interspeech 2008 incorporating SST 08

Parameter Clustering and Sharing in 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

Recently we proposed a cubic-spline-based variable-parameter hidden Markov model (CS-VPHMM) whose mean and variance parameters vary according to some cubic spline functions of additional environment-dependent parameters. We have shown good properties of the CS-VPHMM and demonstrated on the Aurora-3 corpus that MCE-trained CS-VPHMM greatly outperforms the MCE-trained conventional HMM at the cost of increased total number of model parameters. In this paper, we propose to share spline functions across different Gaussian mixture components to reduce the total number of model parameters and develop a clustering algorithm to do so. We demonstrate the effectiveness of our parameter clustering and sharing algorithm for the CS-VPHMM on Aurora-3 corpus and show that proper parameter sharing can reduce the number of parameters from 4 times of that used in the conventional HMM to 1.13 times and still get 18% relative WER reduction over the MCE trained conventional HMM under the well-matched condition. Effective parameter sharing makes the CS-VPHMM an attractive model for noise robustness.

Index Terms: speech recognition, variable-parameter hidden Markov model, cubic spline, parameter sharing, clustering

1. Introduction

Recently Cui and Gong [2] proposed a new model, named variable-parameter hidden Markov model (VPHMM), for robust automatic speech recognition (ASR). In their original VPHMM model the means and variances of the Gaussian mixtures change according to a polynomial function of some environment-dependant conditioning parameters such as signal-to-noise ratio (SNR).

We further advanced the technique with the cubic-spline-based VPHMM (CS-VPHMM) [6]. In the CS-VPHMM, the continuous observation density function $b_i(\mathbf{x}_{r,t},\boldsymbol{\zeta}_{r,t})$ for state *i*, acoustic observation $\mathbf{x}_{r,t}$ and the conditioning parameter $\boldsymbol{\zeta}_{r,t}$ at frame *t* in the utterance *r* is

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

where *L* is the number of Gaussian mixture components, $w_{i,l}$ is a positive weight for the *l*-th Gaussian component with the constraint $\sum_{l=1,\ldots,L} w_{i,l} = 1$, and $N(\mathbf{x}_{r,t} \mid \boldsymbol{\mu}_{i,l}\left(\boldsymbol{\zeta}_{r,t}\right), \boldsymbol{\Sigma}_{i,l}\left(\boldsymbol{\zeta}_{r,t}\right))$ is the *l*-th Gaussian mixture component whose mean and variance vary based on the conditioning parameter $\boldsymbol{\zeta}_{r,t}$. In our CS-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

$$\mu_{i,l,d}\left(\zeta_{r,t,d}\right) = \mu^{(0)}_{i,l,d} + \xi\left(\zeta_{r,t,d} \mid \mu^{(1)}_{\varpi(i,l,d)}, \cdots, \mu^{(K)}_{\varpi(i,l,d)}\right), \quad (2)$$

$$\Sigma_{i,l,d}\left(\zeta_{r,t,d}\right) = \Sigma^{(0)}_{i,l,d}\xi^{-2}\left(\zeta_{r,t,d} \mid \Sigma^{(1)}_{\varpi(i,l,d)}, \cdots, \Sigma^{(K)}_{\varpi(i,l,d)}\right), \quad (3)$$

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 (will be discussed in Section 2) that can be shared across different Gaussian mixture components, and $\varpi(i,l,d)$ is the regression class so that many different pairs of (i,l,d) may be mapped to the same regression class.

In our companion paper [6], we developed the discriminative training algorithm for the CS-VPHMM defined by (1), (2) and (3). We showed that the CS-VPHMM can use the dimension-wise instantaneous SNR as the conditioning parameter and so is much more flexible and powerful than the polynomial function based VPHMM proposed by Cui and Gong [2]. We also demonstrated on the Aurora-3 corpus that the discriminatively trained CS-VPHMM greatly outperforms the discriminatively trained conventional HMM both with and without our recently developed Mel-frequency cepstral minimum mean square error (MFCC-MMSE) motivated noise suppressor [5], esp. under the well-matched condition, at the cost of increased total number of model parameters.

In this paper, we explore the parameter sharing capability of the CS-VPHMM and answer the question whether it is possible to reduce the number of parameters in the CS-VPHMM without losing the gains achieved when no parameters are shared. We develop and describe a clustering algorithm to determine how the splines should be tied and report our experimental results on Aurora-3 corpus. We show that proper parameter sharing can reduce the number of parameters from 4 times of that used in the conventional HMM to 1.13 times and still get 18% relative WER reduction over the MCE trained conventional HMM under the well-matched condition. Effective parameter sharing makes the CS-VPHMM an attractive model for noise robustness.

The rest of the paper is organized as follows. In Section 2, we review some concepts related to the cubic spline and CS-VPHMM. In Section 3, we describe the detailed spline clustering algorithm. In Section 4, we report our experimental results on Aurora-3 with different degrees of parameter sharing and demonstrate the effectiveness of the clustering algorithm. We conclude the paper in Section 4.

2. Cubic Spline and CS-VPHMM

In this section, we briefly review some concepts related to the cubic spline and CS-VPHMM to set the background. Detailed information on the CS-VPHMM and the discriminative training algorithm used to estimate the model parameters can be found in our companion paper [6].

As mentioned in section 1, the mean and variance of each Gaussian mixture component in the CS-VPHMM vary

Copyright © 2008 ISCA

according to (2) and (3) given the conditioning parameter $\zeta_{r,t,d}$. The core of (2) and (3) is the cubic spline function ξ which is solely determined by the control pointes (knots) and the boundary conditions. There are two typical boundary conditions for the cubic spline: one that whose first derivative is zero and one that whose second derivative is zero. The spline with the latter boundary condition is usually called natural spline and is the one used in this study.

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

$$y=ay^{(j)}+by^{(j+1)}+c\frac{\partial^2 y}{\partial x^2}\Big|_{x=x^{(j)}}+d\frac{\partial^2 y}{\partial x^2}\Big|_{x=x^{(j+1)}}, \tag{4}$$

where

$$a=\frac{x^{(j+1)}-x}{x^{(j+1)}-x^{(j)}},\ c=\frac{1}{6}\left(a^3-a\right)\left(x^{(j+1)}-x^{(j)}\right)^2, \tag{5}$$

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

are interpolation parameters, and $\left[x^{(j)},x^{(j+1)}\right]$ is the section where the point x falls. Note that for a K-knot cubic spline, it requires $2K$ parameters: K parameters for $x^{(i)}$ and other K parameters for $y^{(i)}$. The number of parameters can be greatly reduced if we choose evenly distributed $x^{(j)}$

$$h=x^{(j+1)}-x^{(j)}=x^{(k+1)}-x^{(k)}>0,\forall j,k\in\left\{1,\cdots,K-1\right\} \tag{7}$$

since we only need to store $y^{(i)}$ and $\left\{K,x^{(1)},x^{(K)}\right\}$. Note that only one $\left\{K,x^{(1)},x^{(K)}\right\}$ is needed for each dimension of the conditioning parameter as will be clear later and so the average number of parameters needed for each spline is very close to K.

With evenly distributed knots, (5) and (6) can be simplified into

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

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

As we have indicated in our companion paper [6], (4) can be rewritten as

$$y=\left(\mathbf{E}_x^T+\mathbf{F}_x^T\mathbf{C}^{-1}\mathbf{D}\right)\tilde{\mathbf{y}}, \tag{10}$$

where

$$\tilde{\mathbf{y}}=\left[\begin{array}{ccc}y^{(1)} & \cdots & y^{(K)}\end{array}\right]^T, \tag{11}$$

$$\mathbf{E}_x=\left[\begin{array}{ccccccc}0 & \cdots & \underbrace{a}_{j} & \underbrace{b}_{j+1} & \cdots & 0\end{array}\right]^T, \tag{12}$$

$$\mathbf{F}_x=\left[\begin{array}{ccccccc}0 & \cdots & \underbrace{c}_{j} & \underbrace{d}_{j+1} & \cdots & 0\end{array}\right]^T, \tag{13}$$

$$\mathbf{C}=\frac{h}{6}\begin{bmatrix}1&0&0&\cdots&\cdots&\cdots&0\\1&4&1&0&\cdots&\cdots&0\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\\cdots&0&1&4&1&0&\cdots\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\0&\cdots&\cdots&0&1&4&1\\0&\cdots&\cdots&\cdots&0&0&1\end{bmatrix}, \tag{14}$$

$$\mathbf{D}=\frac{1}{h}\begin{bmatrix}0&0&0&\cdots&\cdots&\cdots&0\\1&-2&1&0&\cdots&\cdots&0\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\\cdots&0&1&-2&1&0&\cdots\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\0&\cdots&\cdots&0&1&-2&1\\0&\cdots&\cdots&\cdots&0&0&0\end{bmatrix}. \tag{15}$$

Note that $\mathbf{E}_x$ and $\mathbf{F}_x$ are functions of x since a,b,c,d are functions of x, and $\mathbf{C}^{-1}\mathbf{D}$ is independent of x and hence can be pre-calculated. Using this simplified notation, the parametric form (2) and (3) can be rewritten succinctly as

$$\mu_{i,l,d}\left(\zeta_{r,t,d}\right)=\mu_{i,l,d}^{(0)}+\nu_{\varpi(i,l,d)}^T\left(\zeta_{r,t,d}\right)\tilde{\mu}_{\varpi(i,l,d)}\text{, and} \tag{16}$$

$$\Sigma_{i,l,d}\left(\zeta_{r,t,d}\right)=\Sigma_{i,l,d}^{(0)}\left(\varsigma_{\varpi(i,l,d)}^T\left(\zeta_{r,t,d}\right)\tilde{\Sigma}_{\varpi(i,l,d)}\right)^{-2} \tag{17}$$

respectively where

$$\tilde{\mu}_{\varpi(i,l,d)}=\left[\begin{array}{ccc}\mu_{\varpi(i,l,d)}^{(1)} & \cdots & \mu_{\varpi(i,l,d)}^{(K)}\end{array}\right]^T, \tag{18}$$

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

$$\tilde{\Sigma}_{\varpi(i,l,d)}=\left[\begin{array}{ccc}\Sigma_{\varpi(i,l,d)}^{(1)} & \cdots & \Sigma_{\varpi(i,l,d)}^{(K)}\end{array}\right]^T\text{, and} \tag{20}$$

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

One of the key decisions to make in the CS-VPHMM is the choice of the environment dependent conditioning parameter $\zeta_{r,t,d}$. In our model and system $\zeta_{r,t,d}$ is chosen to be the dimension-wise instantaneous posterior SNR in the cepstral domain:

$$\zeta_{r,t,d}=\sum_i a_{d,i}\log\frac{\sigma_{i,y}^2}{\sigma_{i,n}^2}=\sum_i a_{d,i}\left(\log\sigma_{i,y}^2-\log\sigma_{i,n}^2\right), \tag{22}$$

where $a_{d,i}$ is the inverse discrete cosine transformation (IDCT) coefficient, $\sigma_{i,y}^2$ and $\sigma_{i,n}^2$ are the power of noisy signal and noise from the i-th Mel-frequency filter, respectively. A minimum-controlled recursive moving-average noise tracker [1] was used in our system to track the noise power $\sigma_{i,n}^2$ with the same procedure and parameters used in our MFCC-MMSE noise suppresser work reported in [5].

Note that we have set

$$\zeta_d^{(1)}=\mu_{\zeta_d}-\alpha\sigma_{\zeta_d}\text{, and} \tag{23}$$

$$\zeta_d^{(K)}=\mu_{\zeta_d}+\alpha\sigma_{\zeta_d} \tag{24}$$

as the conditioning value of the first and the last knots, where α was set to 2 in our experiments and we have assumed that each dimension of the conditioning parameter follows a Gaussian distribution whose mean μ_{ζ_d} and standard deviation σ_{ζ_d} can be estimated from the training data. Since $\zeta_d^{(1)}$ and $\zeta_d^{(K)}$ are independent on the Gaussian components, they can be shared across all Gaussian components.

3. Clustering of Cubic Splines

Our parametric formulation (1), (2) and (3) (or equivalently (1), (16) and (17)) allows for sharing the spline parameters across different Gaussian components. In this section we describe the spline clustering algorithm used in our CS-VPHMM.

The distance between two functions f_1 and f_2 given the

distribution of the domain $p(x)$ can be defined as

$$d(f_1,f_2)=\int_x\left(f_1(x)-f_2(x)\right)^2 p(x)dx\,. \tag{25}$$

This distance is also valid for two splines determined by the evenly distributed knots

$$\left\{y_1^{(i)}\mid i=1,\cdots,K\right\}, \text{ and} \tag{26}$$

$$\left\{y_2^{(i)}\mid i=1,\cdots,K\right\}. \tag{27}$$

Note that the calculation of the exact distance using (25) can be time consuming. For this reason, we approximate the integration in (25) with quantized summations

$$d(f_1,f_2)=h\sum_{x=1}^{K}\left(y_1^{(i)}-y_2^{(i)}\right)^2 p\left(x^{(i)}\right). \tag{28}$$

Since we have assumed that $p(x)=N(x;\mu,\sigma^2)$ follows the Gaussian distribution determined by the mean μ and the variance σ^2, (28) can be rewritten as

$$d(f_1,f_2)=h\sum_{x=1}^{K}\left(y_1^{(i)}-y_2^{(i)}\right)^2\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{\left(x^{(i)}-\mu\right)^2}{2\sigma^2}\right). \tag{29}$$

Note that the parameters h, μ and σ are the same for all the splines to be clustered, and

$$h=\frac{2\alpha\sigma}{K-1}, \tag{30}$$

$$x^{(i)}=x^{(1)}+(i-1)h, \tag{31}$$

$$\mu=x^{(1)}+\frac{K-1}{2}h\,. \tag{32}$$

The distance calculation can thus be further simplified as

$$\begin{aligned} d(f_1,f_2) &\propto \sum_{x=1}^{K}\left(y_1^{(i)}-y_2^{(i)}\right)^2\exp\left(-\frac{\left(x^{(i)}-\mu\right)^2}{2\sigma^2}\right)\\ &=\sum_{x=1}^{K}\left(y_1^{(i)}-y_2^{(i)}\right)^2\exp\left(-\frac{h^2\left(i-1-\frac{K-1}{2}\right)^2}{2\sigma^2}\right)\\ &=\sum_{x=1}^{K}\left(y_1^{(i)}-y_2^{(i)}\right)^2\exp\left(-\frac{2\alpha^2\left(i-\frac{K+1}{2}\right)^2}{(K-1)^2}\right). \end{aligned} \tag{33}$$

Note that our essential goal is to minimize the distances between the conditioning-parameter-dependent means and variances before and after the spline sharing. For this reason, when applying (33) to the variance splines, we replace $y_1^{(i)}$ and $y_2^{(i)}$ with $\log y_1^{(i)}$ and $\log y_2^{(i)}$ respectively.

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

4. Experiments

We have evaluated our parameter clustering and sharing algorithm for CS-VPHMM on the Aurora-3 corpus. We aim at finding out whether the gain can be attained when some splines are shared across different Gaussian components. In this section, we describe the experimental setting and results.

4.1. Experimental Setup

The Aurora-3 noisy digit recognition task under realistic automobile environments contains recordings from either a high, low, or quiet noise environment, and with either a close-talk microphone or a hands-free, far-field microphone. It is consisted of digit recognition sub-tasks for languages German, Finnish, Spanish, and Danish. For each language, three experimental settings are defined for the evaluation: the w*ell-matched* condition is the multi-training scenario where both the training and the testing sets contain all combinations of noise environments and microphones. In the *mid-mismatched* condition, the mismatch is mainly caused by the noise as 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, both channel distortion and additive noise exist as the training set contains close-talk data from all noise classes, and the testing set contains high noise and low noise far-field data.

The conventional 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, with 546 total number of Gaussian components. The 39-dimensional features used in our experiments are formed with the 13-dimension (with energy and without C0) static MFCC features and their first and second derivatives. The MFCC-MMSE motivated additive noise suppressor [5] has been applied to enhance the speech signal. The same noise tracking component is shared by both the noise suppressor and the conditioning parameter estimator.

To set the baseline we have trained a conventional HMM system using the ML criterion, on top of which a conventional HMM system was 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. The MCE baseline was trained using 10 percent of the training data as the held out set with detailed information available in the companion paper [6].

All the CS-VPHMMs reported in this paper were discriminatively trained (also using the MCE criterion) upon the MCE-trained conventional HMM with the number of knots in the cubic spline set to four. Due to the time complexity, we only ran four iterations of training and we report the result after the fourth iteration.

4.2. Experimental Results

Table 1 summarizes the number of spline clusters and the associated number of parameters used in different settings relative to the conventional HMM. The 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 only by one Gaussian component, the Gaussian component-specific mean and variance can be absorbed into the spline and that's the reason only 4 times of the parameters used in the conventional HMM are needed in the Setting 8 although each spline has 4 knots.

Table 2 summarizes the absolute WER on the Aurora-3 corpus and Figure 1 illustrates how the WER changes as a function of the number of spline clusters. In these experiments, we first trained the CS-VPHMM model for the Setting 8. We then determine the regression classes using the clustering algorithm described in Section 3 with the number of spline clusters predetermined according to Table 1. The

CS-VPHMM model with the specified number of spline clusters is then trained on top of the MCE trained conventional HMM.

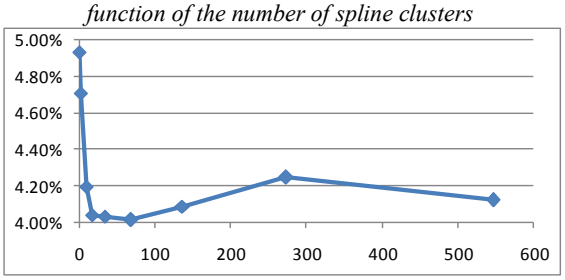
	# of Spline Clusters	# of Parameters (times)
Conventional HMM (MCE)	0	**1.00**
CS-VPHMM (MCE) Setting 1	1	**1.01**
CS-VPHMM (MCE) Setting 2	9	**1.06**
CS-VPHMM (MCE) Setting 3	17	**1.13**
CS-VPHMM (MCE) Setting 4	34	**1.25**
CS-VPHMM (MCE) Setting 5	68	**1.50**
CS-VPHMM (MCE) Setting 6	136	**2.00**
CS-VPHMM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

Table 1. *Summary of the number of spline clusters and the number of parameters relative to that used in the conventional HMM for different settings.*

Summary of Aurora 3 Absolute Word Error Rate				
	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%**
CS-VPHMM (MCE) Setting 1	4.71%	11.58%	22.94%	**11.67%**
CS-VPHMM (MCE) Setting 2	4.20%	11.07%	22.52%	**11.18%**
CS-VPHMM (MCE) Setting 3	4.04%	11.13%	22.79%	**11.21%**
CS-VPHMM (MCE) Setting 4	4.03%	11.12%	22.30%	**11.08%**
CS-VPHMM (MCE) Setting 5	4.01%	11.04%	22.57%	**11.11%**
CS-VPHMM (MCE) Setting 6	4.09%	10.99%	22.74%	**11.17%**
CS-VPHMM (MCE) Setting 7	4.25%	10.94%	22.57%	**11.17%**
CS-VPHMM (MCE) Setting 8	4.12%	11.27%	22.31%	**11.17%**

Table 2. *Summary of the absolute WER on Aurora-3 corpus*

Figure 1. *Absolute WER under well-matched condition as a function of the number of spline clusters*



From these tables and figures, we can see that the CS-VPHMM outperforms the MCE-trained conventional HMM under all conditions with the largest gain observed under the well-matched condition.

The curve in the Figure 1 demonstrated some important relationship between the number of parameters and the recognition accuracy. When no spline is shared (Setting 8) the CS-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 9, the WER increases again 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. This behavior is likely caused by the fact that two factors are affecting the final result when the number of clusters is decreased: the modeling ability becomes poorer since means and variances that share the same spline need to follow the same changing pattern; 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 factor 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 CS-VPHMM outperforms the MCE trained conventional HMM with 18.09% relative WER reduction even if only 17 spline clusters are used, or equivalently, 1.13 times of parameters used in the conventional HMM. This improvement is statistically significant at the significance level of 1%.

5. Conclusions

In the companion paper [6] we have shown that the CS-VPHMM can use the dimension-wise instantaneous SNR as the conditioning parameter, and the model parameters can be discriminatively trained using the growth-transformation based formula [3] [4] without using quantized conventional HMMs as the initial model and/or using quantization based approximation approach for parameter estimation. We have also shown that the CS-VPHMM significantly outperforms the conventional HMM at the cost of greatly increased number of parameters.

In this paper, we addressed the problem of increased number of parameters and presented a spline clustering and sharing algorithm. We demonstrated the effectiveness and the behavior pattern of the parameter clustering algorithm on Aurora-3 and showed that we can attain the gain even with 1.13 times of the parameters used in the conventional HMM. This is a great indication that CS-VPHMM may have great practical implication.

6. 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.

7. 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., Wu, 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. "Discriminative training of variable-parameter HMMs for noise robust speech recognition", Interspeech 2008 (to appear).