 93.55 |
| 0 dB | 68.67 | 76.36 | 79.50 |
| -5 dB | 30.74 | 37.58 | 43.70 |
| Ave. | 90.35 | 92.30 | 93.75 |

TABLE II
WORD ACCURACY FOR EACH SET OF AURORA 2 USING MODELS TRAINED ON MULTI-CONDITION DATA

| Method | Set A | Set B | Set C | Ave. |
|---|---|---|---|---|
| Baseline | 91.68 | 89.74 | 88.91 | 90.35 |
| CMN | 92.97 | 92.62 | 93.32 | 92.90 |
| CMVN | 93.80 | 93.09 | 93.70 | 93.50 |
| AFE | 93.74 | 93.26 | 92.21 | 93.24 |
| VTS | 92.20 | 91.87 | 93.37 | 92.30 |
| **NAT** | **93.66** | **93.77** | **93.89** | **93.75** |

TABLE III
DETAILED WORD ACCURACY FOR AURORA 2 USING NAT MODELS TRAINED ON MULTI-CONDITION DATA

| | Set A | | | | | Set B | | | | | Set C | | | All |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Subway | Babble | Car | Exhib. | Ave. | Restau. | Street | Airport | Station | Ave. | Subway | Street | Ave. | Ave. |
| Clean | 99.57 | 99.52 | 99.64 | 99.69 | 99.6 | 99.57 | 99.52 | 99.64 | 99.69 | 99.6 | 99.63 | 99.55 | 99.59 | 99.60 |
| 20 dB | 99.26 | 99.06 | 99.46 | 99.2 | 99.24 | 99.36 | 99.03 | 99.4 | 99.57 | 99.34 | 99.29 | 99.18 | 99.24 | 99.27 |
| 15 dB | 98.89 | 98.58 | 99.16 | 98.95 | 98.89 | 98.96 | 98.85 | 99.05 | 99.14 | 99.00 | 98.89 | 98.58 | 98.73 | 98.88 |
| 10 dB | 97.64 | 97.31 | 98.18 | 97.13 | 97.56 | 97.27 | 97.28 | 98.45 | 98.15 | 97.79 | 98.04 | 97.28 | 97.66 | 97.67 |
| 5 dB | 93.25 | 91.32 | 95.59 | 93.92 | 93.52 | 92.69 | 92.99 | 94.93 | 94.01 | 93.66 | 94.35 | 92.59 | 93.47 | 93.55 |
| 0 dB | 79.67 | 68.05 | 85.48 | 83.12 | 79.08 | 72.55 | 80.11 | 82.7 | 80.84 | 79.05 | 81.61 | 79.11 | 80.36 | 79.50 |
| -5 dB | 42.37 | 19.23 | 49.93 | 57.54 | 42.27 | 29.23 | 44.89 | 46.08 | 47.55 | 41.94 | 48.42 | 45.34 | 46.88 | 43.70 |
| Ave. | 93.74 | 90.86 | 95.57 | 94.46 | 93.66 | 92.17 | 93.65 | 94.91 | 94.34 | 93.77 | 94.44 | 93.35 | 93.89 | 93.75 |

TABLE IV
WORD ACCURACY FOR EACH SET OF AURORA 2 USING MODELS TRAINED ON CLEAN DATA

| Method | Set A | Set B | Set C | Ave. |
|---|---|---|---|---|
| Baseline | 60.43 | 55.85 | 69.01 | 60.31 |
| CMN | 68.65 | 73.71 | 69.69 | 70.88 |
| CMVN | 84.46 | 85.55 | 84.84 | 84.97 |
| AFE | 89.27 | 87.92 | 88.53 | 88.58 |
| VTS | 92.61 | 92.87 | 92.76 | 92.75 |
| **NAT** | **92.79** | **93.26** | **92.59** | **92.94** |

TABLE V
WORD ACCURACY FOR THE AURORA 3 EXPERIMENTAL CONDITIONS

| Method | Well | Mid | High | Ave |
|---|---|---|---|---|
| Baseline | 91.34 | 78.4 | 55.84 | 77.94 |
| CMN | 92.97 | 84.43 | 71.57 | 84.63 |
| CMVN | 94.22 | 87.92 | 83.40 | 89.31 |
| AFE | 95.30 | 86.79 | 87.25 | 90.31 |
| VTS | 91.33 | 80.25 | 86.57 | 86.26 |
| **NAT** | **94.44** | **87.55** | **88.98** | **90.66** |

TABLE VI
DETAILED WORD ACCURACY FOR AURORA 3 USING NAT MODELS

| | Finnish | Spanish | German | Danish | Ave. |
|---|---|---|---|---|---|
| Well | 95.41 | 95.97 | 94.87 | 91.49 | 94.44 |
| Mid | 87.14 | 92.25 | 89.17 | 81.64 | 87.55 |
| High | 90.57 | 92.42 | 90.38 | 82.56 | 88.98 |
| Ave. | 91.31 | 93.78 | 91.75 | 85.81 | 90.66 |

trained with clean data, both VTS and NAT, perform substantially better than the front-end feature enhancement methods under the noisy test conditions, and NAT achieves the highest accuracy.

In Table V, the results obtained with Aurora 3 are presented for the baseline, the CMN, the CMVN, and the AFE methods. In Aurora 3, there is no clean data available for training. Hence, the acoustic models are generated using the standard training data provided with the database for each experimental condition. When the ML trained models are adapted at runtime using the VTS algorithm, the average word recognition accuracy is 86.26% for the Aurora 3 task. The proposed NAT algorithm achieves 90.66% average word recognition accuracy and outperforms all other methods. NAT provides 39.23% relative improvement over CMN, 12.63% relative improvement over CMVN, 3.61% relative improvement over AFE, and 32.02% relative improvement over the VTS model adaptation. The detailed results obtained with NAT for Aurora 3 are presented in Table VI.

## VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a noise adaptive training algorithm for noise robust automatic speech recognition. The method estimates noise and channel distortion parameters for each utterance and uses these parameters to normalize environmental distortion as part of HMM training using a new formulation of the EM algorithm. In contrast to the feature enhancement methods, NAT estimates the underlying "pseudo-clean" model parameters directly without relying on point estimates of the clean speech features as an intermediate step. NAT uses a vector Taylor series expansion approach to linearize the nonlinear environment distortion model in the cepstral domain. The pseudo-clean model parameters learned with NAT are later used with VTS model adaptation for decoding noisy utterances at test time. Incorporating the VTS approximation used at test time in training is aimed at reducing the mismatch between training and test.

A set of experiments were conducted to test the proposed noise adaptive training method. NAT has achieved 93.75% average word recognition accuracy for Aurora 2 using multi-condition training data and a complex HMM backend and 90.66% average word recognition accuracy for Aurora 3 using a simple HMM backend. We compared the performance of NAT to the state-of-the-art (to the best of our knowledge) model adaptation (VTS) [15] and to feature-based noise adaptive training (applying AFE feature enhancement [29] to both training and test data), and demonstrated that NAT outperforms both methods on the Aurora 2 and 3 tasks. From these experiments, we can conclude that the proposed noise adaptive training method is an effective method for noise robust automatic speech recognition.

One of the strengths of the NAT method is that it can use both clean and corrupted speech for training. This is especially beneficial when there is no clean data available for training. However, if clean training data are available, we have shown experimentally that training the acoustic models using NAT results in better recognition performance than models trained in the conventional manner. One explanation for this improvement is that although clean data has little noise from the environment, there is still significant variability due to other factors such as microphone characteristics and positioning, instrumental noise conditions, and speaker variations. By applying NAT to clean training data, these additional sources of variance are normalized. Another reason for this improvement is that VTS adaptation makes some approximations and incorporating those approximations into training is beneficial. In other words, we achieve a better match between training and test conditions.

As we discussed in Section IV, NAT has a similar objective to two other adaptive training schemes, IVN and JAT. However,

there are key differences among the algorithms. In IVN, only the static components of the "pseudo-clean" HMM parameters are updated using the VTS environmental distortion model while in NAT, the static, delta, and delta-delta components are estimated. Furthermore, different forms of the auxiliary function are used in IVN and NAT, and we believe the one used in this work (and in [15]) better accounts for the approximations used in the VTS algorithm. We believe these differences explain the improvement observed on the Aurora 2 task using NAT (93.75%) compared to IVN (93.10%) [17].

In JAT, the HMM parameters are estimated using JUD-based adaptation to normalize the environmental distortion. In contrast to NAT (and VTS), where every Gaussian component is adapted individually, JAT (and JUD) performs adaptation utilizing a shared set of transforms estimated from regression classes. Using regression classes improves computational efficiency, typically at the expense of recognition accuracy. For this reason, the combination of NAT in training and VTS at runtime can be viewed as an upper bound on the performance of JAT and JUD. Furthermore, JAT uses iterative gradient-based optimization to estimate both the means and variances of the HMMs. In NAT, an iterative approach is only required for estimating the variances. By avoiding the use of gradient-based methods in the mean updates, NAT allows for more efficient and potentially more reliable estimation of the HMM parameters.

In the future, we plan to address the assumption used in our algorithm that there is zero cross correlation between speech and noise. In fact, other researchers have shown that this term can in fact be nonzero and have obtained improved results by performing VTS with a phase-sensitive model of speech corruption. We hope to similarly improve the NAT algorithm by incorporating this phase-sensitive term into the algorithm. We also plan to apply the NAT algorithm to a large vocabulary task.

## APPENDIX A
## DERIVATION OF THE UPDATE FORMULAS FOR DISTORTION PARAMETERS[1]

We start with the following EM auxiliary function:

$$Q(\Phi,\Lambda_X,\bar{\Phi},\bar{\Lambda}_X)=\sum_{i=1}^{I}\sum_{t,s,m}\gamma_{tsm}^{(i)}\log(p(\mathbf{y}_t^{(i)}|s,m,\Lambda_X,\Phi)) \tag{29}$$

where $i$ is utterance index, $\sum_{t,s,m}$ represents summation over frames, states, and Gaussians, $\Lambda_X$ and $\Phi$ are the set of model and distortion parameters we seek to optimize, and $\bar{\Lambda}_X$ and $\bar{\Phi}$ are the current estimate of these parameters. It is obvious that we need to compute $\gamma_{tsm}^{(i)}$ the posterior probability of the $m$th Gaussian in the $s$th state of the HMM for frame $t$ of the $i$th utterance. We can compute it using (14), where $\boldsymbol{\nu}_{sm}^{(i)}$, $\boldsymbol{\Psi}_{sm}^{(i)}$ are computed using VTS model adaptation with (6) and (7), respectively.

*Updating Means of Noise and Channel:* We first compute the distortion parameters for the $i$th utterance. By ignoring the constant terms with respect to $\phi^{(i)}$, we can rewrite the expression for the auxiliary functions

$$Q=\sum_{t,s,m}\gamma_{tsm}^{(i)} \times\left\{-\frac{1}{2}\log|\boldsymbol{\Psi}_{sm}^{(i)}|-\frac{1}{2}(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm}^{(i)})\right\} \tag{30}$$

where $\sum_{t,s,m}$ denotes summation over all frames of $i$th utterance, states and Gaussians. In other words, the distortion parameters are estimated on an utterance-by-utterance basis. $\boldsymbol{\nu}_{sm}^{(i)}$ is a function of $\boldsymbol{\mu}_{\mathbf{n}}^{(i)}$ and $\boldsymbol{\mu}_{\mathbf{h}}^{(i)}$ as in (6), so is the $Q$ function in (30). To compute the mean formula for noise, we take the derivative of the $Q$ function in (30) with respect to $\boldsymbol{\mu}_{\mathbf{n}}^{(i)}$ and set the result equal to zero. This produces

$$\sum_{t,s,m}\gamma_{tsm}^{(i)}(\mathbf{F}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm}^{(i)})=0. \tag{31}$$

Then,

$$\sum_{t,s,m}\gamma_{tsm}^{(i)}(\mathbf{F}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}\mathbf{F}_{sm}^{(i)}(\boldsymbol{\mu}_{\mathbf{n}}^{(i)}-\boldsymbol{\mu}_{\mathbf{n},0}^{(i)}) =\sum_{t,s,m}\gamma_{tsm}^{(i)}\times\left\{(\mathbf{F}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm,0}^{(i)})\right\} \tag{32}$$

which can be solved for $\boldsymbol{\mu}_{\mathbf{n}}^{(i)}$ to obtain the following update formula:

$$\boldsymbol{\mu}_{\mathbf{n}}^{(i)}=\boldsymbol{\mu}_{\mathbf{n},0}^{(i)}+\left\{\sum_{t,s,m}\gamma_{tsm}^{(i)}(\mathbf{F}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{F}_{sm}^{(i)})\right\}^{-1} \times\left\{\sum_{t,s,m}\gamma_{tsm}^{(i)}(\mathbf{F}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm,0}^{(i)})\right\}. \tag{33}$$

Similarly, to compute the mean formula for the channel, we take the derivative of the $Q$ function in (30) with respect to $\boldsymbol{\mu}_{\mathbf{h}}^{(i)}$ and set the result equal to zero. This produces

$$\boldsymbol{\mu}_{\mathbf{h}}^{(i)}=\boldsymbol{\mu}_{\mathbf{h},0}^{(i)}+\left\{\sum_{t,s,m}\gamma_{tsm}^{(i)}(\mathbf{G}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{G}_{sm}^{(i)})\right\}^{-1} \times\left\{\sum_{t,s,m}\gamma_{tsm}^{(i)}(\mathbf{G}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm,0}^{(i)})\right\}. \tag{34}$$

*Updating Covariance Matrix of Noise:* Given the auxiliary function in (30), there is no closed form solution for the noise covariance matrices, so they are optimized iteratively using Newton's method according to the following update equation:

$$\boldsymbol{\Sigma}_{\mathbf{n}}^{(i)}=\boldsymbol{\Sigma}_{\mathbf{n},0}^{(i)}-\left[\left(\frac{\partial^2 Q}{\partial^2\boldsymbol{\Sigma}_{\mathbf{n}}^{(i)}}\right)^{-1}\left(\frac{\partial Q}{\partial\boldsymbol{\Sigma}_{\mathbf{n}}^{(i)}}\right)\right]_{\boldsymbol{\Sigma}_{\mathbf{n}}^{(i)}=\boldsymbol{\Sigma}_{\mathbf{n},0}^{(i)}}. \tag{35}$$

[1]Similar derivations of the update equations for the VTS parameters can be found in [25]. In [25], the noise variances are updated using a first-order gradient-ascent method rather than the second-order method shown here.

We assume that the covariance matrices $\mathbf{\Psi}_{sm}^{(i)}, \mathbf{\Sigma}_{sm}, \mathbf{\Sigma}_{\mathbf{n}}^{(i)}$ are all diagonal. Then, we can write the covariance matrices as vectors

$$\boldsymbol{\psi}_{sm}^{2^{(i)}} = [\psi_{sm,1}^{2^{(i)}}, \psi_{sm,2}^{2^{(i)}}, \ldots, \psi_{sm,d}^{2^{(i)}}] \quad (36)$$

$$\boldsymbol{\sigma}_{sm}^{2} = [\sigma_{sm,1}^{2}, \sigma_{sm,2}^{2}, \ldots, \sigma_{sm,D}^{2}] \quad (37)$$

$$\boldsymbol{\sigma}_{\mathbf{n}}^{2^{(i)}} = [\sigma_{n,1}^{2^{(i)}}, \sigma_{n,2}^{2^{(i)}}, \ldots, \sigma_{n,D}^{2^{(i)}}] \quad (38)$$

where $D$ is the dimension of the feature vector; i.e., typically $D = 13$ dimensional cepstra are used in the traditional automatic speech recognition. Then, we can rewrite the auxiliary function in (30) explicitly as

$$Q = \sum_{t,s,m} -\frac{1}{2}\gamma_{tsm}^{(i)} \times \left\{ \sum_{d=1}^{D} \left( \log \psi_{sm,d}^{2^{(i)}} + \frac{(y_{t,d}^{(i)} - \nu_{sm,d}^{(i)})^2}{\psi_{sm,d}^{2^{(i)}}} \right) \right\} \quad (39)$$

where $y_{t,d}^{(i)}$ is the $d$th dimension of the feature vector at time point $t$ that belongs to the $i$th utterance $\mathbf{y}_t^{(i)}$, and $\nu_{sm,d}^{(i)}$ is the $d$th element of the VTS adapted mean vector $\boldsymbol{\nu}_{sm}^{(i)}$ for the $i$th utterance.

To compute the first and second derivatives of the $Q$ function with respect to the noise variance, we can expand $\mathbf{G}_{sm}^{(i)}$ and $\mathbf{F}_{sm}^{(i)}$ matrices in (7) as:

$$\mathbf{G}_{sm}^{(i)} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1D} \\ g_{21} & g_{22} & \cdots & g_{2D} \\ \vdots & \vdots & \vdots & \vdots \\ g_{D1} & g_{D2} & \cdots & g_{DD} \end{bmatrix} \quad (40)$$

$$\mathbf{F}_{sm}^{(i)} = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1D} \\ f_{21} & f_{22} & \cdots & f_{2D} \\ \vdots & \vdots & \vdots & \vdots \\ f_{D1} & f_{D2} & \cdots & f_{DD} \end{bmatrix}. \quad (41)$$

Then, we can write the formula for the diagonal elements of the covariance matrix $\mathbf{\Psi}_{sm}^{(i)}$ given in (7) explicitly as

$$\psi_{sm,d}^{2^{(i)}} = \sum_{k=1}^{D} g_{dk}^2 \cdot \sigma_{sm,k}^2 + f_{dk}^2 . \sigma_{n,k}^{2^{(i)}}, \quad d = 1, \ldots, D. \quad (42)$$

We can compute the first-order derivative of the $Q$ function given in (39) with respect to the noise variance by applying the chain rule as follows:

$$\frac{\partial Q}{\partial \sigma_{n,p}^{2^{(i)}}} = \sum_{d} \frac{\partial Q}{\partial \psi_{sm,d}^{2^{(i)}}} \frac{\partial \psi_{sm,d}^{2^{(i)}}}{\partial \sigma_{n,p}^{2^{(i)}}} \quad (43)$$

where

$$\frac{\partial Q}{\partial \psi_{sm,d}^{2^{(i)}}} = -\frac{1}{2} \sum_{t,s,m} \gamma_{tsm}^{(i)} \left\{ \frac{1}{\psi_{sm,d}^{2^{(i)}}} \left( 1 - \frac{(y_{t,d}^{(i)} - \nu_{sm,d}^{(i)})^2}{\psi_{sm,d}^{2^{(i)}}} \right) \right\} \quad (44)$$

and

$$\frac{\partial \psi_{sm,d}^{2^{(i)}}}{\partial \sigma_{n,p}^{2^{(i)}}} = f_{dp}^2. \quad (45)$$

Using (43)–(45), we can write the first derivative of the $Q$ function with respect to the noise mean as

$$\frac{\partial Q}{\partial \sigma_{n,p}^{2^{(i)}}} = -\frac{1}{2} \sum_{t,s,m} \gamma_{tsm}^{(i)} \left\{ \sum_{d} \frac{f_{dp}^2}{\psi_{sm,d}^{2^{(i)}}} \left( 1 - \frac{(y_{t,d}^{(i)} - \nu_{sm,d}^{(i)})^2}{\psi_{sm,d}^{2^{(i)}}} \right) \right\}. \quad (46)$$

We can compute the second-order derivative of the $Q$ function given in (39) with respect to the noise variance as follows:

$$\frac{\partial Q^2}{\partial \sigma_{n,p}^{2^{(i)}} \partial \sigma_{n,l}^{2^{(i)}}} = \frac{\partial}{\partial \sigma_{n,l}^{2^{(i)}}} \left( \frac{\partial Q}{\partial \sigma_{n,p}^{2^{(i)}}} \right) \quad (47)$$

and it is computed as

$$\frac{\partial Q^2}{\partial \sigma_{n,p}^{2^{(i)}} \partial \sigma_{n,l}^{2^{(i)}}} = \frac{1}{2} \sum_{t,s,m} \gamma_{tsm}^{(i)} \times \left\{ \sum_{d} \frac{f_{dp}^2 f_{dl}^2}{\psi_{sm,d}^{4^{(i)}}} \left( 1 - 2\frac{(y_{t,d}^{(i)} - \nu_{sm,d}^{(i)})^2}{\psi_{sm,d}^{2^{(i)}}} \right) \right\}. \quad (48)$$

To ensure that the variance remains positive, the logarithm of the variance is estimated. A change of variable is made as

$$\tilde{\sigma}_{n,p}^{2^{(i)}} = \log(\sigma_{n,p}^{2^{(i)}}) \quad (49)$$

and at the end the exponential function is applied to compute the actual variance; i.e.,

$$\sigma_{n,p}^{2^{(i)}} = \exp(\tilde{\sigma}_{n,p}^{2^{(i)}}). \quad (50)$$

Hence, the derivatives are actually computed with respect to new variable $\tilde{\sigma}_{n,p}^{2^{(i)}}$. Similar to (43), we can compute the first-order derivative of the $Q$ function with respect to new variable $\tilde{\sigma}_{n,p}^{2^{(i)}}$ as follows:

$$\frac{\partial Q}{\partial \tilde{\sigma}_{n,p}^{2^{(i)}}} = \sum_{d} \frac{\partial Q}{\partial \psi_{sm,d}^{2^{(i)}}} \frac{\partial \psi_{sm,d}^{2^{(i)}}}{\partial \sigma_{n,p}^{2^{(i)}}} \frac{\partial \sigma_{n,p}^{2^{(i)}}}{\partial \tilde{\sigma}_{n,p}^{2^{(i)}}}. \quad (51)$$

Then,

$$\frac{\partial Q}{\partial \tilde{\sigma}_{n,p}^{2^{(i)}}} = -\frac{1}{2} \sum_{t,s,m} \gamma_{tsm}^{(i)} \times \left\{ \sum_{d} \frac{f_{dp}^2 \sigma_{n,p}^{2^{(i)}}}{\psi_{sm,d}^{2^{(i)}}} \left( 1 - \frac{\left(y_{t,d}^{(i)} - \nu_{sm,d}^{(i)}\right)^2}{\psi_{sm,d}^{2^{(i)}}} \right) \right\}. \quad (52)$$

The only difference between (52) and (46) is the additional term of $\sigma_{n,p}^{2^{(i)}}$ (52) has, since

$$\frac{d\sigma_{n,p}^{2^{(i)}}}{d\tilde{\sigma}_{n,p}^{2^{(i)}}} = \exp(\tilde{\sigma}_{n,p}^{2^{(i)}}) = \sigma_{n,p}^{2^{(i)}}. \quad (53)$$

Similarly, the second-order derivative of the $Q$ function with respect to new variable $\tilde{\sigma}_n^{2^{(i)}}$ can be found as shown in (54) at the bottom of the page.

## APPENDIX B
## DERIVATION OF THE UPDATE FORMULAS FOR MODEL PARAMETERS

To compute the pseudo-clean model parameters of the $m$th Gaussian in the HMM state $s$, we can re-write the auxiliary function given in (12) by ignoring the constant terms with respect to the model parameters $\boldsymbol{\mu}_{sm}$ and $\boldsymbol{\Sigma}_{sm}$:

$$Q=\sum_{i,t}\gamma_{tsm}^{(i)} \times\left\{-\frac{1}{2}\log|\boldsymbol{\Psi}_{sm}^{(i)}|-\frac{1}{2}(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm}^{(i)})\right\}. \tag{55}$$

The main difference between the auxiliary function for the model parameters in (55) and the auxiliary function for the environment distortion parameters given in (30) is that the summation is over all frames and utterances in the training set in (55), whereas it is over all states, Gaussians and frames of a single utterance in (30). In other words, the pseudo-clean model parameters of each Gaussian of the HMM states are estimated over all utterances available in the training set.

*Updating Means of Pseudo-Clean Model:* $\boldsymbol{\nu}_{sm}^{(i)}$ is a function of $\boldsymbol{\mu}_{sm}$ as given in (6), so is the Q function in (55). To compute the mean of the $m$th Gaussian in the $s$th state of the HMM, $\boldsymbol{\mu}_{sm}$, we take the derivative of the $Q$ function given in (55) with respect to $\boldsymbol{\mu}_{sm}$, and set the result to zero. This leads to following expression:

$$\sum_{i,t}\gamma_{tsm}^{(i)}(\mathbf{G}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm}^{(i)})=0. \tag{56}$$

Then, writing $\boldsymbol{\nu}_{sm}^{(i)}$ explicitly in (56) produces

$$\sum_{i,t}\gamma_{tsm}^{(i)}(\mathbf{G}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}\mathbf{G}_{sm}^{(i)}(\boldsymbol{\mu}_{sm}-\boldsymbol{\mu}_{sm,0}) =\sum_{i,t}\gamma_{tsm}^{(i)}\times\left\{(\mathbf{G}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm,0}^{(i)})\right\} \tag{57}$$

which can be solved for $\boldsymbol{\mu}_{sm}$ to obtain the following update formula

$$\boldsymbol{\mu}_{sm}=\boldsymbol{\mu}_{sm,0}+\left\{\sum_{i,t}\gamma_{tsm}^{(i)}(\mathbf{G}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}\mathbf{G}_{sm}^{(i)}\right\}^{-1} \times\left\{\sum_{i,t}\gamma_{tsm}^{(i)}(\mathbf{G}_{sm}^{(i)})^T(\boldsymbol{\Psi}_{sm}^{(i)})^{-1}\left(\mathbf{y}_t^{(i)}-\boldsymbol{\nu}_{sm,0}^{(i)}\right)\right\}. \tag{58}$$

*Updating Covariance Matrices of Pseudo-Clean Model:* There is no closed form solution for covariance matrices of the HMM distributions. As with the noise covariance, Newton's method is used to estimate them iteratively using (22). We follow the steps detailed in Appendix A for the derivation of the variance update equation, and to ensure that the variance remains positive a change of variable is made such that the logarithm of the variance is estimated, and later the exponential function is applied to compute the actual variance. Thus, the derivatives are actually computed to optimize the logarithm of the HMM variance $\tilde{\sigma}_{sm}^2$.

We use the chain rule to compute the first order derivative of the $Q$ function in (55) with respect to new variable $\tilde{\sigma}_{sm}^2$ as follows:

$$\frac{\partial Q}{\partial\tilde{\sigma}_{sm,p}^2}=\sum_{i,d}\frac{\partial Q}{\partial\psi_{sm,d}^{2^{(i)}}}\frac{\partial\psi_{sm,d}^{2^{(i)}}}{\partial\sigma_{sm,p}^2}\frac{\partial\sigma_{sm,p}^2}{\partial\tilde{\sigma}_{sm,p}^2} \tag{59}$$

where

$$\frac{\partial\psi_{sm,d}^{2^{(i)}}}{\partial\sigma_{sm,p}^2}=g_{dp}^2 \tag{60}$$

from (42), and similar to (53),

$$\frac{\partial\sigma_{sm,p}^2}{\partial\tilde{\sigma}_{sm,p}^2}=\exp(\tilde{\sigma}_{sm,p}^2)=\sigma_{sm,p}^2. \tag{61}$$

Then, the first order derivative of the $Q$ function with respect to $\tilde{\sigma}_{sm}^2$ is found to be:

$$\frac{\partial Q}{\partial\tilde{\sigma}_{sm,p}^2}=-\frac{1}{2}\sum_{i,t}\gamma_{tsm}^{(i)} \times\left\{\sum_d\frac{g_{dp}^2\sigma_{sm,p}^2}{\psi_{sm,d}^{2^{(i)}}}\left(1-\frac{\left(y_{t,d}^{(i)}-\nu_{sm,d}^{(i)}\right)^2}{\psi_{sm,d}^{2^{(i)}}}\right)\right\}. \tag{62}$$

$$\frac{\partial Q^2}{\partial\tilde{\sigma}_{n,p}^{2^{(i)}}\partial\tilde{\sigma}_{n,l}^{2^{(i)}}}=\frac{1}{2}\sum_{t,s,m}\gamma_{tsm}^{(i)}\left\{\sum_d\frac{f_{dp}^2\sigma_{n,p}^{2^{(i)}}f_{dl}^2\sigma_{n,l}^{2^{(i)}}}{\psi_{sm,d}^{4^{(i)}}}\left(1-2\frac{\left(y_{t,d}^{(i)}-\nu_{sm,d}^{(i)}\right)^2}{\psi_{sm,d}^{2^{(i)}}}\right) -\delta(l-p)\sum_d\frac{f_{dp}^2\sigma_{n,p}^{2^{(i)}}}{\psi_{sm,d}^{2^{(i)}}}\left(1-\frac{\left(y_{t,d}^{(i)}-\nu_{sm,d}^{(i)}\right)^2}{\psi_{sm,d}^{2^{(i)}}}\right)\right\} \tag{54}$$

$$\frac{\partial Q^2}{\partial \tilde{\sigma}^2_{sm,p} \partial \tilde{\sigma}^2_{sm,l}} = \frac{1}{2} \sum_{i,t} \gamma^{(i)}_{tsm} \left\{ \sum_d \frac{g^2_{dp} \sigma^2_{sm,p} g^2_{dl} \sigma^2_{sm,l}}{\psi^{4(i)}_{sm,d}} \left( 1 - 2 \frac{\left( y^{(i)}_{t,d} - \nu^{(i)}_{sm,d} \right)^2}{\psi^{2(i)}_{sm,d}} \right) - \delta(l-p) \sum_d \frac{g^2_{dp} \sigma^2_{sm,p}}{\psi^{2(i)}_{sm,d}} \left( 1 - \frac{\left( y^{(i)}_{t,d} - \nu^{(i)}_{sm,d} \right)^2}{\psi^{2(i)}_{sm,d}} \right) \right\}. \quad (63)$$

The equation in (62) and (52) are the same except: i) the summation is over all frames and utterances in (62) whereas it is over all states, Gaussians, and frames of a single utterance in (52) since the auxiliary functions in (55) and (30) are different; ii) the $f^2_{dp}$ and $\sigma^{2(i)}_{n,p}$ terms in (52) are replaced by $g^2_{dp}$ and $\sigma^2_{sm,p}$ terms in (62), respectively.

Similarly, the second-order derivative of the $Q$ function with respect to new variable $\tilde{\sigma}^2_{sm}$ can be found as given in (63) at the top of the page.

## ACKNOWLEDGMENT

We would like to thank Dr. Jinyu Li at Microsoft for valuable discussions.

## REFERENCES

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

[2] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "A minimum-mean-square-error noise reduction algorithm on mel-frequency cepstra for robust speech recognition," in *Proc. ICASSP*, Las Vegas, NV, 2008, pp. 4041–4044.

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

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

[5] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Process.*, vol. 2, pp. 291–298, Apr. 1994.

[6] V. Digalakis, D. Rtischev, L. Neumeyer, and E. Sa, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 5, pp. 357–366, Sep. 1995.

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

[8] M. K. Omar, "Regularized feature-based maximum likelihood linear regression for speech recognition," in *Proc. Interspeech*, Antwerp, Belgium, 2007, pp. 1561–1564.

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

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

[11] M. Gales, S. Young, and S. J. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 352–359, Sep. 1996.

[12] P. J. Moreno, B. Raj, and R. M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Proc. ICASSP*, 1996, pp. 733–736.

[13] D. Y. Kim, C. K. Un, and N. S. Kim, "Speech recognition in noisy environments using first-order vector Taylor series," *Speech Commun.*, vol. 24, no. 1, pp. 39–49, 1998.

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

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

[16] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP*, Philadelphia, PA, 1996, pp. 1137–1140.

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

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

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

[20] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*. Norwell, MA: Kluwer, 1993.

[21] R. A. Gopinath, M. Gales, P. S. Gopalakrishnan, S. Balakrishnan-Aiyer, and M. A. Picheny, "Robust speech recognition in noise: Performance of the IBM continuous speech recognizer on the ARPA noise spoke task," in *Proc. ARPA Workshop Spoken Lang. Syst. Technol.*, 1995.

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

[23] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[24] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englwood Cliffs, NJ: Prentice-Hall, 1983.

[25] H. Liao and M. J. F. Gales, "Joint uncertainty decoding for robust large vocabulary speech recognition," Cambridge Univ., 2006, Tech. Rep. CUED/F-INFENG/TR-522.

[26] S. J. Young, *The HTK Hidden Markov Model Toolkit: Design and Philosophy*. Cambridge, U.K.: Univ. of Cambridge, Dept. of Eng., 1994.

[27] H. G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR*, Paris, France, Sep. 2000, pp. 181–188.

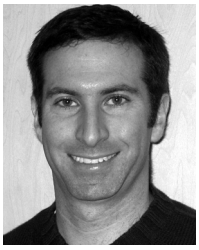
[28] D. Pierce and A. Gunawardana, "Aurora 2.0 Speech Recognition in Noise: Update 2. Complex Backend Definition for Aurora 2.0," 2002 [Online]. Available: http://icslp2002.colorado.edu/special_sessions/aurora

[29] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, J. Allen, and S. Euler, "Speechdat-car: A large speech database for automotive environments," in *Proc. LREC*, Athens, Greece, 2000.

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

**Ozlem Kalinli** (M'09) received the B.S. degree in electronics and communication engineering (*summa cum laude*) and with honors from Istanbul Technical University (ITU), Istanbul, Turkey, in 2001, the M.S. degree in electrical engineering with outstanding academic achievement from the Illinois Institute of Technology (IIT), Chicago, in 2003, and the Ph.D. degree in electrical engineering from the University of Southern California (USC), Los Angeles, in 2009.

She was a member of the Immersive Audio Laboratory at USC from 2003 to 2005 and a member of the Signal Analysis and Interpretation Laboratory (SAIL) at USC from 2005 to 2009. In the summer of 2008, she worked as an intern in the Speech Technology Group at Microsoft Research, Redmond, WA. In 2010, she joined the R&D Group at Sony Computer Entertainment of America, Foster City, CA. Her current research interests include bio-inspired signal processing for speech and audio applications, auditory attention, auditory perception, speech recognition, speech analysis, pattern recognition, and statistical techniques for speech and audio applications.



**Michael L. Seltzer** (M'95–SM'07) received the Sc.B. degree (with honors) from Brown University, Providence, RI, in 1996, and M.S. and Ph.D. degrees from Carnegie Mellon University, Pittsburgh, PA, in 2000 and 2003, respectively, all in electrical engineering.

From 1996 to 1998, he was an Applications Engineer at Teradyne, Inc., Boston, MA, working on semiconductor test solutions for mixed-signal devices. From 1998 to 2003, he was a member of the Robust Speech Recognition Group at Carnegie Mellon University. In 2003, Dr. Seltzer joined the Speech Technology Group at Microsoft Research, Redmond, WA. His current research interests include speech enhancement, speech recognition in adverse acoustical environments, acoustic modeling, microphone array processing, and machine learning for speech and audio applications.

Dr. Seltzer was awarded the Best Young Author paper award from the IEEE Signal Processing Society in 2006. From 2006 to 2008, he was a member of the Speech and Language Technical Committee (SLTC) and was the Editor-in-Chief of the SLTC e-Newsletter. He was a general co-chair of the 2008 International Workshop on Acoustic Echo and Noise Control and Publicity Chair of the 2008 IEEE Workshop on Spoken Language Technology. He is currently an Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.

**Jasha Droppo** (M'03–SM'07) received the B.S. degree in electrical engineering (with honors) from Gonzaga University, Spokane, WA, in 1994, and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, in 1996 and 2000, respectively.

At the University of Washington, he helped to develop and promote a discrete theory for time–frequency representations of audio signals, with a focus on speech recognition. He joined the Speech Technology Group, Microsoft Research, Redmond, WA, in the summer of 2000. His core interest is speech enhancement and features for automatic speech recognition, including the SPLICE algorithm, tree-based acoustic models, several techniques for model-based speech feature enhancement, and algorithms for learning non-parametric feature space warpings. His other interests include techniques for acoustic modeling, pitch tracking, multiple stream ASR, novel speech recognition features, multimodal interfaces, and cepstral compression and transport.

**Alex Acero** (M'90–SM'00–F'04) received the M.S. degree from the Polytechnic University of Madrid, Madrid, Spain, in 1985, the M.S. degree from Rice University, Houston, TX, in 1987, and the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, in 1990, all in electrical engineering.

He worked in Apple Computer's Advanced Technology Group in 1990–1991. In 1992, he joined Telefonica I+D, Madrid, as Manager of the Speech Technology Group. Since 1994, he has been with Microsoft Research, Redmond, WA, where he is currently a Research Area Manager directing an organization with 70 engineers conducting research in audio, speech, multimedia, communication, natural language, and information retrieval. He is also an affiliate Professor of Electrical Engineering at the University of Washington, Seattle. Dr. Acero is author of the books *Acoustical and Environmental Robustness in Automatic Speech Recognition* (Kluwer, 1993) and *Spoken Language Processing* (Prentice-Hall, 2001), has written invited chapters in four edited books and over 200 technical papers. He holds 71 U.S. patents.

Dr. Acero has served the IEEE Signal Processing Society as Vice President Technical Directions (2007–2009), 2006 Distinguished Lecturer, member of the Board of Governors (2004–2005), Associate Editor for the IEEE SIGNAL PROCESSING LETTERS (2003–2005) and the IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING (2005–2007), and member of the editorial board of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING (2006–2008) and the IEEE *Signal Processing Magazine* (2008–2010). He also served as member (1996–2000) and Chair (2000–2002) of the Speech Technical Committee of the IEEE Signal Processing Society. He was Publications Chair of ICASSP98, Sponsorship Chair of the 1999 IEEE Workshop on Automatic Speech Recognition and Understanding, General Co-Chair of the 2001 IEEE Workshop on Automatic Speech Recognition and Understanding and Panel Chair for 2009 IEEE Workshop on Automatic Speech Recognition and Understanding. Since 2004, he, along with coauthors Drs. Huang and Hon, has been using proceeds from their textbook *Spoken Language Processing* to fund the "IEEE Spoken Language Processing Student Travel Grant" for the best ICASSP student papers in the speech area. He has served as member of the editorial board of *Computer Speech and Language* and as member of Carnegie Mellon University Dean's Leadership Council for College of Engineering.