interspeech 2008 incorporating SST 08

Discriminative Training of Variable-Parameter HMMs for Noise Robust Speech Recognition

Dong Yu, Li Deng, Yifan Gong, Alex Acero

Microsoft Corporation, Redmond, WA, USA

{dongyu, deng, ygong, alexac}@microsoft.com

Abstract

We propose a new type of variable-parameter hidden Markov model (VPHMM) whose mean and variance parameters vary each as a continuous function of additional environment-dependent parameters. Different from the polynomial-function-based VPHMM proposed by Cui and Gong (2007), the new VPHMM uses cubic splines to represent the dependency of the means and variances of Gaussian mixtures on the environment parameters. Importantly, the new model no longer requires quantization in estimating the model parameters and it supports parameter sharing and instantaneous conditioning parameters directly. We develop and describe a growth-transformation algorithm that discriminatively learns the parameters in our cubic-spline-based VPHMM (CS-VPHMM), and evaluate the model on the Aurora-3 corpus with our recently developed MFCC-MMSE noise suppressor applied. Our experiments show that the proposed CS-VPHMM outperforms the discriminatively trained and maximum-likelihood trained conventional HMMs with relative word error rate (WER) reduction of 14% and 20% respectively under the well-matched conditions when both mean and variances are updated.

Index Terms: speech recognition, variable-parameter hidden Markov model, discriminative training, cubic spline, growth transformation

1. Introduction

Automatic speech recognition (ASR) under noisy environments continues to be an active research area. Recently, Cui and Gong [2] proposed a new model, named variable-parameter hidden Markov model (VPHMM), for robust ASR. Different from the conventional hidden Markov model (HMM), the means and variances of the Gaussian mixtures in the VPHMM change as functions of some environment-dependant conditioning parameters such as signal-to-noise ratio (SNR).

In the conventional HMM, the continuous observation density function $b_i(\mathbf{x}_{r,t})$ for state i and acoustic observation $\mathbf{x}_{r,t}$ at frame t in the utterance r is

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

where the probability is estimated using a mixture of L Gaussian components, $N(\mathbf{x}_{r,t} \mid \boldsymbol{\mu}_{i,l}, \boldsymbol{\Sigma}_{i,l})$ is the l-th Gaussian mixture component with fixed $\boldsymbol{\mu}_{i,l}$ and $\boldsymbol{\Sigma}_{i,l}$, $w_{i,l}$ is a positive weight for the l-th Gaussian component, and $\sum_{l=1,\ldots,L} w_{i,l} = 1$. In the VPHMM, $\boldsymbol{\mu}_{i,l}$ and $\boldsymbol{\Sigma}_{i,l}$ become functions of some environment-dependent parameter set $\boldsymbol{\zeta}$, i.e.,

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

In this VPHMM, it is assumed that the parameter $\boldsymbol{\zeta}$ can be easily and reliably estimated and that the functions $\boldsymbol{\mu}_{i,l}(\boldsymbol{\zeta})$ and $\boldsymbol{\Sigma}_{i,l}(\boldsymbol{\zeta})$ can be learned from the training data.

While non-parametric methods might be used, $\boldsymbol{\mu}_{i,l}(\boldsymbol{\zeta})$ and $\boldsymbol{\Sigma}_{i,l}(\boldsymbol{\zeta})$ are usually constructed with a parametric approach. For example, in the original work of [2], Cui and Gong used the polynomial regression function over the utterance SNR to represent $\boldsymbol{\mu}_{i,l}(\boldsymbol{\zeta})$ and $\boldsymbol{\Sigma}_{i,l}(\boldsymbol{\zeta})$, and used the maximum likelihood (ML) algorithm to estimate the parameters in the polynomial functions. Specifically, a diagonal covariance matrix is assumed in their model. The means and variances in the d-th dimension are determined by

$$\mu_{i,l,d}(\zeta_{r,t,d}) = \xi(\zeta_{r,t,d} \mid \mu_{i,l,d}^{(1)}, \cdots, \mu_{i,l,d}^{(K)}), \text{ and} \quad (3)$$

$$\Sigma_{i,l,d}(\zeta_{r,t,d}) = \Sigma_{i,l,d}^{(0)} e^{\xi(\zeta_{r,t,d} \mid \Sigma_{i,l,d}^{(1)}, \cdots, \Sigma_{i,l,d}^{(K)})}, \quad (4)$$

where $\mu_{i,l,d}^{(1)}, \cdots, \mu_{i,l,d}^{(K)}$ and $\Sigma_{i,l,d}^{(1)}, \cdots, \Sigma_{i,l,d}^{(K)}$ are polynomial parameters for the means and variances, respectively, and $\Sigma_{i,l,d}^{(0)}$ is the original variance in the conventional HMM. Note that (4) is chosen to guarantee $\Sigma_{i,l,d}(\zeta_{r,t,d}) > 0$. Cui and Gong [2] showed positive results on the Aurora-2 corpus using the polynomial-function-based VPHMM.

Several questions remain to be answered following the work of [2]. First, is there an alternative functional form to (3) and (4) that can be used to represent the means and variances so that quantization is not needed in estimating the model parameters? Second, is it possible to use instantaneous SNR instead of utterance SNR as proposed in [2] as the conditioning parameter? Third, is it possible to train the VPHMM parameters using discriminative methods instead of the ML one as adopted in [2]? Fourth, is it possible to train VPHMM directly from a single conventional HMM instead of from a set of them trained under quantized SNR conditions as proposed in [2]? Fifth, can we share the VPHMM parameters?

In this paper we aim to answer the above questions and to improve the earlier VPHMM. Specifically, we propose to approximate $\boldsymbol{\mu}_{i,l}(\boldsymbol{\zeta})$ and $\boldsymbol{\Sigma}_{i,l}(\boldsymbol{\zeta})$ with a new, cubic-spline-based parameterization form that supports parameter sharing and to train the parameters discriminatively. We show that instantaneous SNR can be used as the conditioning parameter. We demonstrate the effectiveness of our cubic-spline-based VPHMM (CS-VPHMM) on the Aurora-3 corpus with our recently developed Mel-frequency cepstral minimum mean square error (MFCC-MMSE) motivated noise suppressor [5] applied. Our experiments show that CS-VPHMM outperforms the discriminatively trained and ML trained conventional HMMs with relative word error rate (WER) reduction of 14% and 20% respectively under the well-matched conditions when both mean and variances are updated.

The rest of the paper is organized as follows. In Section 2,

we introduce the parameterization form used in CS-VPHMM. In Section 3, we discuss the estimation of the environment conditioning parameters. In Section 4, we describe the discriminative training algorithm for CS-VPHMM. We report our experimental results in Section 5 and conclude the paper in Section 6.

2. Cubic-Spline-Based VPHMM

In this section, we introduce a new, cubic-spline-based parameterization form for VPHMM. We assume that covariance matrices are diagonal and each dimension *d* of the mean and variance vector can be approximated with a cubic spline ξ as

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

$$\Sigma_{i,l,d}\left(\zeta_{r,t,d}\right)=\Sigma^{(0)}_{i,l,d}\xi^{-2}\left(\zeta_{r,t,d}\mid\Sigma^{(1)}_{\varpi(i,l,d)},\cdots,\Sigma^{(K)}_{\varpi(i,l,d)}\right), \tag{6}$$

where $\mu^{(0)}_{i,l,d}$ and $\Sigma^{(0)}_{i,l,d}$ are the Gaussian-component-specific mean and variance, $\mu^{(1)}_{\varpi(i,l,d)},\cdots,\mu^{(K)}_{\varpi(i,l,d)}$ and $\Sigma^{(1)}_{\varpi(i,l,d)},\cdots,\Sigma^{(K)}_{\varpi(i,l,d)}$ are the spline knots that can be shared across different Gaussian components, and $\varpi(i,l,d)$ is the regression class. Note that (6) is different from (4) and can lead to a significantly simplified re-estimation formula.

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

$$y=ay^{(j)}+by^{(j+1)}+c\frac{\partial^2 y}{\partial x^2}\Big|_{x=x^{(j)}}+d\frac{\partial^2 y}{\partial x^2}\Big|_{x=x^{(j+1)}}, \tag{7}$$

where

$$a=\frac{x^{(j+1)}-x}{x^{(j+1)}-x^{(j)}},\ c=\frac{1}{6}\left(a^3-a\right)\left(x^{(j+1)}-x^{(j)}\right)^2, \tag{8}$$

$$b=1-a\text{, and }d=\frac{1}{6}\left(b^3-b\right)\left(x^{(j+1)}-x^{(j)}\right)^2 \tag{9}$$

are interpolation parameters, and $\left[x^{(j)},x^{(j+1)}\right]$ is the section where the point *x* falls. If $x^{(j)}$ are evenly distributed with

$$h=x^{(j+1)}-x^{(j)}=x^{(k+1)}-x^{(k)}>0,\forall j,k\in\left\{1,\cdots,K-1\right\} \tag{10}$$

and natural spline is used, (7) can be rewritten as

$$y=\left(\mathbf{E}_x^T+\mathbf{F}_x^T\mathbf{C}^{-1}\mathbf{D}\right)\tilde{\mathbf{y}}, \tag{11}$$

where

$$\tilde{\mathbf{y}}=\begin{bmatrix}y^{(1)} & \cdots & y^{(K)}\end{bmatrix}^T, \tag{12}$$

$$\mathbf{E}_x=\begin{bmatrix}0 & \cdots & \underbrace{a}_{j} & \underbrace{b}_{j+1} & \cdots & 0\end{bmatrix}^T, \tag{13}$$

$$\mathbf{F}_x=\begin{bmatrix}0 & \cdots & \underbrace{c}_{j} & \underbrace{d}_{j+1} & \cdots & 0\end{bmatrix}^T, \tag{14}$$

$$\mathbf{C}=\frac{h}{6}\begin{bmatrix}1&0&0&\cdots&\cdots&\cdots&0\\1&4&1&0&\cdots&\cdots&0\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\\cdots&0&1&4&1&0&\cdots\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\0&\cdots&\cdots&0&1&4&1\\0&\cdots&\cdots&\cdots&0&0&1\end{bmatrix}, \tag{15}$$

$$\mathbf{D}=\frac{1}{h}\begin{bmatrix}0&0&0&\cdots&\cdots&\cdots&0\\1&-2&1&0&\cdots&\cdots&0\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\\cdots&0&1&-2&1&0&\cdots\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\0&\cdots&\cdots&0&1&-2&1\\0&\cdots&\cdots&\cdots&0&0&0\end{bmatrix}. \tag{16}$$

It follows that

$$\frac{dy}{d\tilde{\mathbf{y}}}=\left(\mathbf{E}_x^T+\mathbf{F}_x^T\mathbf{C}^{-1}\mathbf{D}\right)^T. \tag{17}$$

Since a,b,c,d are functions of x, $\mathbf{E}_x$ and $\mathbf{F}_x$ are also functions of x. However, $\mathbf{C}^{-1}\mathbf{D}$ is independent of *x*. So it can be pre-calculated, stored, and shared across different splines, making it attractive computationally. Cubic spline comes with several other favorable features. First, the interpolation is smooth up to the second-order derivative. Second, the interpolation value depends only on the nearby data points (knots). Third, the interpolation accuracy can be improved by increasing the number of knots.

By noting

$$\tilde{\mu}_{\varpi(i,l,d)}=\begin{bmatrix}\mu^{(1)}_{\varpi(i,l,d)} & \cdots & \mu^{(K)}_{\varpi(i,l,d)}\end{bmatrix}^T, \tag{18}$$

$$\nu^T_{\varpi(i,l,d)}\left(\zeta_{r,t,d}\right)=\mathbf{E}^T_{\zeta_{r,t,d}}+\mathbf{F}^T_{\zeta_{r,t,d}}\mathbf{C}^{-1}\mathbf{D}, \tag{19}$$

$$\tilde{\Sigma}_{\varpi(i,l,d)}=\begin{bmatrix}\Sigma^{(1)}_{\varpi(i,l,d)} & \cdots & \Sigma^{(K)}_{\varpi(i,l,d)}\end{bmatrix}^T\text{, and} \tag{20}$$

$$\varsigma^T_{\varpi(i,l,d)}\left(\zeta_{r,t,d}\right)=\mathbf{E}^T_{\zeta_{r,t,d}}+\mathbf{F}^T_{\zeta_{r,t,d}}\mathbf{C}^{-1}\mathbf{D}, \tag{21}$$

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

$$\mu_{i,l,d}\left(\zeta_{r,t,d}\right)=\mu^{(0)}_{i,l,d}+\nu^T_{\varpi(i,l,d)}\left(\zeta_{r,t,d}\right)\tilde{\mu}_{\varpi(i,l,d)}\text{, and} \tag{22}$$

$$\Sigma_{i,l,d}\left(\zeta_{r,t,d}\right)=\Sigma^{(0)}_{i,l,d}\left(\varsigma^T_{\varpi(i,l,d)}\left(\zeta_{r,t,d}\right)\tilde{\Sigma}_{\varpi(i,l,d)}\right)^{-2}. \tag{23}$$

3. Estimation of Conditioning Parameters

The environment-conditioning parameter used in our model and system is the instantaneous posterior SNR in the cepstral domain:

$$\zeta_d=\sum_i a_{d,i}\log\frac{\sigma^2_{i,y}}{\sigma^2_{i,n}}=\sum_i a_{d,i}\left(\log\sigma^2_{i,y}-\log\sigma^2_{i,n}\right), \tag{24}$$

where $a_{d,i}$ is the inverse discrete cosine transformation (IDCT) coefficient, $\sigma^2_{i,y}$ and $\sigma^2_{i,n}$ are the power of noisy signal and noise from the *i*-th Mel-frequency filter, respectively. The noise power $\sigma^2_{i,n}$ is estimated using a minimum-controlled recursive moving-average noise tracker similar to the one described in [1] and is estimated with the procedure developed in our MFCC-MMSE noise suppresser as reported in [5].

The cubic spline is appropriate for interpolation but not for extrapolation. For this reason, we need to determine the start and end points for the spline knots. In our CS-VPHMM, we have assumed that each dimension of the conditioning parameter follow a Gaussian distribution whose mean μ_{ζ_d} and standard deviation σ_{ζ_d} can be estimated from the training data. We then set $\left[\mu_{\zeta_d}-2\sigma_{\zeta_d},\mu_{\zeta_d}+2\sigma_{\zeta_d}\right]$ as the range for spline interpolation.

4. Discriminative Training of VPHMM

In this section, we describe a growth-transformation (GT) based discriminative training algorithm [1][4] for the CS-VPHMM. In the initialization stage, $\mu_{i,l,d}^{(0)}$ and $\Sigma_{i,l,d}^{(0)}$ are copied from the conventional HMM, $\mu_{\varpi(i,l,d)}^{(1)},\cdots,\mu_{\varpi(i,l,d)}^{(K)}$ are set to zero, and $\Sigma_{\varpi(i,l,d)}^{(1)},\cdots,\Sigma_{\varpi(i,l,d)}^{(K)}$ are set to one. If we keep $\mu_{\varpi(i,l,d)}^{(1)},\cdots,\mu_{\varpi(i,l,d)}^{(K)}$ and $\Sigma_{\varpi(i,l,d)}^{(1)},\cdots,\Sigma_{\varpi(i,l,d)}^{(K)}$ at their initial values, CS-VPHMM becomes equivalent to the conventional HMM. In each discriminative training iteration, we first reestimate $\mu_{\varpi(i,l,d)}^{(1)},\cdots,\mu_{\varpi(i,l,d)}^{(K)}$, then $\Sigma_{\varpi(i,l,d)}^{(1)},\cdots,\Sigma_{\varpi(i,l,d)}^{(K)}$, and finally $\mu_{i,l,d}^{(0)}$ and $\Sigma_{i,l,d}^{(0)}$ with the rest of the parameters fixed.

Before presenting the training algorithm, we denote

$$d(r,t,i,l)=\sum_s d'(s)p(q_{r,t}=i,l\mid s,\Lambda'), \tag{25}$$

where Λ' is the current parameter set, $q_{r,t}$ is the state at time t in the r-th utterance, s is the label sequence, and $d'(s)$ is a parameter to control the convergence speed [1][4]. We denote the occupation probability of Gaussian mixture component l of state i, at time t in the r-th utterance as

$$\gamma_{i,l,r,s_r}(t)=p(q_{r,t}=i,l\mid X_r,s_r,\Lambda'), \tag{26}$$

which can be obtained through an efficient forward-backward algorithm. Further, we define

$$\Delta\gamma(i,l,r,t)=\sum_s p(s\mid X,\Lambda')\big(C(s)-O(\Lambda')\big)\gamma_{i,l,r,s_r}(t), \tag{27}$$

where $O(\Lambda')$ is the discriminative training criterion and $C(s)$ is the weighting factor. For conciseness in presenting the re-estimation formulas, in the rest of this section we simplify $\zeta_{r,t,d}$ as ζ, $\varpi(i,l,d)$ as ϖ, $x_{r,t,d}$ as x, $\mu_{i,l,d}$ as μ, $\Sigma_{i,l,d}$ as Σ, $\Delta\gamma(i,l,r,t)$ as $\Delta\gamma$, and $d(r,t,i,l)$ as d as long as no confusion is introduced. We have omitted the derivation of the following re-estimation formulas due to the space limit.

4.1. Reestimation of $\mu_{i,l,d}^{(0)}$ and $\Sigma_{i,l,d}^{(0)}$

$$\mu_{i,l,d}^{(0)}=\frac{\sum_r\sum_t\Delta\gamma\big(x-u_{\varpi}'(\zeta)\big)\Sigma^{-1}(\zeta)}{\sum_r\sum_t(\Delta\gamma+d)\Sigma^{-1}(\zeta)}+\mu_{i,l,d}'^{(0)}, \tag{28}$$

$$\Sigma_{i,l,d}^{(0)}=\Sigma_{i,l,d}'^{(0)}\frac{\sum_r\sum_t\big(\Delta\gamma(x-\mu(\zeta))^2\Sigma'^{-1}(\zeta)+d\big)}{\sum_r\sum_t\Delta\gamma+d}. \tag{29}$$

4.2. Reestimation of $\tilde{\mu}_{\varpi(i,l,d)}$

$$\tilde{\mu}_{\varpi(i,l,d)}=A_{\varpi(i,l,d)}^{-1}B_{\varpi(i,l,d)}, \tag{30}$$

where $A_{\varpi(i,l,d)}$ is a matrix whose element at the k-th row and the j-th column is

$$A_{\varpi(i,l,d)}^{(k,j)}=\sum_r\sum_t\sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}}(\Delta\gamma+d)\Sigma^{-1}(\zeta)\nu_{\varpi}^{(k)}(\zeta)\nu_{\varpi}^{(j)}(\zeta),$$

and $B_{\varpi(i,l,d)}$ is a vector whose k-th value is

$$B_{\varpi(i,l,d)}^{(k)}=\sum_r\sum_t\sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}}\Delta\gamma\Sigma^{-1}(\zeta)\big(x-\mu_{i',l',d}^{(0)}\big)\nu_{\varpi}^{(k)}(\zeta)$$
$$+\sum_r\sum_t\sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}}d\Sigma^{-1}(\zeta)\big(\mu'(\zeta)-\mu_{i',l',d}^{(0)}\big)\nu_{\varpi}^{(k)}(\zeta).$$

4.3. Reestimation of $\tilde{\Sigma}_{\varpi(i,l,d)}$

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

$$\tilde{\Sigma}_{\varpi(i,l,d)}=\tilde{\Sigma}'_{\varpi(i,l,d)}-\big(F_{\varpi(i,l,d)}\big)^{-1}E_{\varpi(i,l,d)}, \tag{31}$$

where $F_{\varpi(i,l,d)}$ is a matrix whose element at the k-th row and the j-th column is

$$F_{\varpi(i,l,d)}^{(k,j)}$$
$$=\sum_r\sum_t\sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}}[\Delta\gamma\big(\Sigma_{i,l,d}^{(0)}\big)^{-1}\Big(\big(x-\mu(\zeta)\big)^2+\Sigma'_{\varpi}(\zeta)\Big)$$
$$+2d\big(\Sigma_{i,l,d}^{(0)}\big)^{-1}\Sigma']\zeta_{\varpi}^{(k)}(\zeta)\zeta_{\varpi}^{(j)}(\zeta),$$

and $E_{\varpi(i,l,d)}$ is a vector whose k-th value is

$$E_{\varpi(i,l,d)}^{(k)}=\sum_r\sum_t\sum_{\substack{i',l',st.\\ \varpi(i',l',d)=\varpi(i,l,d)}}\Delta\gamma\big(\Sigma_{i,l,d}^{(0)}\big)^{-1/2}\Sigma'^{-1/2}(\zeta)$$
$$\Big(\big(x-\mu(\zeta)\big)^2-\Sigma'(\zeta)\Big)\zeta_{\varpi}^{(k)}(\zeta). \tag{32}$$

5. Experiments

We have evaluated our CS-VPHMM on the Aurora-3 corpus. In this section, we describe the experimental setting and results.

5.1. Experimental Setup

The Aurora-3 corpus contains noisy digit recordings under realistic automobile environments. In the Aurora-3 corpus, each utterance is labeled as coming from either a high, low, or quiet noise environment, and as being recorded using a close-talk microphone or a hands-free, far-field microphone.

The Aurora-3 corpus consists of four separate digit recognition sub-tasks based on the languages. For each language, three experimental settings are defined for the evaluation: In the w*ell-matched* condition both the training and the testing sets contain all combinations of noise environments and microphones. In the *mid-mismatched* condition, the training set contains quiet and low noise data recorded using the far-field microphone, and the testing set contains the high noisy data recorded using the far-field microphone. In the *high-mismatched* condition, the training set contains close-talk data from all noise classes, and the testing set contains high noise and low noise far-field data. In the *mid-mismatched* condition, the mismatch is mainly caused by the additive noise, while in the *high-mismatched* condition both channel distortion and additive noise exist.

In this paper we report two baselines: 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 manner prescribed by the scripts included with the Aurora-3 task. On top of the ML baseline, 8 iterations of MCE training were conducted and the best system was selected on the development set reserved from 10 percent 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 HMMs used in our experiments consist of 6-mixture 16-state whole-word models for each digit in addition to the "sil" and "sp" models. The 39-dimensional features used in our experiments contain the 13-dimension (with energy and without C0) static MFCC features and their delta and delta-

delta features. The parameters (such as smoothing factors and the size of the minimum tracking windows) used for noise tracking is exactly the same as that used in [5].

The CS-VPHMMs were discriminatively trained (also using the MCE criterion) upon the MCE-trained conventional HMM. Due to the time complexity, we only ran four iterations of training and we report the result after the fourth iteration. In our experiments, the number of knots in the cubic spline is set to be four. To show how the parameter sharing may affect the result, we have run two sets of experiments with none of the parameters shared (*no share*) and with all the spline parameters shared (*share all*), respectively. In the *no share* setting, the number of total parameters is four times as many as the conventional HMM, while in the *share all* setting, the total number of parameters is only 1.008 times.

5.2. Experimental Results

Tables 1-4 summarize the experimental results on the Aurora-3 corpus with the MFCC-MMSE noise suppressor applied. Table 1 and Table 3 show the absolute WER without and with updating the variance parameters in the MCE training. Table 2 and Table 4 summarize the relative WER reduction against the corresponding MCE-trained conventional HMM baseline.

	Well	Mid	High	Average
Conventional HMM (ML)	5.08%	12.26%	23.26%	**12.13%**
Conventional HMM (MCE)	4.93%	11.80%	23.15%	**11.89%**
VPHMM (MCE) - Share All	4.71%	11.58%	22.94%	**11.67%**
VPHMM (MCE) - No Share	4.12%	11.27%	22.31%	**11.17%**

Table 1. *Absolute WER on Aurora-3 corpus (no variance update)*

	Well	Mid	High	Average
VPHMM (MCE) - Share All	4.56%	1.93%	0.87%	**1.85%**
VPHMM (MCE) - No Share	16.43%	4.49%	3.63%	**6.05%**

Table 2. *Relative WER reduction against the MCE trained* conventional *HMM (no variance update)*

	Well	Mid	High	Average
Conventional HMM (ML)	5.08%	12.26%	23.26%	**12.13%**
Conventional HMM (MCE)	4.69%	11.67%	22.92%	**11.69%**
VPHMM (MCE) - Share All	4.51%	11.43%	22.76%	**11.49%**
VPHMM (MCE) - No Share	4.04%	11.20%	22.45%	**11.15%**

Table 3. *Absolute WER on Aurora-3 corpus (with variance update)*

	Well	Mid	High	Average
VPHMM (MCE) - Share All	3.84%	2.06%	0.70%	**1.68%**
VPHMM (MCE) - No Share	13.86%	4.03%	2.05%	**4.64%**

Table 4. *Relative WER reduction against the MCE trained* conventional *HMM (with variance update)*

From these tables, we observe that if only Gaussian mean is updated in the MCE training, the MCE trained CS-VPHMM reduced the WER by 6.05% relatively on average and by 16.43% relatively on the well-matched condition against the MCE trained conventional HMM, or 7.91% and 18.9% respectively over the ML baseline which is better than SPLICE. If both the mean and variances are updated in the MCE training, the CS-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%.

To examine whether the CS-VPHMM can improve the recognition accuracy on different features, we have also run experiments without the MFCC-MMSE noise suppressor and gained a 6.11% relative WER reduction on average, and 14.99% relative WER reduction on the well-matched condition over the MCE trained conventional HMM.

Note that although the CS-VPHMM outperforms the conventional HMM under all conditions, the largest gain is observed 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.

Also note that the CS-VPHMM with all spline parameters shared performs slightly better than the MCE-trained conventional HMM. A compromise can be made between the number of parameters shared and the performance. In our companion paper [6], we show that the same performance can be retained and even exceeded with only 1.13 times of parameters if proper parameter tying is conducted.

6. Conclusions

In this paper, we have presented a cubic-spline-based VPHMM and described the related discriminative training algorithm. We have shown that the CS-VPHMM can work effectively using the instantaneous SNR as the conditioning parameter. CS-VPHMM introduces no additional latency and can achieve significant accuracy improvement over the discriminatively trained conventional HMM.

Compared with the conventional HMM, CS-VPHMM tends to use many more parameters. To reduce the number of parameters, we can tie the splines since some of the parameters change in the same direction. Our framework supports the spline parameter sharing naturally. We report the clustering algorithm for optimal tying and the corresponding experimental results in the companion paper [6].

7. Acknowledgements

The authors would like to thank Dr. Xiaodong He, Dr. Jasha Droppo, Dr. Jian Wu, and Dr. Ye Tian at Microsoft Corporation for valuable discussions and assistance in conducting experiments.

8. References

[1] Cohen, I., and Berdugo. B., “Noise estimation by minima controlled recursive averaging for robust speech enhancement,” IEEE Signal Proc. Letters, Vol. 9, 2002, pp. 12-15.

[2] Cui, X. and Gong, Y., "A Study of Variable-Parameter Gaussian Mixture Hidden Markov Modeling for Noisy Speech Recognition”, IEEE Trans. On Audio, Speech, and Language Processing, Vol. 15, No. 4, May 2007, pp. 1366-1376.

[3] He, X., Deng, L., and Chou, W., "Discriminative Learning in Sequential Pattern Recognition - A Unifying Review for Optimization-Oriented Speech Recognition", IEEE Signal Processing Magazine, 2008 (to appear).

[4] He, X., Deng, L., and Chou, W., "A Novel Learning Method for Hidden Markov Models in Speech and Audio Processing", Proc. Intl. Workshop of Multimedia Signal Processing, 2006.

[5] Yu, D., Deng, L., Droppo, J., W, J., Gong, Y., and Acero, A., "Robust Speech Recognition Using a Cepstral Minimum-Mean-Square-Error-Motivated Noise Suppressor", IEEE Trans. on Audio, Speech and Language Processing, 2008 (to appear).

[6] Yu, D., Deng, L, Gong, Y. and Acero, A. “Parameter Clustering and Sharing in Variable-Parameter HMMs for Noise Robust Speech Recognition”, Interspeech 2008 (to appear).