Model-Space MLLR for Trajectory HMMs

Heiga Zen, Yoshihiko Nankaku, Keiichi Tokuda

Department of Computer Science and Engineering
Nagoya Institute of Technology, Nagoya, Japan

{zen,nankaku,tokuda}@sp.nitech.ac.jp

Abstract

This paper proposes model-space Maximum Likelihood Linear Regression (mMLLR) based speaker adaptation technique for trajectory HMMs, which have been derived from HMMs by imposing explicit relationships between static and dynamic features. This model can alleviate two limitations of the HMM: constant statistics within a state and conditional independence assumption of state output probabilities without increasing the number of model parameters. Results in a continuous speech recognition experiments show that the proposed algorithm can adapt trajectory HMMs to a specific speaker and improve the performance of a trajectory HMM-based speech recognition system.

Index Terms: trajectory HMM, speaker adaptation, model-space MLLR

1. Introduction

For decades, hidden Markov models (HMMs) have emerged as the dominant acoustic models for both automatic speech recognition (ASR) and text-to-speech (TTS) synthesis due to their ease of implementation and modeling flexibility. The HMM performs well considering various assumptions it makes, such as piecewise constant statistics within states, frame-wise conditional independence assumption of state output probabilities, and simple geometric state duration probability density functions (PDF) [1, 2]. None of these assumptions seem to be held for real speech. To overcome these shortcomings of the HMM, a variety of alternative models have been proposed, e.g., [3–5]. Most of them have attempted to capture explicit dynamics of speech parameter trajectories. Although these models can improve model accuracy and speech recognition performance, they generally require an increase in the number of model parameters and computational complexity. Alternatively, the use of dynamic features (e.g., delta and delta-delta cepstral coefficients) [6] can also enhance the performance of HMM-based speech recognizers and has been widely adopted. It can be considered as a simple mechanism to capture time dependencies. Nevertheless, it has been thought of as an ad hoc rather than an essential solution. Generally, dynamic features are calculated as regression coefficients from static features of their neighboring frames. Thus, relationships between the static and dynamic features are deterministic. However, these relationships are ignored and the static and dynamic features are modeled as independent statistical variables. Ignoring these interdependencies allows inconsistency between the static and dynamic features when the HMM is used as a generative model in the obvious way.

Recently, a trajectory model, derived from the HMM by imposing the explicit relationships between static and dynamic features, has been proposed [7]. This model, named *trajectory HMM*, can overcome the conditional independence assumption of state output probabilities and piecewise constant statistics of the HMM without any additional parameters. Maximum likelihood (ML) training algorithms for the trajectory HMM based on the Viterbi and Monte Carlo approximations have also been derived [7, 8]. It was successfully applied to speaker-dependent acoustic modeling in both speech recognition and synthesis [7].

Most of state-of-the-art speech recognition systems use speaker adaptation techniques. These techniques aim to adapt speaker-independent acoustic models to a specific speaker to improve the speech recognition performance. In addition, these techniques are also used in the HMM-based speech synthesis framework [9] to construct a speaker-specific synthesis system using only a small amount of speech [10]. Speaker-adaptation techniques can roughly be clustered into three classes [11]; Maximum A Posteriori (MAP) adaptation [12], Maximum Likelihood Linear Regression (MLLR) [13], or Eigenvoice [14].

In this paper, we derive and evaluate model-space MLLR (mMLLR) [15] for the trajectory HMMs. Although the trajectory HMM has the same parameterization as the HMM, the definition of its output probability is different from that of the HMM. Accordingly, the adaptation algorithm should be re-derived based on its output probability.

The rest of this paper is organized as follows: Section 2 reviews the definition of the trajectory HMM. In Section 3, mMLLR-based adaptation algorithm for the trajectory HMM is derived. Results in a continuous speech recognition experiment are presented in Section 4. Concluding remarks and future plans are presented in the final section.

2. Definition of Trajectory HMMs

The output probability density function at a sequence of static feature vectors $\boldsymbol{c} = \left[\boldsymbol{c}_1^\top, \ldots, \boldsymbol{c}_T^\top\right]^\top$ for a trajectory

HMM Λ is given by

$$p(\boldsymbol{c} \mid \Lambda) = \sum_{\forall \boldsymbol{q}} p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda)\ P(\boldsymbol{q} \mid \Lambda), \tag{1}$$

$$p(\boldsymbol{c} \mid \boldsymbol{q}, \Lambda) = \mathcal{N}\left(\boldsymbol{c} \mid \bar{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right), \tag{2}$$

$$P(\boldsymbol{q} \mid \Lambda) = P(q_1 \mid \Lambda) \prod_{t=2}^{T} P(q_t \mid q_{t-1}, \Lambda), \tag{3}$$

where $\boldsymbol{c}_t$ is an M-dimensional acoustic static feature vector at time t (e.g., MFCC, PLP, etc.), $\boldsymbol{q} = \{q_1, q_2, \ldots, q_T\}$ is a state sequence,[1] q_t is the state at time t, and T is the number of frames in $\boldsymbol{c}$. In Eq. (2), $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ and $\boldsymbol{P}_{\boldsymbol{q}}$ are the $MT \times 1$ mean vector (smooth trajectory) and the $MT \times MT$ temporal covariance matrix for $\boldsymbol{q}$, respectively. They are given by

$$\boldsymbol{R}_{\boldsymbol{q}} \bar{\boldsymbol{c}}_{\boldsymbol{q}} = \boldsymbol{r}_{\boldsymbol{q}}, \tag{4}$$

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{W} = \boldsymbol{P}_{\boldsymbol{q}}^{-1}, \tag{5}$$

$$\boldsymbol{r}_{\boldsymbol{q}} = \boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{q}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}}, \tag{6}$$

$$\boldsymbol{\mu}_{\boldsymbol{q}} = \left[\boldsymbol{\mu}_{q_1}^\top, \ldots, \boldsymbol{\mu}_{q_T}^\top\right]^\top, \tag{7}$$

$$\boldsymbol{\mu}_i = \left[\boldsymbol{\mu}_i^\top, \Delta\boldsymbol{\mu}_i^\top, \Delta^2\boldsymbol{\mu}_i^\top\right]^\top, \quad i = 1, \ldots, N \tag{8}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{q}} = \mathrm{diag}\left[\boldsymbol{\Sigma}_{q_1}, \ldots, \boldsymbol{\Sigma}_{q_T}\right], \tag{9}$$

$$\boldsymbol{\Sigma}_i = \mathrm{diag}\left[\boldsymbol{\Sigma}_i, \Delta\boldsymbol{\Sigma}_i, \Delta^2\boldsymbol{\Sigma}_i\right], \quad i = 1, \ldots, N \tag{10}$$

where N is the total number of state output PDFs, and $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the $3M \times 1$ mean vector and the $3M \times 3M$ covariance matrix associated with the i-th state, respectively. In Eqs. (5) and (6), $\boldsymbol{W}$ is a $3MT \times MT$ window matrix whose elements are given as regression window coefficients to calculate delta and delta-delta features as follows:

$$\Delta\boldsymbol{c}_t = \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau)\boldsymbol{c}_{t+\tau}, \quad \Delta^2\boldsymbol{c}_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau)\boldsymbol{c}_{t+\tau}, \tag{11}$$

$$\boldsymbol{W} = [\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_T]^\top \otimes \boldsymbol{I}_{M\times M}, \tag{12}$$

$$\boldsymbol{W}_t = \left[\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}\right], \tag{13}$$

$$\boldsymbol{w}_t^{(d)} = \Big[\underbrace{0, \ldots, 0}_{t-L_-^{(d)}-1}, w^{(d)}(-L_-^{(d)}), \ldots, w^{(d)}(0), \ldots, w^{(d)}(L_+^{(d)}), \underbrace{0, \ldots, 0}_{T-\left(t+L_+^{(d)}\right)}\Big]^\top, d = 0, 1, 2 \tag{14}$$

where $L_-^{(0)} = L_+^{(0)} = 0$, $w^{(0)}(0) = 1$, and $\otimes$ denotes the Kronecker product for matrices.

[1] For notation simplicity, we assume that each state has a Gaussian distribution with a diagonal covariance matrix for its state output PDF.

Note that $\boldsymbol{c}$ is modeled by a mixture of Gaussian distributions whose dimensionality is MT, and the covariance matrices $\boldsymbol{P}_{\boldsymbol{q}}$ are generally full. As a result, the trajectory HMM can overcome the deficiencies of the HMM. It is also noted that the parameterization of the trajectory HMM is completely the same as that of the HMM with the same model topology.

3. Adaptation of Trajectory HMMs

Most of state-of-the-art speech recognition systems use speaker adaptation techniques. These techniques aim to adjust speaker-independent acoustic models to a specific speaker to improve the speech recognition performance. In addition, these techniques have also been applied to the HMM-based speech synthesis framework to obtain HMM-based synthesis systems with desired speaker's voice characteristics or speaking styles using small amount of speech data [10].

One of the most popular speaker adaptation techniques is Maximum Likelihood Linear Regression (MLLR) [13]. There are two forms in the MLLR framework: model-space and feature-space MLLR [15]. Feature-space MLLR for trajectory HMMs has been derived by Zen et al. [16]. In this section, we derive the model-space MLLR (mMLLR) for the trajectory HMM.

In mMLLR, mean vectors $\{\boldsymbol{\mu}_i\}$ and diagonal covariance matrices $\{\boldsymbol{\Sigma}_i\}$ of state output PDFs are transformed as

$$\hat{\boldsymbol{\mu}}_i = \boldsymbol{A}_{r(i)}\,\boldsymbol{\mu}_i + \boldsymbol{b}_{r(i)} = \boldsymbol{X}_{r(i)}\,\boldsymbol{\xi}_i, \tag{15}$$

$$\hat{\boldsymbol{\Sigma}}_i^{-1} = \boldsymbol{H}_{r(i)}^\top \boldsymbol{\Sigma}_i^{-1} \boldsymbol{H}_{r(i)}, \tag{16}$$

where $\boldsymbol{A}_j$, $\boldsymbol{b}_j$, and $\boldsymbol{H}_j$ are the $3M \times 3M$ mean transformation matrix, $3M \times 1$ bias vector, and $3M \times 3M$ variance transformation matrix, associated with the j-th regression class, respectively, and $\boldsymbol{X}_j = \left[\boldsymbol{b}_j\ \ \boldsymbol{A}_j\right]$ is the extended transform, $\boldsymbol{\xi}_i = \left[1\ \ \boldsymbol{\mu}_i^\top\right]^\top$ is the extended mean vector, $r(i)$ is the regression class to which the i-th state belongs ($1 \le r(i) \le R$), and R is the number of regression classes. The goal of mMLLR is to find $\{\boldsymbol{X}_j\}$ and $\{\boldsymbol{H}_j\}$ that maximize the model likelihood for given adaptation data $\boldsymbol{c}$. As in mMLLR for the HMM, the expectation-maximization (EM) algorithm may be used. The auxiliary function of the EM algorithm is defined as

$$Q\left(\{\boldsymbol{X}_j, \boldsymbol{H}_j\}, \{\boldsymbol{X}'_j, \boldsymbol{H}'_j\}\right) = K - \frac{1}{2}\sum_{\forall \boldsymbol{q}} \gamma_{\boldsymbol{q}} \cdot \left[-\log\left|\hat{\boldsymbol{R}}_{\boldsymbol{q}}\right| + \boldsymbol{c}^\top \hat{\boldsymbol{R}}_{\boldsymbol{q}} \boldsymbol{c} - 2\hat{\boldsymbol{r}}_{\boldsymbol{q}}^\top \boldsymbol{c} + \hat{\boldsymbol{r}}_{\boldsymbol{q}}^\top \hat{\boldsymbol{P}}_{\boldsymbol{q}} \hat{\boldsymbol{r}}_{\boldsymbol{q}}\right], \tag{17}$$

where K is a constant independent of $\{\boldsymbol{X}_j\}$ and $\{\boldsymbol{H}_j\}$, and

$$\gamma_{\boldsymbol{q}} = p\left(\boldsymbol{q} \mid \boldsymbol{c}, \{\boldsymbol{X}'_j, \boldsymbol{H}'_j\}, \Lambda\right), \tag{18}$$

$$\hat{\boldsymbol{R}}_{\boldsymbol{q}} \hat{\boldsymbol{c}}_{\boldsymbol{q}} = \hat{\boldsymbol{r}}_{\boldsymbol{q}}, \tag{19}$$

$$\hat{\boldsymbol{R}}_{\boldsymbol{q}} = \boldsymbol{W}^\top \hat{\boldsymbol{\Sigma}}_{\boldsymbol{q}}^{-1} \boldsymbol{W} = \hat{\boldsymbol{P}}_{\boldsymbol{q}}^{-1}, \quad \hat{\boldsymbol{r}}_{\boldsymbol{q}} = \boldsymbol{W}^\top \hat{\boldsymbol{\Sigma}}_{\boldsymbol{q}}^{-1} \hat{\boldsymbol{\mu}}_{\boldsymbol{q}}, \tag{20}$$

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{q}} = \left[\hat{\boldsymbol{\mu}}_{q_1}^\top, \ldots, \hat{\boldsymbol{\mu}}_{q_T}^\top\right]^\top, \quad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{q}} = \mathrm{diag}\left[\hat{\boldsymbol{\Sigma}}_{q_1}, \ldots, \hat{\boldsymbol{\Sigma}}_{q_T}\right]. \tag{21}$$

For exact computation of Eq. (17), all possible state sequences should be evaluated. However, it is intractable because the temporal covariance matrix $\hat{\boldsymbol{P}}_{\boldsymbol{q}}$ is generally full. Therefore, approximations such as Viterbi approximation [7] or Markov Chain Monte Carlo [8] should be introduced.

By setting the first partial derivative of Eq. (17) with respect to $\{\mathrm{vec}(\boldsymbol{X}_j)\}$ equal to $\boldsymbol{0}$, we obtain a set of linear equations to determine mean transformations as

$$\begin{bmatrix} \boldsymbol{G}_{11} & \cdots & \boldsymbol{G}_{1R} \\ \vdots & \ddots & \vdots \\ \boldsymbol{G}_{R1} & \cdots & \boldsymbol{G}_{RR} \end{bmatrix} \begin{bmatrix} \mathrm{vec}(\boldsymbol{X}_1) \\ \vdots \\ \mathrm{vec}(\boldsymbol{X}_R) \end{bmatrix} = \begin{bmatrix} \boldsymbol{k}_1 \\ \vdots \\ \boldsymbol{k}_R \end{bmatrix}, \tag{22}$$

$$\boldsymbol{G}_{ij} = \sum_{\forall \boldsymbol{q}} \sum_{t_1=1}^{T} \sum_{t_2=1}^{T} \gamma_{\boldsymbol{q}} \cdot \delta\left(r(q_{t_1}) = i\right) \cdot \delta\left(r(q_{t_2}) = j\right) \cdot \left(\boldsymbol{D}_{\boldsymbol{q}}^{(t_1,t_2)} \otimes \boldsymbol{V}_{\boldsymbol{q}}^{(t_1,t_2)}\right), \tag{23}$$

$$\boldsymbol{k}_i = \sum_{\forall \boldsymbol{q}} \sum_{t_1=1}^{T} \gamma_{\boldsymbol{q}} \cdot \delta\left(r(q_{t_1}) = i\right) \cdot \mathrm{vec}\left(\boldsymbol{Z}_{\boldsymbol{q}}^{(t_1,t_1)}\right), \tag{24}$$

where[2]

$$\delta\left(r(q_t) = i\right) = \begin{cases} 1 & r(q_t) = i \\ 0 & \text{otherwise} \end{cases}, \tag{25}$$

$$\boldsymbol{Z}_{\boldsymbol{q}}^{(t_1,t_2)} = \hat{\boldsymbol{\Sigma}}_{q_{t_1}}^{-1} \boldsymbol{o}_{t_1} \boldsymbol{\xi}_{q_{t_2}}^{\top}, \tag{26}$$

$$\boldsymbol{D}_{\boldsymbol{q}}^{(t_1,t_2)} = \boldsymbol{\xi}_{q_{t_1}} \boldsymbol{\xi}_{q_{t_2}}^{\top}, \tag{27}$$

$$\boldsymbol{V}_{\boldsymbol{q}}^{(t_1,t_2)} = \hat{\boldsymbol{\Sigma}}_{q_{t_1}}^{-1} \cdot \left[\boldsymbol{W} \hat{\boldsymbol{P}}_{\boldsymbol{q}} \boldsymbol{W}^{\top}\right]_{(t_1,t_2)}^{(3M,3M)} \cdot \hat{\boldsymbol{\Sigma}}_{q_{t_2}}^{-1}, \tag{28}$$

$$\boldsymbol{o}_{t_1} = \left[\boldsymbol{c}_{t_1}^{\top}, \Delta \boldsymbol{c}_{t_1}^{\top}, \Delta^2 \boldsymbol{c}_{t_1}^{\top}\right]^{\top} \tag{29}$$

The dimensionality of Eq. (22) becomes $3M(3M+1)R \times 3M(3M+1)R$. Although it could be thousands, we can solve it using currently available computational resources. If the number of adaptation data is small, $\{\boldsymbol{G}_{ij}\}$ cannot have full rank. We can avoid this problem using singular value decomposition (SVD).

To maximize Eq. (17) with respect to $\{\boldsymbol{H}_j\}$, a gradient ascent method such as steepest ascent or limited-memory BFGS method [17] is applied using its first partial derivative

$$\frac{\partial Q\left(\{\boldsymbol{X}_j, \boldsymbol{H}_j\}, \{\boldsymbol{X}'_j, \boldsymbol{H}'_j\}\right)}{\partial \boldsymbol{H}_i} = \sum_{\forall \boldsymbol{q}} \sum_{t_1=1}^{T} \gamma_{\boldsymbol{q}} \cdot \delta\left(r(q_{t_1}) = i\right) \cdot \boldsymbol{\Sigma}_i^{-1} \boldsymbol{H}_i \left[2\boldsymbol{W}\left(\boldsymbol{c} - \hat{\boldsymbol{c}}_{\boldsymbol{q}}\right)\hat{\boldsymbol{\mu}}_{\boldsymbol{q}}^{\top} + \boldsymbol{W}\left(\hat{\boldsymbol{P}}_{\boldsymbol{q}} - \boldsymbol{c}\boldsymbol{c}^{\top} + \hat{\boldsymbol{c}}_{\boldsymbol{q}}\hat{\boldsymbol{c}}_{\boldsymbol{q}}^{\top}\right)\boldsymbol{W}^{\top}\right]_{(t_1,t_1)}^{(3M,3M)}, \tag{30}$$

because Eq. (30) is not a quadratic function of $\boldsymbol{H}_i$. Because $\{\boldsymbol{H}_j\}$ are optimized by a gradient ascent method, setting their initial values properly is very important to achieve good convergence. If we assume $\boldsymbol{H}_1 = \cdots = \boldsymbol{H}_R = \sigma^{-1}\boldsymbol{I}$ and set the first partial derivative of Eq. (17) with respect to σ equal to 0, the analytical solution of σ that maximizes the auxiliary function is obtained as

$$\sigma^2 = \frac{1}{T} \sum_{\forall \boldsymbol{q}} \gamma_{\boldsymbol{q}} \cdot \left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right)^{\top} \boldsymbol{R}_{\boldsymbol{q}} \left(\boldsymbol{c} - \bar{\boldsymbol{c}}_{\boldsymbol{q}}\right). \tag{31}$$

We can use this $\sigma^{-1}\boldsymbol{I}$ as the initial values for optimization.

Note that all $\{\boldsymbol{X}_j\}$ and $\{\boldsymbol{H}_j\}$ should be estimated simultaneously because they are dependent one another via the temporal covariance matrix $\hat{\boldsymbol{P}}_{\boldsymbol{q}}$.

4. Experiments

4.1. Experimental conditions

Phonetically balanced 440 of 503 sentences uttered by male speakers MHO, MMY, MSH, MTK, and MYI (2200 sentences in total) from the ATR Japanese speech database B-set were used to train context-independent HMMs and trajectory HMMs. The remaining 10 and 53 sentences uttered by a male speaker MHT were used for adaptation and evaluation, respectively. These test utterances had an average length of 43 phonemes and an average duration of 4 seconds.

Speech signals were sampled at 16 kHz and windowed by a 25-ms Blackman window with a 10-ms shift, and then mel-cepstral coefficients were obtained by a mel-cepstral analysis technique. Static feature vectors consisted of 19 mel-cepstral coefficients including the zeroth coefficient. They were augmented by appending their first and second order dynamic features.

A three-state, left-to-right, no skip structure was used to model 36 Japanese phonemes including silence and short pause. Each state had a single Gaussian distribution with a diagonal covariance matrix. After training the HMMs in the standard way using HTK, the trajectory HMMs were iteratively re-estimated (two iterations) by the Viterbi training using the HMMs as their initial models. The number of delay for the delayed decision Viterbi algorithm [7] was 4 (beam width was 1500).

In this experiment, we used full structure for $\{\boldsymbol{A}_j\}$ and diagonal structure for $\{\boldsymbol{H}_j\}$. Here, we optimized $\{\log(\boldsymbol{H}_j)\}$ rather than $\{\boldsymbol{H}_j\}$ to keep $\hat{\boldsymbol{P}}_{\boldsymbol{q}}$ positive definite via the limited memory BFGS method. They were estimated in a supervised adaptation mode. To approximate Eq. (17), the Viterbi approximation was used. The number of regression classes R was 1 (global) or 2 (silence including short pause and others).

4.2. Experimental results

In the recognition experiment reported in this section, the re-scoring paradigm was used. Five sets of 500-best lists were generated by the HTK Viterbi decoder using the HMMs without adaptation, with mean adaptation ($R = 1$ or 2), or with mean and variance adaptation ($R = 1$ or 2). Each set of 500-best lists generated

[2] In this paper, we define $[\boldsymbol{A}]_{(k,l)}^{(m,n)}$ as

$$[\boldsymbol{A}]_{(m,n)}^{(k,l)} = \begin{bmatrix} \boldsymbol{A}_{11} & \cdots & \boldsymbol{A}_{1N} \\ \boldsymbol{A}_{21} & \cdots & \boldsymbol{A}_{2N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{A}_{T1} & \cdots & \boldsymbol{A}_{MN} \end{bmatrix}_{(m,n)} = \boldsymbol{A}_{mn},$$

where $\forall_{i,j} \boldsymbol{A}_{ij} \in \mathbb{R}^{k \times l}$.

Table 1: *Phoneme Error Rates (PER) of the HMMs and the trajectory HMMs with and without speaker adaptation (500+500 best lists re-scoring).*

Model	Adaptation	#Class (R)	PER (%)
HMM	w/o	–	45.7
	mean	1	36.8
		2	37.0
	mean +variance	1	27.5
		2	**27.3**
trajectory HMM	w/o	–	45.9
	mean	1	37.3
		2	36.8
	mean + variance	1	27.4
		2	**27.3**

using the adapted HMMs was merged with that of 500-best lists generated using the HMMs without adaptation. They were re-segmented and re-scored by the trajectory HMMs with and without adaptation.

Table 1 shows the phoneme error rates of the HMMs and trajectory HMMs with and without adaptation. It can be seen from the table that the mean and mean+variance adapted trajectory HMMs achieved about 20% and 40% relative error reductions over the trajectory HMM without adaption. However, there was no significant difference between the performance of the HMMs and trajectory HMMs. Although variance adaptation usually does not give large improvements, high gains were observed in this experiment. We think that the small number of training speakers caused this phenomena.

In the speaker-dependent experiments using the same data reported in [7], significant improvements over the HMMs have been reported. Further investigations about the difference between speaker-dependent model and speaker-independent model with adaptation are necessary.

5. Conclusion

In the present paper, a speaker adaptation technique for the trajectory HMM based on model-space MLLR was derived and evaluated. Although the mean and mean+variance adapted trajectory HMMs achieved 20% and 40% relative error reduction over the trajectory HMMs without adaptation, there was no significant difference between the performance of the HMMs and trajectory HMMs.

Future plan includes applying it to the HMM-based speech synthesis. Large-scale evaluation and further investigations are also necessary.

6. Acknowledgments

This work was partly supported by the MEXT e-Society project, the Hori information science promotion foundation, and the Grant-in-Aid for Scientific Research (No. 1880009) of Japan society for the promotion of science (JSPS).

7. References

[1] M. Ostendorf, V. Digalakis, and O.A. Kimball, "From HMMs to segment models," *IEEE Trans. Speech & Audio Processing*, vol. 4, no. 5, pp. 360–378, 1996.

[2] X.-D. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: a guide to theory, algorithm and system development*, Prentice Hall, 2001.

[3] M.J.F. Gales and S.J. Young, "Segmental hidden Markov models," in *Proc. Eurospeech*, 1993, pp. 1579–1582.

[4] A.V.I. Rosti and M.J.F Gales, "Switching linear dynamical systems for speech recognition," Tech. Rep. CUED/F-INFENG/TR.461, Cambridge University, 2003.

[5] K.K. Paliwal, "Use of temporal correlation between successive frames in hidden Markov model based speech recognizer," in *Proc. ICASSP*, 1993, pp. 215–218.

[6] S. Furui, "Speaker independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoustics, Speech, & Signal Processing*, vol. 34, pp. 52–59, 1986.

[7] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic features," *Computer Speech & Language*, vol. 21, no. 1, pp. 153–173, 2007.

[8] H. Zen, K. Tokuda, and T. Kitamura, "Estimating trajectory HMM parameters by Monte Carlo EM with Gibbs sampler," in *Proc. ICASSP*, 2006, pp. 1173–1176.

[9] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. Eurospeech*, 1999, pp. 2347–2350.

[10] J. Yamagishi, *Average-Voice-Based Speech Synthesis*, Ph.D. thesis, Tokyo Institute of Technology, 2006.

[11] P.C. Woodland, "Speaker adaptation for continuous density HMMs: A review," in *ITRW on adaptation methods for speech recognition*, 2001, pp. 11–19.

[12] J.L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[13] C.J. Leggetter and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, vol. 9, pp. 171–185, 1995.

[14] R. Kuhn, J.C. Janqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 6, pp. 695–707, 2000.

[15] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[16] H. Zen, Y. Nankaku, K. Tokuda, and T. Kitamura, "Speaker adaptation of trajectory HMMs using feature-space MLLR," in *Proc. Interspeech*, 2006, pp. 2274–2277.

[17] D.C. Liu and J. Nocedal, "On the limited memory method for large scale optimization," *Mathematical Programming B*, vol. 45, no. 3, pp. 503–528, 1989.