A Novel Framework and Training Algorithm for Variable-Parameter Hidden Markov Models

Dong Yu, *Senior Member, IEEE*, Li Deng, *Fellow, IEEE*, Yifan Gong, *Senior Member, IEEE*, and Alex Acero, *Fellow, IEEE*

***Abstract*—We propose a new framework and the associated maximum-likelihood and discriminative training algorithms for the variable-parameter hidden Markov model (VPHMM) whose mean and variance parameters vary as functions of additional environment-dependent conditioning parameters. Our framework differs from the VPHMM proposed by Cui and Gong (2007) in that piecewise spline interpolation instead of global polynomial regression is used to represent the dependency of the HMM parameters on the conditioning parameters, and a more effective functional form is used to model the variances. Our framework unifies and extends the conventional discrete VPHMM. It no longer requires quantization in estimating the model parameters and can support both parameter sharing and instantaneous conditioning parameters naturally. We investigate the strengths and weaknesses of the model on the Aurora-3 corpus. We show that under the well-matched condition the proposed discriminatively trained VPHMM outperforms the conventional HMM trained in the same way with relative word error rate (WER) reduction of 19% and 15%, respectively, when only mean is updated and when both mean and variances are updated.**

***Index Terms*—Discriminative training, growth transformation, parameter clustering, speech recognition, spline interpolation, variable-parameter hidden Markov model (VPHMM).**

I. Introduction

HIDDEN Markov model (HMM) has been the prevailing modeling technique for automatic speech recognition (ASR) in the past decades. Its success largely lies on HMM's simplicity, flexible modeling ability, and the efficient learning algorithms. However, there has still been a conspicuous performance gap between the human and machine speech recognition, especially under noisy conditions [16]. Many efforts have been put during the past two decades in search for new modeling, adaptation, and training technologies (e.g., [1], [3], [5], [6], [9], [10], [13], [14], [21], [24], [25], [27]–[29]) that can deliver sufficiently high recognition accuracy under all deployment conditions. In this paper, we propose a novel framework and the associated maximum-likelihood and discriminative training algorithms under the umbrella of variable-parameter hidden Markov model (VPHMM) for improved speech recognition performance.

Manuscript received August 29, 2008; revised March 31, 2009. Current version published July 17, 2009. This manuscript greatly enhances and expands the work of [31] and [32] presented at Interspeech 2008, Brisbane, Australia. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mark Hasegawa-Johnson.

The authors are with Microsoft Research, Redmond, WA 98052 USA (e-mail: dongyu@microsoft.com; deng@microsoft.com; ygong@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.2009.2020890

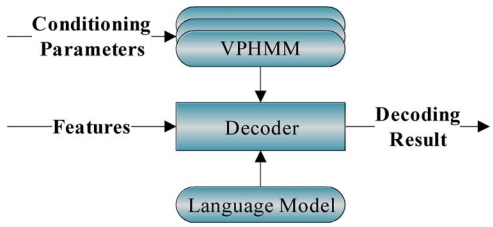


Fig. 1. Illustration of the use of the VPHMM for speech recognition.

The term of VPHMM was coined in the work of [6]. It was proposed as improvement to the multistyle-trained HMM with the goal to increase ASR accuracy under noisy environments, and is an extension to the earlier work of [18]. The basic idea behind the use of the general VPHMM for speech recognition is illustrated in Fig. 1. The main difference between the VPHMM and the conventional HMM is the VPHHM's exploitation of the additional environment-dependant conditioning parameter (or auxiliary feature [11]), such as the signal-to-noise ratio (SNR) or fundamental frequency, which determines the HMM parameters to be used by the decoder. Having the HMM parameters dependent on the conditioning parameter allows for better modeling of the speech under each environmental condition [7].

In the conventional HMM, the continuous observation density function $b_i(\mathbf{o}_{r,t})$ for state i and acoustic observation $\mathbf{o}_{r,t}$ at frame t in the utterance r is estimated using a mixture of L Gaussian components

$$b_i(\mathbf{o}_{r,t}) = \sum_{l=1}^{L} w_{i,l} b_{i,l}(\mathbf{o}_{r,t}) = \sum_{l=1}^{L} w_{i,l} N(\mathbf{o}_{r,t}|\boldsymbol{\mu}_{i,l}, \boldsymbol{\Sigma}_{i,l}) \quad (1)$$

where $N(\mathbf{o}_{r,t}|\boldsymbol{\mu}_{i,l}, \boldsymbol{\Sigma}_{i,l})$ is the lth Gaussian mixture component with fixed mean $\boldsymbol{\mu}_{i,l}$ and variance $\boldsymbol{\Sigma}_{i,l}$, $w_{i,l}$ is a positive weight for the lth Gaussian component with the constraint $\sum_{l=1,\ldots,L} w_{i,l} = 1$.

In the VPHMM, $\boldsymbol{\mu}_{i,l}$ and $\boldsymbol{\Sigma}_{i,l}$ change as functions of the conditioning parameter $\boldsymbol{\zeta}$ (which can be relatively easily and reliably estimated), i.e.,

$$b_i(\mathbf{o}_{r,t}, \boldsymbol{\zeta}) = \sum_{l=1}^{L} w_{i,l} N(\mathbf{o}_{r,t}|\boldsymbol{\mu}_{i,l}(\boldsymbol{\zeta}), \boldsymbol{\Sigma}_{i,l}(\boldsymbol{\zeta})). \quad (2)$$

In its simplest form, the conditioning parameter takes only discrete values (e.g., gender, dialect type) and the VPHMM is equivalent to training a discrete set of HMMs and selecting the appropriate HMM based on the conditioning parameter. In this simple form, there is usually no relationship among the HMM

parameters that are associated with different conditioning parameter values. The conditioning parameter is usually of categorical values, or quantized real values.

The work by Fujinaga *et al.* [11] extended the discrete conditioning parameters to continuous ones by modeling $\boldsymbol{\mu}_{i,l}(\boldsymbol{\zeta})$ as a linear regression function of conditioning parameters. The recent work [6], [7] by Cui and Gong further extended [11] by modeling both $\boldsymbol{\mu}_{i,l}(\boldsymbol{\zeta})$ and $\boldsymbol{\Sigma}_{i,l}(\boldsymbol{\zeta})$ with a polynomial (instead of linear) regression function over the utterance SNR and used the maximum-likelihood (ML) algorithm to estimate the regression parameters in the VPHMM. They assumed a diagonal covariance matrix in their model and allowed the means and variances in the dth dimension change as

$$\mu_{i,l,d}(\zeta_{r,t,d}) = \xi\left(\zeta_{r,t,d}|\rho^{(1)}_{\mu,i,l,d},\ldots,\rho^{(K_\mu)}_{\mu,i,l,d}\right) \tag{3}$$

and

$$\begin{aligned}\sigma^2_{i,l,d}(\zeta_{r,t,d}) &= \Sigma_{i,l,d}(\zeta_{r,t,d}) \\ &= \Sigma^{(0)}_{i,l,d} e^{\xi\left(\zeta_{r,t,d}|\rho^{(1)}_{\Sigma,i,l,d},\ldots,\rho^{(K_\Sigma)}_{\Sigma,i,l,d}\right)}\end{aligned} \tag{4}$$

where $\rho^{(1)}_{\mu,i,l,d},\ldots,\rho^{(K_\mu)}_{\mu,i,l,d}$ and $\rho^{(1)}_{\Sigma,i,l,d},\ldots,\rho^{(K_\Sigma)}_{\Sigma,i,l,d}$ are the polynomial parameters for the means and variances, respectively, and $\Sigma^{(0)}_{i,l,d}$ is the variance in the conventional HMM. The exponential function in (4) was chosen to guarantee $\Sigma_{i,l,d}(\zeta_{r,t,d}) > 0$.

There are several limitations in Cui and Gong's original model. First, due to the use of the exponential function in (4), quantization-based approximation approach has to be invoked to make parameter estimation feasible. Second, instantaneous conditioning parameters such as instantaneous SNR cannot be easily accommodated because their VPHMM has to be initialized from a set of HMMs trained under quantized SNR conditions. Third, the VPHMM parameters were trained using the ML criterion [26] with the expectation maximization algorithm [8]. It is slightly difficult to incorporate the discriminative training (DT) methods to train the regression parameters for the variances because the adoption of the formulation (4) prevents an easy factorization of the regression parameters from the sufficient statistics. Fourth, it cannot naturally incorporate categorical conditioning parameters into the framework. Furthermore, although it is possible to share the parameters in their model, neither algorithm nor experimental results were provided.

In this paper, we propose a substantially improved framework for VPHMM, and derive the related ML training, discriminative training, and parameter sharing algorithms. The key differences between our framework and Cui and Gong's original VPHMM are the choice of a different functional form to model the variances $\boldsymbol{\Sigma}_{i,l}(\boldsymbol{\zeta})$ and the use of spline interpolation to approximate both $\boldsymbol{\mu}_{i,l}(\boldsymbol{\zeta})$ and $\boldsymbol{\Sigma}_{i,l}(\boldsymbol{\zeta})$. As a result, the spline-based VPHMM developed and reported in this paper eliminates many of the limitations observed in the VPHMM proposed by Cui and Gong. Our new model is a unified framework that supports both the categorical and continuous-valued conditioning parameters. It also supports both the utterance-level and instantaneous conditioning parameters. We demonstrate the effectiveness of our spline-based VPHMM (S-VPHMM) on the Aurora-3 corpus (with and without the recently developed Mel-frequency cepstral minimum mean square error (MFCC-MMSE) motivated noise suppressor [29]). We focus on the use of instantaneous conditioning parameters and discriminative training algorithms as these are not supported by the original VPHMM. We show that S-VPHMM outperforms the discriminatively trained conventional HMMs with relative word error rate (WER) reduction of 19% and 15%, respectively, under the well-matched conditions when only mean is updated and when both mean and variances are updated, respectively, with parameter sharing.

The rest of the paper is organized as follows. In Section II, we introduce the parameterization form used in the S-VPHMM and discuss the issues associated with the conditioning parameters. In Section III, we derive the training algorithms for the S-VPHMM. In Section IV, we describe the parameter sharing and clustering algorithm. We report our experimental results in Section V and conclude the paper in Section VI.

II. A New Parameterization Form for VPHMM

As mentioned in Section I, one key novelty of the S-VPHMM comes from the specific parameterization form developed in this work. Instead of using the polynomial regression, the S-VPHMM uses spline interpolation with which only the values at several knots need to be learned. Spline interpolation is preferred over the previously developed polynomial regression because the interpolation error can be made small even when using low degree polynomials for the spline. Specifically, in the S-VPHMM, the dth dimension of the mean and variance vectors can be approximated with a spline ξ as

$$\mu_{i,l,d}(\zeta_{r,t,d}) = \mu^{(0)}_{i,l,d} + \xi\left(\zeta_{r,t,d}|\mu^{(1)}_{\varpi(i,l,d)},\ldots,\mu^{(K_\mu)}_{\varpi(i,l,d)}\right) \tag{5}$$

$$\begin{aligned}\sigma^2_{i,l,d}(\zeta_{r,t,d}) &= \Sigma_{i,l,d}(\zeta_{r,t,d}) \\ &= \Sigma^{(0)}_{i,l,d}\xi^{-2}\left(\zeta_{r,t,d}|\Sigma^{(1)}_{\varpi(i,l,d)},\ldots,\Sigma^{(K_\Sigma)}_{\varpi(i,l,d)}\right)\end{aligned} \tag{6}$$

where we have assumed that all covariance matrices are diagonal, $\mu^{(0)}_{i,l,d}$ and $\Sigma^{(0)}_{i,l,d}$ are the Gaussian-component-specific mean and variance, $\mu^{(1)}_{\varpi(i,l,d)},\ldots,\mu^{(K_\mu)}_{\varpi(i,l,d)}$ and $\Sigma^{(1)}_{\varpi(i,l,d)},\ldots,\Sigma^{(K_\Sigma)}_{\varpi(i,l,d)}$ are the spline knots that may be shared across different Gaussian components, and $\varpi(i,l,d)$ is the regression class that determines the sharing group. The spline parameter sharing works as follows. During the training time, splines are grouped into regression classes based on some criterion as detailed in Section IV. During the decoding time, the means $\mu_{i,l,d}(\zeta_{r,t,d})$ and variances $\sigma^2_{i,l,d}(\zeta_{r,t,d})$ of a Gaussian component are determined by first finding out the regression class $\varpi(i,l,d)$ and the associated spline parameters $\mu^{(1)}_{\varpi(i,l,d)},\ldots,\mu^{(K_\mu)}_{\varpi(i,l,d)}$ and $\Sigma^{(1)}_{\varpi(i,l,d)},\ldots,\Sigma^{(K_\Sigma)}_{\varpi(i,l,d)}$, and then determining the values using (5) and (6). The purpose of the spline parameter sharing is to reduce the number of parameters (and thus the model size and computation time) and to make the estimation of the spline parameters more reliable as will be demonstrated in Section V.

Note that the form (6) differs from (4) and is one of the key characteristics of the S-VPHMM. Using the inverse square function instead of the exponential function leads to significantly

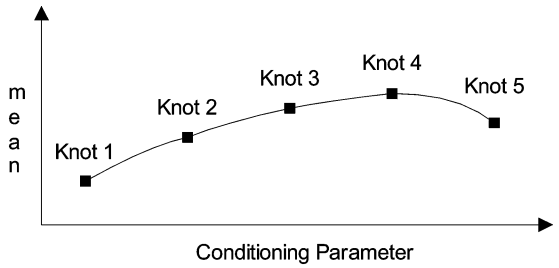


Fig. 2. Approximate the mean of a Gaussian component under different conditioning parameter values with an evenly distributed 5-knot cubic-spline.

simplified re-estimation formulas and overcomes many limitations observed in the original VPHMM. Both approaches, however, can be applied only to the diagonal covariance matrices. Alternatively, we may remove the positive-definite constraint on the covariance matrix in the optimization process and project the result back to the positive-subspace. This would allow us to use the simple covariance matrix (without introducing the inverse square function) as the optimization variable in (6) at the cost of possibly leading to a suboptimal solution due to the projection operation.

Any type of spline (or piecewise functions) may be used. Two most commonly used splines are the linear spline and the cubic spline since the values of these splines can be efficiently calculated. In this paper, we used the cubic spline which is smooth up to the second-order derivative. There are two typical boundary conditions for the cubic spline: one for which the first derivative is zero and one where the second derivative is zero. The spline with the latter boundary condition is usually called natural spline and is the one used in this study. Fig. 2 illustrates an example where an evenly distributed 5-knot cubic spline is used to approximate the mean of a Gaussian component under different values of the conditioning parameter.

As indicated in the Appendix A, if K evenly distributed knots $\left\{\left(x^{(i)}, y^{(i)}\right) | i = 1, \ldots, K; x^{(i+1)} - x^{(i)} = h\right\}$ are known, the value of a data point x can be estimated as

$$y = \left(\mathbf{M}_x^T + \mathbf{Q}_x^T \mathbf{Z}^{-1} \mathbf{W}\right) \tilde{\mathbf{y}} \tag{7}$$

where $\tilde{\mathbf{y}}$, $\mathbf{Q}_x$, $\mathbf{Z}$, and $\mathbf{W}$ are defined in (56)–(60), respectively.

By denoting

$$\tilde{\boldsymbol{\mu}}_{\varpi(i,l,d)} = \left[\mu_{\varpi(i,l,d)}^{(1)} \quad \cdots \quad \mu_{\varpi(i,l,d)}^{(K_\mu)}\right]^T \tag{8}$$

$$\boldsymbol{\nu}_{\varpi(i,l,d)}^T (\zeta_{r,t,d}) = \mathbf{M}_{\mu,\zeta_{r,t,d}}^T + \mathbf{Q}_{\mu,\zeta_{r,t,d}}^T \mathbf{Z}_\mu^{-1} \mathbf{W}_\mu \tag{9}$$

$$\tilde{\boldsymbol{\Sigma}}_{\varpi(i,l,d)} = \left[\Sigma_{\varpi(i,l,d)}^{(1)} \quad \cdots \quad \Sigma_{\varpi(i,l,d)}^{(K_\Sigma)}\right]^T \tag{10}$$

and

$$\boldsymbol{\varsigma}_{\varpi(i,l,d)}^T (\zeta_{r,t,d}) = \mathbf{M}_{\Sigma,\zeta_{r,t,d}}^T + \mathbf{Q}_{\Sigma,\zeta_{r,t,d}}^T \mathbf{Z}_\Sigma^{-1} \mathbf{W}_\Sigma \tag{11}$$

the parametric form (5) and (6) can be rewritten succinctly as

$$\mu_{i,l,d} (\zeta_{r,t,d}) = \mu_{i,l,d}^{(0)} + \boldsymbol{\nu}_{\varpi(i,l,d)}^T (\zeta_{r,t,d}) \tilde{\boldsymbol{\mu}}_{\varpi(i,l,d)} \tag{12}$$

and

$$\begin{aligned} \sigma_{i,l,d}^2 (\zeta_{r,t,d}) &= \Sigma_{i,l,d} (\zeta_{r,t,d}) \\ &= \Sigma_{i,l,d}^{(0)} \left(\boldsymbol{\varsigma}_{\varpi(i,l,d)}^T (\zeta_{r,t,d}) \tilde{\boldsymbol{\Sigma}}_{\varpi(i,l,d)}\right)^{-2}. \end{aligned} \tag{13}$$

Note that the number of knots for the means K_x and variances K_σ need not to be the same and so $\mathbf{M}_x$, $\mathbf{Q}_x$, $\mathbf{Z}$, and $\mathbf{W}$ may be different for the means and variances. Also note that the same formulations of (12) and (13) can be derived for the linear spline and other piecewise and interpolation functions by defining $\boldsymbol{\nu}_{\varpi(i,l,d)}^T (\zeta_{r,t,d})$ and $\boldsymbol{\varsigma}_{\varpi(i,l,d)}^T (\zeta_{r,t,d})$ differently. Interesting examples are the cluster adaptive training (CAT) [14], [33] where the means for a speaker are linearly interpolated from clusters of speakers, and the subspace precision and mean (SPAM) [1], [16] approach where the means and precisions are constructed from basis means and precisions.

The parametric form (12) and (13) can be applied to both the discrete and continuous-valued conditioning parameters. Applying them to the discrete conditioning parameters is straightforward if the discrete values are quantized from the continuous-valued conditioning parameters. If the discrete values are of the categorical type, we need to convert the values into evenly-spaced integers and make each integer a knot. For example, if gender which takes values of *male*, *female* and *child* is used as the conditioning parameter, we can convert them into integers 1, 2, and 3, respectively, and make each value a knot. In this case, the knots learned correspond to the HMMs associated with the discrete conditioning parameter values.

Compared to the approach that trains and uses the discrete set of HMMs directly, the S-VPHMM has four advantages. First, the S-VPHMM can naturally group and share the knots as illustrated in Section IV. Second, the S-VPHMM allows for easy training of systems with instantaneous discrete conditioning parameters. Third, the discrete VPHMM may introduce quantization errors if the values are quantized from the continuous-valued parameters. The S-VPHMM can alleviate this problem by using continuous values directly. Fourth, The S-VPHMM provides a unified training and decoding framework for both the discrete and continuous-valued conditioning parameters and makes the code maintenance easier.

The range of the discrete-valued conditioning parameter can be easily determined by converting the discrete values (e.g., of a categorical variable) into integers. The range of the continuous-valued conditioning parameter can be learned from the training data. In our study, we have used

$$\zeta_{r,t,d}^{(1)} = \zeta_d^{(1)} = \tau_{\zeta_d} - \kappa \psi_{\zeta_d}, \text{ and} \tag{14}$$

$$\zeta_{r,t,d}^{(K)} = \zeta_d^{(K)} = \tau_{\zeta_d} + \kappa \psi_{\zeta_d} \tag{15}$$

as the conditioning value of the first and the last knots, i.e., $x^{(1)}$ and $x^{(K)}$ in the spline definition, where κ was set to 2 in our experiments and we have assumed that each dimension of the continuous-valued conditioning parameter follows a Gaussian distribution whose mean τ_{ζ_d} and standard deviation ψ_{ζ_d} can be estimated from the training data. Since $\zeta_d^{(1)}$ and $\zeta_d^{(K)}$ are independent of the Gaussian components, they can be shared across all Gaussian components.

III. Training Algorithms

The S-VPHMM parameters can be estimated using either the ML criterion or the discriminative training criteria and can be initialized by copying $\mu_{i,l,d}^{(0)}$ and $\Sigma_{i,l,d}^{(0)}$ from the conventional HMM, setting $\mu_{\varpi(i,l,d)}^{(1)},\ldots,\mu_{\varpi(i,l,d)}^{(K_\mu)}$ to zero, and setting $\Sigma_{\varpi(i,l,d)}^{(1)},\ldots,\Sigma_{\varpi(i,l,d)}^{(K_\Sigma)}$ to one. When learning the parameters, we first reestimate $\mu_{\varpi(i,l,d)}^{(1)},\ldots,\mu_{\varpi(i,l,d)}^{(K_\mu)}$, then $\Sigma_{\varpi(i,l,d)}^{(1)},\ldots,\Sigma_{\varpi(i,l,d)}^{(K_\Sigma)}$, and finally $\mu_{i,l,d}^{(0)}$ and $\Sigma_{i,l,d}^{(0)}$ with the rest of the parameters fixed in each training iteration.

A. Maximum-Likelihood Training

In the ML training, we estimate the S-VPHMM parameters using the expectation–maximization (EM) algorithm [8] with the auxiliary function

$$Q(\Lambda;\Lambda')=\sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{i=1}^{I}\sum_{l=1}^{L}\gamma_{i,l,r,s_r}(t)\log b_{i,l}(\mathbf{o}_{r,t},\zeta_{r,t}) \tag{16}$$

where Λ is the model parameter set to be estimated, Λ' is the current parameter set, s_r is the true label sequence of the rth utterance, as shown in (17) at the bottom of the page, and

$$\gamma_{i,l,r,s}(t)=p(q_{r,t}=i,l|\mathbf{o}_{r,t},\zeta_{r,t},s,\Lambda') \tag{18}$$

is the occupation probability of Gaussian mixture component l of state i, at time t in the rth utterance given the label sequence s and can be obtained through an efficient forward–backward algorithm.

Taking the derivative of $Q(\Lambda;\Lambda')$ with respect to $\mu_{i,l,d}^{(0)}$, $\mu_{\varpi(i,l,d)}^{(1)},\ldots,\mu_{\varpi(i,l,d)}^{(K_\mu)}$, $\Sigma_{i,l,d}^{(0)}$, and $\Sigma_{\varpi(i,l,d)}^{(1)},\ldots,\Sigma_{\varpi(i,l,d)}^{(K_\Sigma)}$ we can derive for the ML reestimation formulae. For conciseness in presenting the reestimation formulae we simplify $\zeta_{r,t,d}$ as ζ, $\varpi(i,l,d)$ as ϖ, $o_{r,t,d}$ as o, $\mu_{i,l,d}$ as μ, $\mu_{i',l',d}$ as μ_+, $\Sigma_{i,l,d}$ as Σ, $\Sigma_{i',l',d}$ as Σ_+, and $\gamma_{i,l,r,s_r}(t)$ as γ.

1) ML Reestimation of $\tilde{\mu}_{\varpi(i,l,d)}$:

$$\tilde{\mu}_{\varpi}=\mathbf{A}_{ML,\varpi}^{-1}\mathbf{B}_{ML\cdot\varpi} \tag{19}$$

where $\mathbf{A}_{ML,\varpi}$ is a matrix whose element at kth row and jth column is

$$\mathbf{A}_{ML,\varpi}^{(k,j)}=\sum_r\sum_t\sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}}\gamma\Sigma^{-1}(\zeta)\,\nu_{\varpi}^{(k)}(\zeta)\,\nu_{\varpi}^{(j)}(\zeta) \tag{20}$$

and $\mathbf{B}_{ML,\varpi}$ is a vector whose kth value is

$$B_{ML,\varpi}^{(k)}=\sum_r\sum_t\sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}}\gamma\Sigma^{-1}(\zeta)\times\left(o-\mu_+^{(0)}\right)\nu_{\varpi}^{(k)}(\zeta). \tag{21}$$

2) ML Reestimation of $\tilde{\Sigma}_{\varpi(i,l,d)}$: $\tilde{\Sigma}_{\varpi(i,l,d)}$ is trained using the Newton method

$$\tilde{\Sigma}_{\varpi}=\tilde{\Sigma}'_{\varpi}-\left(\mathbf{F}_{ML,\varpi}\right)^{-1}\mathbf{E}_{ML,\varpi} \tag{22}$$

where $\mathbf{F}_{ML,\varpi}$ is a matrix whose element at the kth row and the jth column is shown in (23) at the bottom of the page, and $\mathbf{E}_{ML,\varpi}$ is a vector whose kth value is

$$E_{ML,\varpi}^{(k)}=\sum_r\sum_t\sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}}\gamma\left(\Sigma_+^{(0)}\right)^{-1/2}\Sigma_+^{\prime-1/2}(\zeta)\times\left((o-\mu_+(\zeta))^2-\Sigma'_+(\zeta)\right)\varsigma_{\varpi}^{(k)}(\zeta). \tag{24}$$

$$\begin{aligned}b_{i,l}(\mathbf{o}_{r,t},\boldsymbol{\zeta}_{r,t})&=\frac{1}{(2\pi)^{D/2}\left|\boldsymbol{\Sigma}_{i,l}\left(\boldsymbol{\zeta}_{r,t}\right)\right|^{1/2}}\exp\left(-\frac{1}{2}\left(\mathbf{o}_{r,t}-\boldsymbol{\mu}_{i,l}\left(\boldsymbol{\zeta}_{r,t}\right)\right)^T\boldsymbol{\Sigma}_{i,l}^{-1}\left(\boldsymbol{\zeta}_{r,t}\right)\left(\mathbf{o}_{r,t}-\boldsymbol{\mu}_{i,l}\left(\boldsymbol{\zeta}_{r,t}\right)\right)\right)\\&=\frac{1}{(2\pi)^{D/2}}\prod_d\frac{1}{\Sigma_{i,l,d}\left(\zeta_{r,t,d}\right)^{1/2}}\exp\left(-\frac{1}{2}\left(o_{r,t,d}-\mu_{i,l,d}\left(\zeta_{r,t,d}\right)\right)^2\Sigma_{i,l,d}^{-1}\left(\zeta_{r,t,d}\right)\right)\end{aligned} \tag{17}$$

$$F_{ML,\varpi}^{(k,j)}=\sum_r\sum_t\sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}}\gamma\left(\Sigma_+^{(0)}\right)^{-1}\left((o-\mu_+(\zeta))^2+\Sigma'_+(\zeta)\right)\times\varsigma_{\varpi}^{(k)}(\zeta)\varsigma_{\varpi}^{(j)}(\zeta) \tag{23}$$

3) ML Reestimation of $\mu^{(0)}_{i,l,d}$ and $\Sigma^{(0)}_{i,l,d}$:

$$\mu^{(0)} = \frac{\sum_r \sum_t \gamma\,(o - u'\,(\zeta))\Sigma^{-1}\,(\zeta)}{\sum_r \sum_t \gamma\Sigma^{-1}\,(\zeta)} + \mu'^{(0)} \tag{25}$$

$$\Sigma^{(0)} = \Sigma'^{(0)} \frac{\sum_r \sum_t \left(\gamma(o - \mu\,(\zeta))^2\Sigma'^{-1}\,(\zeta)\right)}{\sum_r \sum_t \gamma}. \tag{26}$$

B. Discriminative Training

Compared to the original VPHMM proposed in [7], the S-VPHMM described in this paper can be relatively easily discriminatively trained. In this section, we derive a growth-transformation (GT) based minimum classification error (MCE) training algorithm [19], [20], which has been successfully applied in our earlier work on conventional HMM [27]–[29], for the S-VPHMM.

The criterion we aim to minimize is the smoothed average utterance error rate

$$L_{MCE}(\Lambda) = \frac{1}{R}\sum_{r=1}^{R} \frac{1}{1 + \exp\left(-\alpha g(\mathbf{o}_r, \Lambda)\right)} \tag{27}$$

where R is the total number of utterances and α is a smooth factor for the sigmoid function and was set to 1/60 in our experiments. If we denote $s_{r,1}$ as the top competing candidate label sequence in the N-best list, the discriminant function is

$$g(\mathbf{o}_r, \Lambda) = \log p(\mathbf{o}_r, s_{r,1}|\Lambda) - \log p(\mathbf{o}_r, s_r|\Lambda). \tag{28}$$

Note that, minimizing (27) is equivalent to maximizing

$$\begin{aligned} O_{MCE}\,(\Lambda) &= R\,(1 - L_{MCE}\,(\Lambda)) \\ &= \sum_{r=1}^{R} \frac{p^{\alpha}\,(\mathbf{o}_r, s_r|\Lambda)}{p^{\alpha}\,(\mathbf{o}_r, s_{r,1}|\Lambda) + p^{\alpha}\,(\mathbf{o}_r, s_r|\Lambda)} \end{aligned} \tag{29}$$

as shown in [19], [20], and [27]. By following the exact same steps as shown in [19], maximizing (29) can be reduced to the problem of optimizing the auxiliary function

$$\begin{aligned} U(\Lambda;\Lambda') = &\sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{i=1}^{I}\sum_{l=1}^{L} \Delta\gamma(i,l,r,t) \\ &\times \log b_{i,l}(\mathbf{o}_{r,t}, \zeta_{r,t}) \\ &+ \sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{i=1}^{I}\sum_{l=1}^{L} \delta(r,t,i,l) \\ &\times \int_{\boldsymbol{\chi}_{r,t}} b'_{i,l}(\boldsymbol{\chi}_{r,t}, \zeta_{r,t}) \log b_{i,l}(\boldsymbol{\chi}_{r,t}, \zeta_{r,t}) d\boldsymbol{\chi}_{r,t} \end{aligned} \tag{30}$$

where

$$\begin{aligned} \Delta\gamma(i,l,r,t) = &p(s_r|\mathbf{o}_r,\Lambda')p(s_{r,1}|\mathbf{o}_r,\Lambda') \\ &\times \left[\gamma_{i,l,r,s_r}(t) - \gamma_{i,l,r,s_{r,1}}(t)\right] \end{aligned} \tag{31}$$

$$\begin{aligned} \delta(r,t,i,l) = &\beta p(s_r|\mathbf{o}_r,\Lambda') \\ &\left[p(s_r|\mathbf{o}_r,\Lambda')\gamma_{i,l,r,s_r}(t)\right. \\ &\left. + p(s_{r,1}|\mathbf{o}_r,\Lambda')\gamma_{i,l,r,s_{r,1}}(t)\right] \end{aligned} \tag{32}$$

and β is set to 2 in our experiments.

Taking the derivative of $U(\Lambda;\Lambda')$ with respect to $\mu^{(0)}_{i,l,d}, \mu^{(1)}_{\varpi(i,l,d)}, \ldots, \mu^{(K_\mu)}_{\varpi(i,l,d)}$, $\Sigma^{(0)}_{i,l,d}$, and $\Sigma^{(1)}_{\varpi(i,l,d)}, \ldots, \Sigma^{(K_\Sigma)}_{\varpi(i,l,d)}$ we can derive for the MCE reestimation formulae.

In addition to the simplifications mentioned before, we simplify $\Delta\gamma(i,l,r,t)$ as $\Delta\gamma$, and $\delta(r,t,i,l)$ as δ when presenting the reestimation formulae.

1) MCE Reestimation of $\tilde{\mu}_{\varpi(i,l,d)}$:

$$\tilde{\mu}_{\varpi} = \mathbf{A}^{-1}_{\mathrm{MCE},\varpi}\mathbf{B}_{\mathrm{MCE},\varpi} \tag{33}$$

where $\mathbf{A}_{\mathrm{MCE},\varpi}$ is a matrix whose element at the kth row and the jth column is

$$\begin{aligned} A^{(k,j)}_{\mathrm{MCE},\varpi} = \sum_r \sum_t \sum_{\substack{i',l',st. \\ \varpi(i',l',d)=\varpi(i,l,d)}} &(\Delta\gamma + \delta) \\ &\times \Sigma^{-1}_{+}\,(\zeta)\,\nu^{(k)}_{\varpi}\,(\zeta)\,\nu^{(j)}_{\varpi}\,(\zeta), \end{aligned} \tag{34}$$

and $\mathbf{B}_{\mathrm{MCE},\varpi}$ is a vector whose kth value is see equation (35) at the bottom of the page

2) MCE Reestimation of $\tilde{\Sigma}_{\varpi(i,l,d)}$: $\tilde{\Sigma}_{\varpi}$ is trained using Newton's method

$$\tilde{\Sigma}_{\varpi} = \tilde{\Sigma}'_{\varpi} - \left(\mathbf{F}_{\mathrm{MCE},\varpi}\right)^{-1}\mathbf{E}_{\mathrm{MCE},\varpi} \tag{36}$$

$$\begin{aligned} B^{(k)}_{\mathrm{MCE},\varpi} = &\sum_r \sum_t \sum_{\substack{i',l',st. \\ \varpi(i',l',d)=\varpi(i,l,d)}} \Delta\gamma\Sigma^{-1}_{+}\,(\zeta)\left(o - \mu^{(0)}_{+}\right)\nu^{(k)}_{\varpi}\,(\zeta) \\ &+ \sum_r \sum_t \sum_{\substack{i',l',st. \\ \varpi(i',l',d)=\varpi(i,l,d)}} \delta\Sigma^{-1}_{+}\,(\zeta)\left(\mu'_{+}\,(\zeta) - \mu^{(0)}_{+}\right)\nu^{(k)}_{\varpi}\,(\zeta). \end{aligned} \tag{35}$$

where $\mathbf{F}_{\mathrm{MCE},\varpi}$ is a matrix whose element at the kth row and the jth column is shown in (37) at the bottom of the page, and $E_{\mathrm{MCE},\varpi}$ is a vector whose kth value is

$$E^{(k)}_{\mathrm{MCE},\varpi}=\sum_r\sum_t \sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}} \Delta\gamma\left(\Sigma^{(0)}_+\right)^{-1/2}\Sigma'^{-1/2}_+(\zeta) \times\left((o-\mu_+(\zeta))^2-\Sigma'_+(\zeta)\right)\varsigma^{(k)}_\varpi(\zeta). \tag{38}$$

3) MCE Reestimation of $\mu^{(0)}_{i,l,d}$ and $\Sigma^{(0)}_{i,l,d}$:

$$\mu^{(0)}=\frac{\sum_r\sum_t \Delta\gamma\,(o-u'(\zeta))\Sigma'^{-1}(\zeta)}{\sum_r\sum_t(\Delta\gamma+\delta)\,\Sigma'^{-1}(\zeta)}+\mu'^{(0)} \tag{39}$$

$$\Sigma^{(0)}=\Sigma'^{(0)}\frac{\sum_r\sum_t\left(\Delta\gamma(o-\mu(\zeta))^2\Sigma'^{-1}(\zeta)+\delta\right)}{\sum_r\sum_t\Delta\gamma+\delta}. \tag{40}$$

Note that although our reestimation formulae are derived for the MCE criterion, the same formulae may be used for the maximum mutual information (MMI) and minimum phone error (MPE) training with different ways of calculating the $\Delta\gamma(i,l,r,t)$ and $\delta(r,t,i,l)$. Interested readers can find additional information in [19].

Also note that the ML and MCE training algorithms described here for the means of the Gaussian components is similar in principle to the algorithms used in the CAT [14], [33] although the CAT is developed for a different purpose and only deals with the means.

IV. Parameter Clustering

The parametric formulations (5) and (6) allow for sharing the spline parameters across different Gaussian components. In fact, in the training algorithms described in Section III, the spline parameters are estimated for each regression class $\varpi(i,l,d)$ when the regression classes are known. In this section, we describe the algorithm used in the S-VPHMM to determine the regression classes.

Given the distribution of the domain $p(x)$, the distance between two general functions f_1 and f_2 can be defined as

$$d(f_1,f_2)=\int_x(f_1(x)-f_2(x))^2p(x)\,dx \tag{41}$$

which is also applicable to two splines (which are specific forms of general functions) determined by the evenly distributed knots

$$\left\{y^{(i)}_1|i=1,\ldots,K\right\} \tag{42}$$

and

$$\left\{y^{(i)}_2|i=1,\ldots,K\right\}. \tag{43}$$

Note that the calculation of the exact distance using (41) is time consuming. For this reason, we approximate the integration in (41) with the quantized summation of

$$d(f_1,f_2)=h\sum_{x=1}^{K}\left(y^{(i)}_1-y^{(i)}_2\right)^2p\left(x^{(i)}\right). \tag{44}$$

Since we have assumed that $p(x)=N(x,\tau,\psi^2)$ follows the Gaussian distribution determined by mean τ and variance ψ^2 as discussed in Section II, (44) can be rewritten as

$$d(f_1,f_2)=h\sum_{x=1}^{K}\left(y^{(i)}_1-y^{(i)}_2\right)^2\frac{1}{\psi\sqrt{2\pi}}\exp\left(-\frac{\left(x^{(i)}-\tau\right)^2}{2\psi^2}\right). \tag{45}$$

Note that the parameters h, τ, and ψ are the same for all the splines to be clustered, and

$$h=\frac{2\kappa\psi}{K-1} \tag{46}$$

$$x^{(i)}=x^{(1)}+(i-1)\,h \tag{47}$$

$$\mu=x^{(1)}+\frac{K-1}{2}h. \tag{48}$$

We thus simplify (45) to

$$\begin{aligned} d(f_1,f_2)&\propto\sum_{x=1}^{K}\left(y^{(i)}_1-y^{(i)}_2\right)^2\exp\left(-\frac{\left(x^{(i)}-\tau\right)^2}{2\psi^2}\right)\\ &=\sum_{x=1}^{K}\left(y^{(i)}_1-y^{(i)}_2\right)^2\exp\left(-\frac{h^2\left(i-1-\frac{K-1}{2}\right)^2}{2\psi^2}\right)\\ &=\sum_{x=1}^{K}\left(y^{(i)}_1-y^{(i)}_2\right)^2\exp\left(-\frac{2\kappa^2\left(i-\frac{K+1}{2}\right)^2}{(K-1)^2}\right). \end{aligned} \tag{49}$$

Note that our essential goal is to minimize the distance between the conditioning-parameter-dependent means and variances before and after the spline sharing. For this reason, when applying (49) to the variance splines, we replace $y^{(i)}_1$ and $y^{(i)}_2$ with $\log y^{(i)}_1$ and $\log y^{(i)}_2$, respectively.

$$F^{(k,j)}_{\mathrm{MCE},\varpi}=\sum_r\sum_t\sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}}\left[\Delta\gamma\left(\Sigma^{(0)}_+\right)^{-1}\left((o-\mu_+(\zeta))^2+\Sigma'_+(\zeta)\right)+2\delta\left(\Sigma^{(0)}_+\right)^{-1}\Sigma'_+(\zeta)\right]\varsigma^{(k)}_\varpi(\zeta)\,\varsigma^{(j)}_\varpi(\zeta) \tag{37}$$

Given the distance between two splines, we used the well-known k-means clustering algorithm to determine the regression classes. The number of clusters was predetermined based on the constraint on the number of parameters.

Please note that the conventional clustering methods for the Gaussian components based on the decision trees and tri-phone models can also be used to determine the regression classes for the splines and may be equally effective. However, to use the conventional methods we need to assume that the means and variances of the similar Gaussian components (determined by the conventional method) change in the same manner and all the dimensions of the splines cluster in the same way.

V. Empirical Analysis

To understand the strengths and weaknesses of the S-VPHMM, we have run a set of experiments on the Aurora-3 corpus. We focus on scenarios that use instantaneous conditioning parameters, which were not naturally supported in the original VPHMM and open the door to new opportunities for performance improvement.

A. Experiment Settings

The Aurora-3 corpus contains noisy digit recordings under realistic automobile environments. Each utterance in the Aurora-3 corpus is recorded under high-noise, low-noise, or quiet environment, with either a close-talk microphone or a hands-free, far-field microphone.

The corpus can be further separated into four subtasks based on the languages. There are three experimental settings for each language: In the *well-matched* (WM) condition, both the training and testing sets contain all combinations of noise environments and microphones. In the *mid-mismatched* (MM) condition, the training set contains only the quiet and low noisy data, and the testing set contains the high noisy data. In the *high-mismatched* (HM) condition, the training set contains close-talk data from all noise classes, and the testing set contains high noise and low noise data recorded with the far-field microphone. The *well-matched* setting is equivalent to the multi-style training [22]. In the *mid-mismatched* condition, the mismatch is mainly caused by additive noise, while in the *high-mismatched* condition both channel distortion and additive noise contribute to the mismatch.

The 39-dimensional features used in our experiments consisted of the 13-dimensional (with energy and without C0) static MFCC features and their first and second-order derivatives. To test the effectiveness of the S-VPHMM with different features, we have conducted experiments using two feature extraction pipelines: Fig. 3 illustrates the basic feature with cepstral mean normalization (CMN) but without any noise suppressor. Fig. 4 shows the enhanced feature that with our MFCC-MMSE noise suppressor [29] applied. The parameters used in the MFCC-MMSE suppressor (such as smoothing factors and the size of the minimum tracking windows) is exactly the same as that used in [29].

To evaluate the S-VPHMM, we report two baselines in the experiments: the conventional HMM trained using the ML criterion and that trained using the minimum classification error (MCE) criterion. The ML baseline system was trained in the

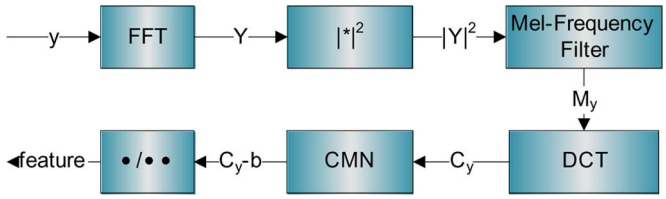


Fig. 3. Feature extraction pipeline for the basic feature with CMN but without any noise suppressor.

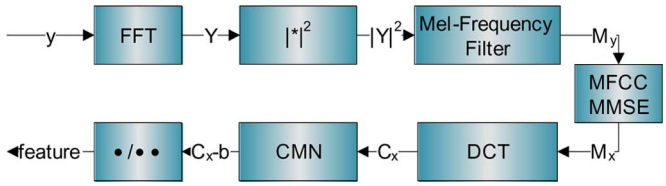


Fig. 4. Feature extraction pipeline for the enhanced feature on which the MFCC-MMSE noise suppressor is applied.

manner prescribed by the scripts included with the Aurora-3 task. On top of the ML baseline, eight iterations of MCE training were conducted and the best system was selected on the development set reserved from 10% of the training data. The system was then retrained using the full set of training data with the same number of iterations as the best system selected. The resulting system is the MCE-trained baseline.

The conventional HMMs used in our experiments are 3-Gaussians-per-state, 16-state, whole-word models for each digit in addition to the "sil" and "sp" models.

One of the key decisions to make in the S-VPHMM is the choice of the conditioning parameter $\zeta_{r,t,d}$. In our experiments $\zeta_{r,t,d}$ was chosen to be the dimension-wise instantaneous posterior SNR [29] in the cepstral domain for the static features

$$\begin{aligned}\zeta_{r,t,d} &= \sum_i a_{d,i} \log \frac{\sigma^2_{i,y,t}}{\sigma^2_{i,n,t}} \\ &= \sum_i a_{d,i} \left(\log \sigma^2_{i,y,t} - \log \sigma^2_{i,n,t}\right) \end{aligned} \tag{50}$$

where $a_{d,i}$ is the discrete cosine transformation (DCT) coefficient, $\sigma^2_{i,y,t}$ and $\sigma^2_{i,n,t}$ are the power of noisy signal and noise from the ith Mel-frequency filter at time t, respectively. The same dimension-wise instantaneous SNR of the static feature is used for the corresponding dynamic features. A minimum-controlled recursive moving-average noise tracker [4] was used in our system to track the noise power $\sigma^2_{i,n,t}$ with the same procedure and parameters used in our MFCC-MMSE noise suppresser work reported in [29]. In our experiments, the number of knots in the cubic spline is set to four, which is the smallest number of knots required for a cubic spline. We have also tried larger numbers and only seen little additional improvements, an indication that a four-knot spline is enough to approximate the nonlinear change pattern of the means and variances [7] for this task, although additional knots may be helpful for large vocabulary tasks. Note that, although the instantaneous SNR (50) was used in the experiments, our framework and algorithm can support other instantaneous SNRs such as the one proposed in [23], and other conditioning parameters such as the rate of speech.

TABLE I
COMPARISON BETWEEN THE CONVENTIONAL HMM AND THE S-VPHMM USING THE BASIC FEATURE AND WITH BOTH MEANS AND VARIANCES UPDATED

		Well	Mid	High	Average
Conventional HMM (ML)		6.87%	16.52%	31.11%	16.31%
S-VPHMM (ML) (Share All)	Abs.	6.69%	16.35%	31.21%	16.20%
	Rel.	2.55%	1.03%	-0.32%	0.64%
S-VPHMM (ML) (No Share)	Abs.	6.58%	16.24%	31.02%	16.07%
	Rel.	4.18%	1.68%	0.30%	1.44%
Conventional HMM (MCE)		6.47%	15.61%	31.33%	15.88%
S-VPHMM (MCE) (Share All)	Abs.	6.32%	15.36%	30.88%	15.62%
	Rel.	2.32%	1.60%	1.44%	1.64%
S-VPHMM (MCE) (No Share)	Abs.	5.50%	14.66%	30.33%	14.91%
	Rel.	14.99%	6.09%	3.19%	6.11%

TABLE II
COMPARISON BETWEEN THE CONVENTIONAL HMM AND THE S-VPHMM USING THE ENHANCED FEATURE AND WITHOUT VARIANCE PARAMETERS UPDATED

		Well	Mid	High	Average
Conventional HMM (ML)		5.08%	12.26%	23.26%	12.13%
S-VPHMM (ML) (Share All)	Abs.	4.96%	12.17%	23.12%	12.02%
	Rel.	2.32%	0.73%	0.59%	0.93%
S-VPHMM (ML) (No Share)	Abs.	4.91%	11.92%	23.01%	11.89%
	Rel.	3.30%	2.75%	1.05%	2.03%
Conventional HMM (MCE)		4.93%	11.80%	23.15%	11.89%
S-VPHMM (MCE) (Shall All)	Abs.	4.71%	11.58%	22.94%	11.67%
	Rel.	4.56%	1.93%	0.87%	1.85%
S-VPHMM (MCE) (No Share)	Abs.	4.12%	11.27%	22.31%	11.17%
	Rel.	16.43%	4.49%	3.63%	6.05%

TABLE III
COMPARISON BETWEEN THE CONVENTIONAL HMM AND THE S-VPHMM USING THE ENHANCED FEATURE AND WITH BOTH MEAN AND VARIANCE PARAMETERS UPDATED

		Well	Mid	High	Average
Conventional HMM (ML)		5.08%	12.26%	23.26%	12.13%
S-VPHMM (ML) (Share All)	Abs.	4.88%	12.09%	23.11%	11.96%
	Rel.	3.84%	1.37%	0.66%	1.44%
S-VPHMM (ML) (No Share)	Abs.	4.79%	11.93%	22.955	11.835
	Rel.	5.71%	2.67%	1.33%	2.54%
Conventional HMM (MCE)		4.69%	11.67%	22.92%	11.69%
S-VPHMM (MCE) (Shall All)	Abs.	4.51%	11.43%	22.76%	11.49%
	Rel.	3.84%	2.06%	0.70%	1.68%
S-VPHMM (MCE) (No Share)	Abs.	4.04%	11.20%	22.45%	11.15%
	Rel.	13.86%	4.03%	2.05%	4.64%

TABLE IV
SUMMARY OF THE NUMBER OF SPLINE CLUSTERS AND THE NUMBER OF PARAMETERS RELATIVE TO THAT USED IN THE CONVENTIONAL HMM FOR DIFFERENT SETTINGS

	# of Spline Clusters	# of Parameters (times)
Conventional HMM (MCE)	0	1.00
S-VPHMM (MCE) Setting 1	1	1.01
S-VPHMM (MCE) Setting 2	9	1.06
S-VPHMM (MCE) Setting 3	17	1.13
S-VPHMM (MCE) Setting 4	34	1.25
S-VPHMM (MCE) Setting 5	68	1.50
S-VPHMM (MCE) Setting 6	136	2.00
S-VPHMM (MCE) Setting 7	273	3.00
S-VPHMM (MCE) Setting 8	546	4.00

Since the instantaneous SNR and the means and variances at each operating point need to be estimated for each frame, the S-VPHMM takes more computational time than the conventional HMM with the same number of Gaussian components. Note that the estimation of the instantaneous SNR is a byproduct of the noise suppressor and its cost can be ignored compared to the time spent on the front-end processing. The additional cost is mainly introduced by the calculation of the spline, which takes about the same time as that needed to calculate a Gaussian component given that most of the spline matrices can be precalculated and cached (as discussed in Appendix A). The cost of the spline calculation can be further reduced when we cluster and share the splines across different Gaussian components. In fact, the total number of the parameters in the S-VPHMM is a good indicator of the total computational cost in the decoding phase. If the S-VPHMM has the same number of parameters as that of the conventional HMM, little computational overhead would be introduced. Note, however, the training time can be five times as that of the conventional HMM even if the model size is comparable because each iteration in the S-VPHMM consists of three steps (as shown in Section III), each of which takes similar computational resource as that required in one iteration of the conventional HMM training, and also because $\Sigma'^{-1}(\zeta)$ in the reestimation formulae cannot be factored and cancelled out, e.g., in (25) and (40).

B. ML and MCE Training

In this experiment, the S-VPHMM was trained using the ML and MCE criteria upon the ML and MCE-trained conventional HMMs respectively, and compared against the corresponding conventional HMMs.

Table I compares the conventional HMM with the S-VPHMM using the basic feature illustrated in Fig. 3. In the *share all* setting, one spline parameter set was shared by all the Gaussian components. In the *no share* setting, no spline parameters were shared. Improvements can be observed under both settings and especially under the *no share* setting where a 14.99% relative WER reduction against the MCE-trained conventional HMM has been achieved with the MCE-trained S-VPHMM. However, the gain obtained using the ML-trained S-VPHMM over the ML-trained conventional HMM is not significant and sometimes even negative. One possible explanation of this behavior is that in the MCE training the splines are adapted only if better discrimination between different classes can be achieved, while in the ML training the spline shapes are tuned to match the observed utterances and do not necessary provide additional discrimination ability especially if some phones (or words) dominate the corpus.

There is an interesting observation in Table I. Under the high-mismatched condition the MCE-trained conventional HMM underperformed the ML-trained conventional HMM because the discriminative training algorithms tend to minimize the empirical error rate on the training set and the gain may not be generalized to a highly mismatched test set. The fact that the MCE-trained S-VPHMM outperformed the ML and MCE-trained conventional HMM under all conditions suggests that the S-VPHMM does provide better modeling ability by allowing the HMM parameters change as functions of the instantaneous SNR.

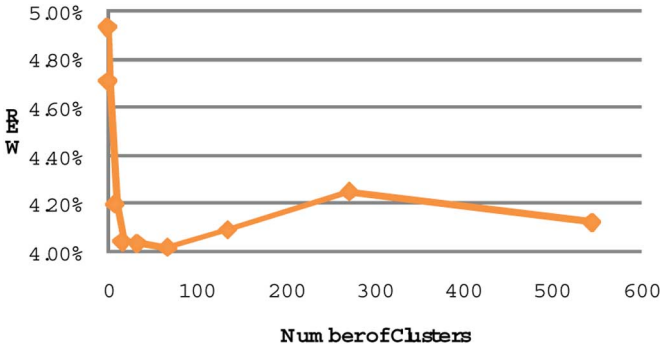


Fig. 5. Absolute WER under well-matched condition as a function of the number of spline clusters using the MCE-trained VPHMM without variance parameters updated.

Tables II and III summarize the experimental results on the Aurora-3 corpus with the enhanced feature.

From these tables, we observe that if only Gaussian mean is updated in the MCE training, the MCE-trained S-VPHMM reduced the WER by relative 6.05% and 16.43% on average and on the well-matched condition (which is the multi-style training), respectively, against the MCE-trained conventional HMM, or 7.91% and 18.9%, respectively, over the ML baseline. If both the means and variances are updated in the MCE training, the S-VPHMM achieved 4.64% and 13.86% relative WER reduction on average and under the well-matched condition respectively against the MCE-trained conventional HMM. This translates to 8.08% and 20.47% relative WER reduction respectively over the ML baseline. All the improvements under well-matched condition are statistically significant at the significance level of 1%. These results indicate that the S-VPHMM can achieve consistent relative WER reduction on different features and different parameter updating settings. The results also show that the effect of updating the variances is smaller than that of updating the means and that the relative gain achieved using the ML-trained S-VPHMM is much less compared to that achieved using the MCE-trained S-VPHMM under the well-matched condition.

Note that although the MCE-trained S-VPHMM outperforms the MCE-trained conventional HMM under all conditions, the gains under the mid-mismatched and high-mismatched conditions are less than that obtained under the well-matched condition. This is consistent with the intuition that some of the characteristics learned from the training set under *mismatched* conditions cannot be carried over to the test set. Further improvement under the mismatched conditions may be achieved if we can properly extrapolate the HMM parameters outside of the conditioning parameter's range that is observable from the training data.

C. Parameter Clustering

To show how the parameter sharing may affect the result, we have run experiments with different number of clusters, all with the enhanced feature and the MCE training criterion with which big gains have been observed. Table IV summarizes the number of spline clusters and the associated number of parameters used in different settings relative to the conventional HMM. Setting 1 is the setting where a single spline cluster is used by all the Gaussian components, and the Setting 8 is the setting where no spline is shared. Note that when a cubic spline is used by only one Gaussian component, the Gaussian component-specific mean and variance can be absorbed into the spline and this is the reason why only 4 times of the parameters in the conventional HMM are needed in Setting 8 although each spline has 4 knots.

TABLE V
SUMMARY OF THE ABSOLUTE WER AND THE RELATIVE WER REDUCTION ON THE AURORA-3 CORPUS WITH DIFFERENT PARAMETER SHARING SETTINGS USING THE MCE-TRAINED S-VPHMM WITHOUT VARIANCE PARAMETERS UPDATED

		Well	Mid	High	Average
Conventional HMM (MCE)		4.93%	11.80%	23.15%	11.89%
S-VPHMM (MCE) (Setting 1)	Abs.	4.71%	11.58%	22.94%	11.67%
	Rel.	4.56%	1.93%	0.87%	1.85%
S-VPHMM (MCE) (Setting 2)	Abs.	4.20%	11.07%	22.52%	11.18%
	Rel.	14.95%	6.23%	2.72%	5.97%
S-VPHMM (MCE) (Setting 3)	Abs.	4.04%	11.13%	22.79%	11.21%
	Rel.	18.09%	5.70%	1.52%	5.72%
S-VPHMM (MCE) (Setting 4)	Abs.	4.03%	11.12%	22.30%	11.08%
	Rel.	18.25%	5.78%	3.66%	6.82%
S-VPHMM (MCE) (Setting 5)	Abs.	4.01%	11.04%	22.57%	11.11%
	Rel.	18.65%	6.46%	2.47%	6.54%
S-VPHMM (MCE) (Setting 6)	Abs.	4.09%	10.99%	22.74%	11.17%
	Rel.	17.13%	6.91%	1.75%	6.09%
S-VPHMM (MCE) (Setting 7)	Abs.	4.25%	10.94%	22.57%	11.17%
	Rel.	13.94%	7.29%	2.48%	6.05%
S-VPHMM (MCE) (Setting 8)	Abs.	4.12%	11.27%	22.31%	11.17%
	Rel.	16.47%	4.53%	3.61%	6.06%

In these experiments, we first trained the S-VPHMM model for Setting 8 (no sharing). We then determine the regression classes using the clustering algorithm described in Section IV with the number of spline clusters predetermined according to Table IV. The S-VPHMM model with the specified number of spline clusters is then trained on top of the MCE-trained conventional HMM.

Table V summarizes the absolute WER and the relative WER reduction on the Aurora-3 corpus using the MCE-trained VPHMM without variance parameters updated. Fig. 5 illustrates how the absolute WER changes as a function of the number of spline clusters.

The curve in the Fig. 5 demonstrated some important relationship between the number of parameters and the recognition accuracy. When no spline is shared (Setting 8) the S-VPHMM obtained the absolute WER of 4.12% under the well-matched condition which outperforms the conventional HMM by relative WER reduction of 16.47% (statistically significant at the significance level of 1%). When 273 spline clusters (or equivalently 3 times of parameters) are used, the WER increases to 4.25%. However, as the number of spline clusters further decreases to 136, 68, 34 and 17, the WER decreases to 4.09%, 4.01%, 4.03% and 4.04%, respectively. Finally, when the number of spline clusters decreases to nine, the WER increases to 4.20%. As all Gaussian components share a single spline, the WER is dramatically increased and reaches 4.71%, which is still better than the MCE-trained conventional HMM by a 4.56% relative WER reduction. A similar pattern can be observed in Table VI and

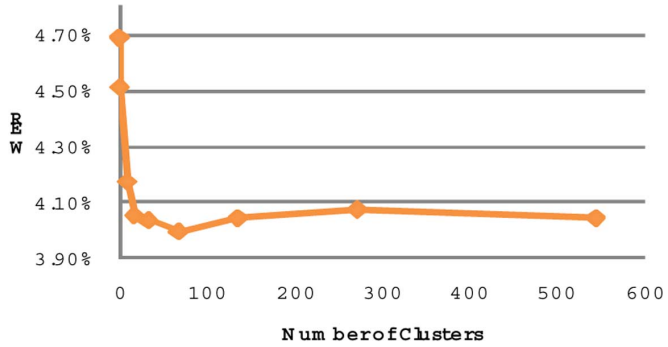


Fig. 6. Absolute WER under well-matched condition as a function of the number of spline clusters using the MCE-trained VPHMM with variance parameters updated.

TABLE VI
SUMMARY OF THE ABSOLUTE WER AND THE RELATIVE WER REDUCTION ON THE AURORA-3 CORPUS WITH DIFFERENT PARAMETER SHARING SETTINGS USING THE MCE-TRAINED VPHMM WITH VARIANCE PARAMETERS UPDATED

		Well	Mid	High	Average
Conventional HMM (MCE)		4.69%	11.67%	22.92%	11.69%
S-VPHMM (MCE) (Setting 1)	Abs.	4.51%	11.43%	22.76%	11.49%
	Rel.	3.84%	2.06%	0.70%	1.68%
S-VPHMM (MCE) (Setting 2)	Abs.	4.17%	11.30%	22.03%	11.13%
	Rel.	11.09%	3.17%	3.88%	4.79%
S-VPHMM (MCE) (Setting 3)	Abs.	4.05%	11.28%	22.07%	11.09%
	Rel.	13.65%	3.34%	3.71%	5.18%
S-VPHMM (MCE) (Setting 4)	Abs.	4.03%	11.16%	22.09%	11.04%
	Rel.	14.07%	4.37%	3.62%	5.56%
S-VPHMM (MCE) (Setting 5)	Abs.	3.99%	11.22%	21.93%	11.01%
	Rel.	14.93%	3.86%	4.32%	5.86%
S-VPHMM (MCE) (Setting 6)	Abs.	4.04%	11.14%	22.45%	11.13%
	Rel.	13.86%	4.54%	2.05%	4.82%
S-VPHMM (MCE) (Setting 7)	Abs.	4.07%	11.06%	22.74%	11.18%
	Rel.	13.22%	5.23%	0.79%	4.33%
S-VPHMM (MCE) (Setting 8)	Abs.	4.04%	11.20%	22.45%	11.15%
	Rel.	13.86%	4.03%	2.05%	4.64%

Fig. 6, where both the means and variances were updated during the MCE training.

This behavior is likely caused by the fact that two opposing factors are affecting the final result when the number of clusters is decreased: 1) the modeling ability becomes poorer since means and variances that share the same spline need to follow the same changing pattern; and 2) the spline parameters can be more reliably estimated as the same spline are shared by more Gaussian components. When the number of clusters decreases, the first factor outweighs the second one and the recognition accuracy drops. As the number of clusters further decreases, the second factor starts to show the effect and the recognition accuracy moves back. When the number of clusters continues to decrease, the effect of the second factor saturates and the effect of the first factor shows up again. For the Aurora-3 corpus, we can see that the S-VPHMM outperforms the MCE trained conventional HMM with 18.09% and 13.65% relative WER reduction without and with variance parameters updated, respectively, even if only 17 spline clusters are used (or equivalently, 1.13 times of parameters used in the conventional HMM). If the optimal number of clusters is chosen, the S-VPHMM can decrease the WER by relatively 18.65% and 14.93% without and with variance parameters updated, respectively, as indicated in Table VII. This happens when 68 clusters, or equivalently, 1.50 times of the conventional VPHMM parameters, were used.

TABLE VII
COMPARISON OF THE MCE-TRAINED S-VPHMM (SETTING 5) AND THE CONVENTIONAL HMM UNDER THE WELL-MATCHED CONDITION USING THE ENHANCED FEATURE WITH AND WITHOUT VARIANCE PARAMETERS UPDATED

		Conventional HMM	S-VPHMM (Setting 5)
Mean Update Only	Abs.	4.93%	4.01%
	Rel.	baseline	18.65%
With Variance Update	Abs.	4.69%	3.99%
	Rel.	baseline	14.93%

TABLE VIII
COMPARISON OF THE S-VPHMM AND THE CONVENTIONAL HMM WITH SIMILAR MODEL SIZE USING THE ENHANCED FEATURE AND WITH BOTH MEAN AND VARIANCE PARAMETERS UPDATED

		Well	Mid	High	Average
Conventional HMM (ML)		4.84%	11.75%	23.58%	11.94%
S-VPHMM (ML) (Setting 4)	Abs.	4.71%	11.77%	23.49%	11.80%
	Rel.	2.69%	-0.13%	0.37%	1.17%
Conventional HMM (MCE)		4.46%	11.39%	23.11%	11.55%
S-VPHMM (MCE) (Setting 4)	Abs.	4.03%	11.16%	22.09%	11.04%
	Rel.	9.64%	2.02%	4.41%	4.42%

D. Comparison With Same Model Size

As a contrast, we have evaluated the conventional HMM with four Gaussian mixtures per state, which has slightly more parameters than the S-VPHMM with 34 splines (setting 4) and slightly less parameter than the S-VPHMM with 68 splines (setting 5). Table VIII compares the S-VPHMM with the conventional HMM with the comparable number of parameters and both mean and variance parameters updated. From the table, we can see that although the relative gain is smaller, the MCE-trained S-VPHMM still outperforms the MCE-trained conventional HMM with comparable model size by about 9.64% under the well-matched condition. However, the gain with the ML training is very small.

VI. SUMMARY AND CONCLUSION

In this paper, we have proposed and presented a novel framework for VPHMM and described the related discriminative training and parameter clustering algorithms. The core of our framework is an improved formulation of the variances and the use of piecewise functions to represent the change of HMM parameters over the conditioning parameters. We demonstrated its effectiveness on the Aurora-3 corpus with both basic and enhanced features, and with different number of clusters. We have shown that the S-VPHMM can effectively support the use of the instantaneous conditioning parameter. We also demonstrated that S-VPHMM introduces no additional latency and can achieve significant accuracy improvement over the discriminatively-trained conventional HMM even with aggressive parameter clustering.

The ability of the S-VPHMM in supporting the use of instantaneous conditioning parameters and discriminative training algorithms opens doors to new opportunities for model design

since many different conditioning parameters such as instantaneous rate of speech may now be incorporated into this framework. Investigation of appropriate conditioning parameters that is effective and can be efficiently and reliably estimated is the direction of our future work.

APPENDIX
THEORY OF CUBIC SPLINE

Given K knots $\left\{\left(x^{(i)}, y^{(i)}\right) | i = 1, \ldots, K; x^{(i)} < x^{(i+1)}\right\}$ in the cubic spline, the value of a data point x can be estimated as

$$y = ay^{(j)} + by^{(j+1)} + c\frac{\partial^2 y}{\partial x^2}\bigg|_{x=x^{(j)}} + d\frac{\partial^2 y}{\partial x^2}\bigg|_{x=x^{(j+1)}} \quad (51)$$

where

$$a = \frac{x^{(j+1)} - x}{x^{(j+1)} - x^{(j)}} \quad c = \frac{1}{6}\left(a^3 - a\right)\left(x^{(j+1)} - x^{(j)}\right)^2 \quad (52)$$

$$b = 1 - a, \text{ and } d = \frac{1}{6}\left(b^3 - b\right)\left(x^{(j+1)} - x^{(j)}\right)^2 \quad (53)$$

are interpolation parameters, and $[x^{(j)}, x^{j+1}]$ is the section where the point x falls.

Note that a K -knot cubic spline requires $2K$ parameters: K parameters for $x^{(i)}$ and other K parameters for $y^{(i)}$. The number of parameters can be reduced to almost half by choosing evenly distributed $x^{(i)}$

$$h = x^{(j+1)} - x^{(j)} = x^{(k+1)} - x^{(k)} > 0, \quad \forall j, \quad k \in \{1, \ldots, K-1\} \quad (54)$$

since we only need to store $y^{(i)}$ given that $\left\{K, x^{(1)}, x^{(K)}\right\}$ can be shared across all splines defined for the same dimension of the conditioning parameter. Equation (51) can thus be rewritten as

$$y = \left(\mathbf{M}_x^T + \mathbf{Q}_x^T \mathbf{Z}^{-1} \mathbf{W}\right) \tilde{\mathbf{y}} \quad (55)$$

where

$$\tilde{\mathbf{y}} = \left[y^{(1)} \quad \cdots \quad y^{(K)}\right]^T, \quad (56)$$

$$\mathbf{M}_x = \begin{bmatrix} 0 & \cdots & \underbrace{a}_{j} & \underbrace{b}_{j+1} & \cdots & 0 \end{bmatrix}^T, \quad (57)$$

$$\mathbf{Q}_x = \begin{bmatrix} 0 & \cdots & \underbrace{c}_{j} & \underbrace{d}_{j+1} & \cdots & 0 \end{bmatrix}^T, \quad (58)$$

$$\mathbf{Z} = \frac{h}{6}\begin{bmatrix} 1 & 0 & 0 & \ldots & \ldots & \ldots & 0 \\ 1 & 4 & 1 & 0 & \ldots & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ldots & 0 & 1 & 4 & 1 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & \ldots & 0 & 1 & 4 & 1 \\ 0 & \ldots & \ldots & \ldots & 0 & 0 & 1 \end{bmatrix}, \text{ and} \quad (59)$$

$$\mathbf{W} = \frac{1}{h}\begin{bmatrix} 0 & 0 & 0 & \ldots & \ldots & \ldots & 0 \\ 1 & -2 & 1 & 0 & \ldots & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ldots & 0 & 1 & -2 & 1 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & \ldots & 0 & 1 & -2 & 1 \\ 0 & \ldots & \ldots & \ldots & 0 & 0 & 0 \end{bmatrix}. \quad (60)$$

It follows that

$$\frac{dy}{d\tilde{\mathbf{y}}} = \left(\mathbf{M}_x^T + \mathbf{Q}_x^T \mathbf{Z}^{-1} \mathbf{W}\right)^T. \quad (61)$$

Since a, b, c, d are functions of x, $\mathbf{M}_x$ and $\mathbf{Q}_x$ are also functions of x. However, $\mathbf{Z}^{-1}\mathbf{W}$ is independent of x. So it can be pre-calculated, stored, and shared across different splines, making it attractive computationally.

ACKNOWLEDGMENT

The authors would like to thank Dr. X. He in Microsoft Research for valuable discussions and help.

REFERENCES

[1] S. Axelrod, R. Gopinath, and P. Olsen, "Modeling with a subspace constraint on inverse covariances," in *Proc. Int. Conf. Spoken Lang. Process.*, 2002, pp. 2177–2180.
[2] M. Blanchet, J. Boudy, and P. Lockwood, "Environment adaptation for speech recognition in noise," in *Proc. EUSIPCO*, 1992, pp. 391–394.
[3] Y. Chung, "A data-driven model parameter compensation method for noise-robust speech recognition," *IEICE Trans. Inf. Syst.*, vol. E88-D, no. 3, pp. 432–434, 2005.
[4] I. Cohen and B. Berdugo, "Noise estimation by minima controlled recursive averaging for robust speech enhancement," *IEEE Signal Proc. Lett.*, vol. 9, no. 1, pp. 12–15, Jan. 2002.
[5] D. V. Compernolle, "Noise adaptation in a hidden Markov model speech recognition system," *Comput. Speech, Lang.*, vol. 3, no. 2, pp. 151–168, 1989.
[6] X. Cui and Y. Gong, "Variable parameter Gaussian mixture hidden Markov modeling for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2003, vol. 1, pp. 12–15.
[7] X. Cui and Y. Gong, "A study of variable-parameter Gaussian mixture hidden Markov modeling for noisy speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1366–1376, May 2007.
[8] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.
[9] L. Deng, D. Yu, and A. Acero, "Structured speech modeling," *IEEE Trans. Audio, Speech and Lang. Process..*, vol. 14, no. 5, pp. 1492–1504, Sep. 2006.
[10] L. Deng, D. Yu, and A. Acero, "Evaluation of a long-contextual-span hidden trajectory model and phonetic recognizer using A^* lattice search," in *Proc. Interspeech*, 2005, pp. 553–556.
[11] K. Fujinaga, M. Nakai, H. Shimodaira, and S. Sagayama, "Multiple-regression hidden Markov model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, pp. 513–516.
[12] S. Furui, "Toward robust, speech recognition under adverse conditions," in *Proc. ESCA Workshop Speech Process. Adverse Conditions*, 1992, pp. 31–41.
[13] M. Gales, "Model-based techniques for noise robust speech recognition," Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K., 1995.
[14] M. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 417–428, Jul. 2000.
[15] M. Gales and S. Young, "A fast and flexible implementation of parallel model combination," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1995, pp. 133–136.

[16] V. Goel, S. Axelrod, R. A. Gopinath, P. Olsen, and K. Visweswariah, "Discriminative estimation of subspace precision & mean (SPAM) models," *Proc. Eurospeech*, pp. 2617–2620, 2003.
[17] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Commun.*, vol. 16, pp. 261–291, 1995.
[18] Y. Gong, "Noise-dependent Gaussian mixture classifiers for robust rejection decision," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 2, pp. 57–64, Feb. 2002.
[19] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition—A unifying review for optimization-oriented speech recognition," *IEEE Signal Process. Mag.*, pp. 14–36, Sep. 2008.
[20] X. He, L. Deng, and W. Chou, "A novel learning method for hidden Markov models in speech and audio processing," in *Proc. Int. Workshop Multimedia Signal Process.*, 2006.
[21] C.-H. Lee, "On feature and model compensation approach to robust speech recognition," in *Proc. ESCA-NATO Workshop Robust Speech Recognition for Unknown Commun. Channels*, 1997, pp. 45–54.
[22] R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1987, pp. 705–708.
[23] R. Martin, "An efficient algorithm to estimate instantanous SNR of speech signals," in *Proc. Eur. Conf. Speech Commun. Technol.*, 1993, pp. 1093–1096.
[24] C. Mokbel and G. Chollet, "Speech recognition in adverse environments: speech enhancement and spectral transformations," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1991, pp. 925–928.
[25] S. Morii, T. Morii, and M. Hoshimi, "Noise robustness in speaker independent speech recognition," in *Proc. Int. Conf. Spoken Lang. Process.*, 1990, pp. 1145–1148.
[26] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 3, pp. 190–202, May 1996.
[27] D. Yu, L. Deng, X. He, and A. Acero, "Large-margin minimum classification error training for large-scale speech recognition tasks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2007, pp. 1137–1140.
[28] D. Yu, L. Deng, X. He, and A. Acero, "Use of incrementally regulated discriminative margins in MCE training for speech recognition," in *Proc. Interspeech*, 2006, pp. 2418–2421.
[29] D. Yu, L. Deng, X. He, and A. Acero, "Large-Margin minimum classification error training: A theoretical risk minimization perspective," *Comput. Speech, Lang.*, vol. 22, no. 4, pp. 415–429, 2008.
[30] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "Robust speech recognition using a cepstral minimum-mean-square-error- motivated noise suppressor," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 16, no. 5, pp. 1061–1070, Jul. 2008.
[31] D. Yu, L. Deng, Y. Gong, and A. Acero, "Discriminative training of variable-parameter HMMs for noise robust speech recognition," in *Proc. Interspeech*, 2008, pp. 285–288.
[32] D. Yu, L. Deng, Y. Gong, and A. Acero, "Parameter clustering and sharing in variable-parameter HMMs for noise robust speech recognition," in *Proc. Interspeech*, 2008, pp. 1253–1256.
[33] K. Yu and M. Gales, "Discriminative cluster adaptive training," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1694–1703, Sep. 2006.

Dong Yu (M'97–SM'06) received the B.S. degree (with honors) in electrical engineering from Zhejiang University, Hangzhou, China, the M.S. degree in electrical engineering from Chinese Academy of Sciences, Beijing, the M.S. degree in computer science from Indiana University at Bloomington, and the Ph.D. degree in computer science from University of Idaho, Moscow.

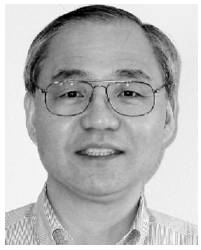
He joined Microsoft Research, Redmond, WA, in 1998 and Microsoft Speech Research Group in 2002, where he is a Researcher. His current research interests include speech processing, robust speech recognition, discriminative training, spoken dialog system, voice search technology, machine learning, and pattern recognition. He has published over 60 book chapters, journal and conference papers in these areas, and is the inventor/co-inventor of more than 30 awarded and pending patents.

Dr. Dong Yu is a member of ACM and ISCA. He is currently serving as an associate editor of the IEEE SIGNAL PROCESSING MAGAZINE.

Li Deng (S'85–M'86–SM'91–F'05) received the B.S. degree from the University of Science and Technology of China, Hefei, (with the Guo Mo-Ruo Award), and the Ph.D. degree from the University of Wisconsin-Madison (with the Jerzy E. Rose Award).

In 1989, he joined the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, as an Assistant Professor, where he became a Full Professor in 1996. From 1992 to 1993, he conducted sabbatical research at the 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 a Senior Researcher, where he is currently a Principal Researcher. He is also an Affiliate Professor in the Department of Electrical Engineering, University of Washington, Seattle. His past and current research activities include automatic speech and speaker recognition, statistical methods and machine learning, neural information processing, machine intelligence, audio and acoustic signal processing, statistical signal processing and digital communication, human speech production and perception, acoustic phonetics, auditory speech processing, auditory physiology and modeling, noise robust speech processing, speech synthesis and enhancement, spoken language understanding systems, multimedia signal processing, and multimodal human–computer interaction. In these areas, he has published over 300 refereed papers in leading international conferences and journals, 12 book chapters, and has given keynotes, tutorials, and lectures worldwide. He has been granted over 20 U.S. or international patents in acoustics, speech/language technology, and signal processing. He authored or coauthored three books in speech processing and learning.

Dr. Deng serves on the Board of Governors of the IEEE Signal Processing Society, and as Editor-in-Chief for the IEEE SIGNAL PROCESSING MAGAZINE. He is a Fellow of the Acoustical Society of America.



Yifan Gong (SM'93) received the B.Sc. degree from the Department of Communication Engineering, Southeast University, Nanjing, China, the M.Sc. degree in electrical engineering and instrumentation from the Department of Electronics, University of Paris, Paris, France, and the Ph.D. degree (with highest honors) in computer science from the Department of Mathematics and Computer Science, University of Henri Poincaré, Nancy, France.

He served the National Scientific Research Center (CNRS), Paris, France, and INRIA-Lorraine,Villerslès- Nancy, France, as Research Engineer and then joined CNRS as Senior Research Scientist. As Associate Lecturer, he taught computer programming and digital signal processing at the Department of Computer Science, University of Henri Poincaré. He also worked as Visiting Research Fellow at the Communications Research Center of Canada. He worked for Texas Instruments as Senior Member of Technical Staff at the Speech Technologies Laboratory. He developed speech modeling technologies robust against noisy environments, designed systems, algorithms, and software for speech and speaker recognition, and delivered memory- and CPU-efficient speech recognizers for mobile devices.

His research interests included mathematical models, software tools, and systems for signal processing, speech and speaker recognition, speech recognition in noisy conditions, and pattern recognition. He is currently a Senior Development Lead at Microsoft Research, Redmond, WA, leading a technology and software development team on speech technology and modeling. His current interests include developing technologies and speech recognition models for improved speech recognition performance across multiple languages for desktop, telephony, and mobile devices. He has authored over 100 publications in journals, IEEE Transactions, books, and conferences. His inventions have been awarded 18 U.S. patents. He is an Associate Editor of the *Pattern Recognition Journal*.

Dr. Gong served on the IEEE Signal Processing Society Speech Technical Committee from 1998 to 2002. He has been selected to give tutorials and other invited presentations in international conferences. He has been serving as member of technical committee and session chair for many international conferences.

Alex Acero (S'85–M'90–SM00–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 from 1990 to 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 200 technical papers. He holds 53 U.S. patents.

Dr. Acero has served the IEEE Signal Processing Society as Vice President Technical Directions (2007–2009), 2006 Distinguished Lecturer, as a member of the Board of Governors (2004–2005), as an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS (2003–2005) and the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING (2005–2007), and as a member of the editorial board of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING (2006–2008) and 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 ICASSP'98, Sponsorship Chair of the 1999 IEEE Workshop on Automatic Speech Recognition and Understanding, and General Co-Chair of the 2001 IEEE Workshop on Automatic Speech Recognition and Understanding. Since 2004, Dr. Acero, 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. Dr. Acero is member of the editorial board of Computer Speech and Language and he served as member of Carnegie Mellon University Dean's Leadership Council for College of Engineering.