Trajectory Modeling based on HMMs with the Explicit Relationship between Static and Dynamic Features

Keiichi Tokuda, Heiga Zen, Tadashi Kitamura

Department of Computer Science
Nagoya Institute of Technology, Japan
{tokuda,zen,kitamura}@ics.nitech.ac.jp

Abstract

This paper shows that the HMM whose state output vector includes static and dynamic feature parameters can be reformulated as a trajectory model by imposing the explicit relationship between the static and dynamic features. The derived model, named trajectory HMM, can alleviate the limitations of HMMs: i) constant statistics within an HMM state and ii) independence assumption of state output probabilities. We also derive a Viterbi-type training algorithm for the trajectory HMM. A preliminary speech recognition experiment based on N-best rescoring demonstrates that the training algorithm can improve the recognition performance significantly even though the trajectory HMM has the same parameterization as the standard HMM.

1. Introduction

The HMM has successfully been applied to modeling the sequence of speech spectra in speech recognition systems. Its tractable and efficient implementations are achieved by assumptions: i) constant statistics within an HMM state and ii) independence assumption of state output probabilities. To overcome these limitations in the standard HMM framework, alternative models have been proposed, e.g., [1]–[10]. Although these models can improve the speech recognition performance, they generally require an increase in model parameters and computational complexity. The use of the dynamic features (delta and delta-delta features) [11] also improves the performance of HMM-based speech recognizers. However, it has been thought of as an ad hoc rather than an essential solution.

This paper derives a trajectory model by reformulating the standard HMM whose state output vector includes static and dynamic feature parameters. The standard HMM allows inconsistent static and dynamic features. By imposing the explicit relationship between static features and dynamic features, the standard HMM is naturally translated into a trajectory model, referred to as "trajectory HMM" in this paper. We also derive a Viterbi-type training algorithm for the trajectory model, and evaluate it in a continuous speech recognition task.

The formulation of the trajectory HMM is closely related to a technique for parameter generation from HMM [12]–[14], in which the speech parameter sequence is determined so as to maximize its output probability for the HMM under the constraints between static and dynamic features. While we derived the speech parameter generation algorithm in order to construct HMM-based speech synthesizers [15] which can synthesize speech with various voice characteristics [16]–[18], the generation algorithm was also applied to speech recognition in [19]. The proposed training algorithm can be viewed as a training algorithm for the method in [19]. The model derived in [20] based on the maximum entropy framework is also closely related to the proposed trajectory model.

The rest of this paper is organized as follows. Section 2 defines the trajectory model. In Section 3 the training algorithm for the model is derived. Results of a continuous speech recognition experiment are shown in Section 4. Concluding remarks and future plans are presented in the final section.

2. Reformulating HMM as trajectory model

The output probability of a speech parameter vector sequence $\boldsymbol{o} = \left[\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_T^\top\right]^\top$ for the standard HMM is given by

$$P(\boldsymbol{o} \,|\, \lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{o} \,|\, \boldsymbol{q}, \lambda) P(\boldsymbol{q} \,|\, \lambda), \tag{1}$$

where $\boldsymbol{q} = \{q_1, q_2, \ldots, q_T\}$ is a state sequence. In most of speech recognition systems, the speech parameter vector $\boldsymbol{o}_t$ is assumed to consist of the static feature vector $\boldsymbol{c}_t = [c_t(1), c_t(2), \ldots, c_t(M)]^\top$ (e.g., cepstral coefficients), and dynamic feature vectors $\Delta\boldsymbol{c}_t$, $\Delta^2\boldsymbol{c}_t$ (e.g., delta and delta-delta cepstral coefficients), that is $\boldsymbol{o}_t = [\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top]^\top$. The dynamic features calculated by

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

$$\Delta^2\boldsymbol{c}_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau)\boldsymbol{c}_{t+\tau}. \tag{3}$$

correspond to the first and second time-derivative of the static feature $\boldsymbol{c}_t$, respectively. Conditions (2) and (3) can be arranged in a matrix form:

$$\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}, \tag{4}$$

where

$$\boldsymbol{c} = \left[\boldsymbol{c}_1^\top, \boldsymbol{c}_2^\top, \ldots, \boldsymbol{c}_T^\top\right]^\top \tag{5}$$

$$\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T]^\top \tag{6}$$

$$\boldsymbol{w}_t = \left[\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}\right] \tag{7}$$

$$\boldsymbol{w}_t^{(n)} = \big[\underbrace{\boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}}_{\text{1st}}, \underbrace{w^{(n)}(-L_-^{(n)})\boldsymbol{I}_{M\times M}}_{(t-L_-^{(n)})\text{-th}},$$
$$\ldots, \underbrace{w^{(n)}(0)\boldsymbol{I}_{M\times M}}_{t\text{-th}}, \ldots, \underbrace{w^{(n)}(L_+^{(n)})\boldsymbol{I}_{M\times M}}_{(t+L_+^{(n)})\text{-th}},$$

$$\underbrace{\mathbf{0}_{M\times M},\ldots,\mathbf{0}_{M\times M}}_{T\text{-th}}\big]^\top,\quad n=0,1,2 \tag{8}$$

$L_-^{(0)}=L_+^{(0)}=0$, and $w^{(0)}(0)=1$. The above model is improper in the sense of statistical modeling: it allows inconsistent static and dynamic feature vector sequences even though they are constrained by (4). The statistical model should be defined as a function of $\boldsymbol{c}$ because the original observation is $\boldsymbol{c}$ rather than the augmented variable $\boldsymbol{o}$.

When each state output probability distribution is assumed to be single Gaussian, $P(\boldsymbol{o}\,|\,\boldsymbol{q},\lambda)$ is given by

$$P(\boldsymbol{o}\,|\,\boldsymbol{q},\lambda)=\prod_{t=1}^{T}\mathcal{N}(\boldsymbol{o}_t\,|\,\boldsymbol{\mu}_{q_t},\boldsymbol{U}_{q_t})=\mathcal{N}(\boldsymbol{o}\,|\,\boldsymbol{\mu}_{\boldsymbol{q}},\boldsymbol{U}_{\boldsymbol{q}}), \tag{9}$$

where $\boldsymbol{\mu}_{q_t}$ and $\boldsymbol{U}_{q_t}$ are the $3M\times 1$ mean vector and the $3M\times 3M$ covariance matrix, respectively, associated with q_t-th state, and

$$\begin{aligned}\boldsymbol{\mu}_{\boldsymbol{q}} &= \left[\boldsymbol{\mu}_{q_1}^\top,\boldsymbol{\mu}_{q_2}^\top,...,\boldsymbol{\mu}_{q_T}^\top\right]^\top &(10)\\ \boldsymbol{U}_{\boldsymbol{q}} &= \mathrm{diag}\left[\boldsymbol{U}_{q_1},\boldsymbol{U}_{q_2},...,\boldsymbol{U}_{q_T}\right]. &(11)\end{aligned}$$

By substituting (4) for (9), we can rewrite (9) as follows:

$$\begin{aligned}P(\boldsymbol{W}\boldsymbol{c}\,|\,\boldsymbol{q},\lambda) &= \mathcal{N}(\boldsymbol{W}\boldsymbol{c}\,|\,\boldsymbol{\mu}_{\boldsymbol{q}},\boldsymbol{U}_{\boldsymbol{q}})\\ &= K_{\boldsymbol{q}}\cdot\mathcal{N}(\boldsymbol{c}\,|\,\overline{\boldsymbol{c}}_{\boldsymbol{q}},\boldsymbol{P}_{\boldsymbol{q}}),\end{aligned} \tag{12}$$

where $\overline{\boldsymbol{c}}_{\boldsymbol{q}}$ is given by

$$\boldsymbol{R}_{\boldsymbol{q}}\overline{\boldsymbol{c}}_{\boldsymbol{q}}=\boldsymbol{r}_{\boldsymbol{q}} \tag{13}$$

and

$$\begin{aligned}\boldsymbol{R}_{\boldsymbol{q}} &= \boldsymbol{W}^\top\boldsymbol{U}_{\boldsymbol{q}}^{-1}\boldsymbol{W} &(14)\\ \boldsymbol{r}_{\boldsymbol{q}} &= \boldsymbol{W}^\top\boldsymbol{U}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} &(15)\\ \boldsymbol{P}_{\boldsymbol{q}} &= \boldsymbol{R}_{\boldsymbol{q}}^{-1} &(16)\\ K_{\boldsymbol{q}} &= \frac{\sqrt{(2\pi)^{MT}|\boldsymbol{P}_{\boldsymbol{q}}|}}{\sqrt{(2\pi)^{3MT}|\boldsymbol{U}_{\boldsymbol{q}}|}}\\ &\quad\cdot\exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{q}}^\top\boldsymbol{U}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}}-\boldsymbol{r}_{\boldsymbol{q}}^\top\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}\right)\right\}. &(17)\end{aligned}$$

By omitting the normalization constant $K_{\boldsymbol{q}}$ in the above expression (12), a new statistical model can be defined:

$$P(\boldsymbol{c}\,|\,\lambda)=\sum_{\text{all }\boldsymbol{q}}P(\boldsymbol{c}\,|\,\boldsymbol{q},\lambda)P(\boldsymbol{q}\,|\,\lambda), \tag{18}$$

where

$$P(\boldsymbol{c}\,|\,\boldsymbol{q},\lambda)=\mathcal{N}(\boldsymbol{c}\,|\,\overline{\boldsymbol{c}}_{\boldsymbol{q}},\boldsymbol{P}_{\boldsymbol{q}}). \tag{19}$$

It is interesting to note that the mean $\overline{\boldsymbol{c}}_{\boldsymbol{q}}$ is exactly the same as the speech parameter trajectory obtained by the speech parameter generation technique (Case 1 in [14], see Appendix also), that is,

$$\overline{\boldsymbol{c}}_{\boldsymbol{q}}=\arg\max_{\boldsymbol{c}}P(\boldsymbol{o}\,|\,\boldsymbol{q},\lambda)=\arg\max_{\boldsymbol{c}}P(\boldsymbol{W}\boldsymbol{c}\,|\,\boldsymbol{q},\lambda). \tag{20}$$

This means that by assuming the parameter trajectory $\overline{\boldsymbol{c}}_{\boldsymbol{q}}$ as the mean for the spectral parameter vector sequence $\boldsymbol{c}$ corresponding to an utterance, the standard HMM can naturally be translated into a trajectory model: the state output probability of observing the static part of the output vector changes during a state, and is affected by statistics of neighboring states. Note that the spectral parameter vector sequence $\boldsymbol{c}$ is modeled by a mixture of Gaussians whose dimensionality is TM, and their covariances $\boldsymbol{P}_{\boldsymbol{q}}$ are generally full.

3. Training Algorithm

In this section, we derive a training algorithm for the trajectory model. It should be noted that although the model has the same parameterization as the standard HMM, the output probability is defined by (18) rather than by (1). Accordingly, the model parameters should be trained based on (18).

An auxiliary function of current parameter set λ and new parameter set λ' is defined by

$$\mathcal{Q}(\lambda,\lambda')=\sum_{\text{all }\boldsymbol{q}}P(\boldsymbol{q}\,|\,\boldsymbol{c},\lambda)\log P(\boldsymbol{c},\boldsymbol{q}\,|\,\lambda'). \tag{21}$$

It can be shown that by substituting λ' which maximizes $\mathcal{Q}(\lambda,\lambda')$ for λ, the likelihood increases unless λ is a critical point of the likelihood. However, it is not tractable since we have to evaluate all possible state sequences. To avoid this difficulty, we apply the single Viterbi path approximation[1]. As a result, the problem is broken down into the following maximization problems:

$$\begin{aligned}\boldsymbol{q}_{\max} &= \arg\max_{\boldsymbol{q}}P(\boldsymbol{c},\boldsymbol{q}\,|\,\lambda) &(22)\\ \lambda' &= \arg\max_{\lambda}P(\boldsymbol{c},\boldsymbol{q}_{\max}\,|\,\lambda) &(23)\end{aligned}$$

3.1. Model parameter update

First, we solve the maximization problem of (23). The problem is equivalent to maximizing

$$\begin{aligned}\log P(\boldsymbol{c}\,|\,\boldsymbol{q},\lambda) &= -\frac{1}{2}\big[MT\log(2\pi)-\log|\boldsymbol{R}_{\boldsymbol{q}}|\\ &\quad+\boldsymbol{c}^\top\boldsymbol{P}_{\boldsymbol{q}}^{-1}\boldsymbol{c}+\boldsymbol{r}_{\boldsymbol{q}}^\top\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}-2\,\boldsymbol{r}_{\boldsymbol{q}}^\top\boldsymbol{c}\big]\end{aligned} \tag{24}$$

with respect to

$$\begin{aligned}\boldsymbol{m} &= \left[\boldsymbol{\mu}_1^\top,\boldsymbol{\mu}_2^\top,\ldots,\boldsymbol{\mu}_N^\top\right]^\top &(25)\\ \boldsymbol{\sigma} &= \left[\boldsymbol{U}_1^{-1},\boldsymbol{U}_2^{-1},\ldots,\boldsymbol{U}_N^{-1}\right]^\top, &(26)\end{aligned}$$

where N is the total number of HMM states. In this paper, we refer to $P(\boldsymbol{c}\,|\,\boldsymbol{q},\lambda)$ as "trajectory likelihood."

By setting $\partial\log P(\boldsymbol{c}\,|\,\boldsymbol{q},\lambda)/\partial\boldsymbol{m}=\mathbf{0}$, we obtain a set of linear equations

$$\boldsymbol{S}_{\boldsymbol{q}}^\top\boldsymbol{W}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{W}^\top\boldsymbol{S}_{\boldsymbol{q}}\boldsymbol{\Sigma}^{-1}\boldsymbol{m}=\boldsymbol{S}_{\boldsymbol{q}}^\top\boldsymbol{W}\boldsymbol{c} \tag{27}$$

for determination of $\boldsymbol{m}$ which maximizes $\log P(\boldsymbol{c}\,|\,\boldsymbol{q},\lambda)$, where

$$\begin{aligned}\boldsymbol{\Sigma}^{-1} &= \mathrm{diag}\,(\boldsymbol{\sigma}) &(28)\\ \boldsymbol{\mu}_{\boldsymbol{q}} &= \boldsymbol{S}_{\boldsymbol{q}}\boldsymbol{m} &(29)\\ \boldsymbol{U}_{\boldsymbol{q}}^{-1} &= \mathrm{diag}\,(\boldsymbol{S}_{\boldsymbol{q}}\boldsymbol{\sigma}), &(30)\end{aligned}$$

and $\boldsymbol{S}_{\boldsymbol{q}}$ is a $3T\times 3MN$ matrix whose elements are 0 or 1 determined according to the state sequence $\boldsymbol{q}$. The dimensionality of (27) is $3MN$: although it could be tens of thousands, it is still small enough to solve the set of linear equations using currently available computational resources.

For maximizing $\log P(\boldsymbol{c}\,|\,\boldsymbol{q},\lambda)$ with respect to $\boldsymbol{\sigma}$, we apply a steepest descent algorithm using the first derivative

$$\begin{aligned}\frac{\partial\log P(\boldsymbol{c}\,|\,\boldsymbol{q},\lambda)}{\partial\boldsymbol{\sigma}} &= \frac{1}{2}\boldsymbol{S}_{\boldsymbol{q}}^\top\mathrm{diag}^{-1}\big(\boldsymbol{W}\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{W}^\top\\ &\quad-\boldsymbol{W}\boldsymbol{c}\boldsymbol{c}^\top\boldsymbol{W}^\top+2\boldsymbol{\mu}_{\boldsymbol{q}}\boldsymbol{c}^\top\boldsymbol{W}^\top\\ &\quad+\boldsymbol{W}\overline{\boldsymbol{c}}_{\boldsymbol{q}}\overline{\boldsymbol{c}}_{\boldsymbol{q}}^\top\boldsymbol{W}^\top-2\boldsymbol{\mu}_{\boldsymbol{q}}\overline{\boldsymbol{c}}_{\boldsymbol{q}}^\top\boldsymbol{W}^\top\big)\end{aligned} \tag{31}$$

because (24) is not a quadratic function of $\boldsymbol{\sigma}$.

[1] We can also use the N-best approximation.

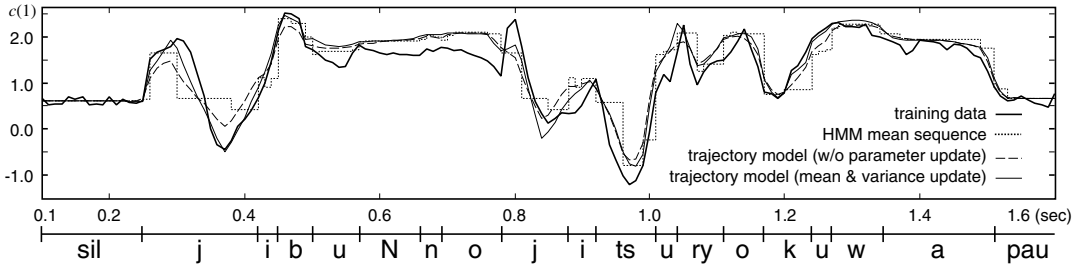

Figure 1: *Generated trajectories $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$ for the trajectory HMM (thin line) and the standard HMM (broken line), the state mean sequence $\boldsymbol{\mu}_{\boldsymbol{q}}$ (dotted line), and one of training data $\mathbf{c}$ (thick line).*

Table 1: *Average log "trajectory likelihood" $P(\boldsymbol{c} \mid \boldsymbol{q}, \lambda)$.*

	trajectory HMM			
	w/o update	$\boldsymbol{m}$	$\boldsymbol{\sigma}$	$\boldsymbol{m}$ & $\boldsymbol{\sigma}$
training data	6.86	7.91	12.0	14.8
test data	6.93	7.83	12.1	14.8

Table 2: *Recognition error rates (%) for the trajectory HMM and the standard HMM.*

	trajectory HMM				standard
	w/o update	$\boldsymbol{m}$	$\boldsymbol{\sigma}$	$\boldsymbol{m}$ & $\boldsymbol{\sigma}$	HMM
error rate	19.4	18.5	19.0	18.1	19.5
rel. imp.	0.5%	5%	2.6%	7%	ref

3.2. Optimal path search

Next, we discuss the optimal path search problem of (22). Based on the approximation

$$\begin{aligned} \boldsymbol{q}_{\max} &= \arg\max_{\boldsymbol{q}} P(\boldsymbol{q}, \boldsymbol{c} \mid \lambda) \\ &= \arg\max_{\boldsymbol{q}} \frac{1}{K_{\boldsymbol{q}}} P(\boldsymbol{q}, \boldsymbol{o} \mid \lambda) \qquad (32) \\ &\simeq \arg\max_{\boldsymbol{q}} P(\boldsymbol{q}, \boldsymbol{o} \mid \lambda), \qquad (33) \end{aligned}$$

we can use the Viterbi algorithm for the standard HMM. To reduce the inaccuracy caused by the approximation, the obtained state boundaries can be adjusted by a recursive algorithm (case 2 in [14]) based on the "trajectory likelihood." By incorporating time-recursive calculation of the normalization factor $K_{\boldsymbol{q}}$ in the Viterbi algorithm, we can obtain an optimal path directly by a modified Viterbi algorithm. This would be possible by using the speech parameter generation algorithm of [21] which can operates in a time-recursive manner with some look-ahead.

Noted that extensions of the training algorithm to the multiple observations and multi-mixture distributions are straightforward.

3.3. Decoding

For decoding, we can use i) rescoring scheme, ii) recursive calculation of normalization factor $K_{\boldsymbol{q}}$ in the Viterbi algorithm. In approach i), N-best candidates are rescored by the "trajectory likelihood." The state boundaries of each candidate can also be adjusted by the recursive algorithm (case 2 in [14]). On the other hand, in approach ii), $K_{\boldsymbol{q}}$ is calculated in the Viterbi decoding using the time-recursive algorithm of [21] with some look-ahead.

4. Experiment

A preliminary experiment was carried out. We used phonetically balanced 450 sentences from ATR Japanese speech database for training speaker-dependent monophone HMMs and their trajectory versions. Diagonal covariances were used for both types of models. Speech signals were sampled at 16 kHz and windowed by a 25.6-ms Blackman window with a 10-ms shift, and then mel-cepstral coefficients were obtained by a mel-cepstral analysis technique [22]. The feature vector consists of 19 mel-cepstral coefficients including the zeroth coefficient as well as their delta and delta-delta coefficients. We used 3-state left-to-right HMM structure with no skips. For a given Viterbi path obtained by the standard HMM, parameters of the trajectory HMM were optimized by the algorithm described in Section 3.1 (i.e., the proposed training algorithm was not iterated in this experiment). The trained trajectory HMM was evaluated in a continuous phoneme recognition task based on the N-best rescoring approach: first, a 10-best list was generated for each test utterance by using the conventional Viterbi algorithm with the standard HMM, and then each candidate was rescored by the trajectory HMM.

Table 1 shows average log likelihoods of the trajectory HMMs ("w/o update": model parameters were not updated, i.e., those of the standard HMM were used, "$\boldsymbol{m}$": only means were updated, "$\boldsymbol{\sigma}$": only variances were updated, "$\boldsymbol{m}$ & $\boldsymbol{\sigma}$": both means and variances were updated). It is shown that the training algorithm improves the "trajectory likelihoods" considerably for both training and test data sets.

Figure 1 shows the mean trajectories (the first coefficient in $\bar{\boldsymbol{c}}_{\boldsymbol{q}}$) for the trajectory HMM ("w/o parameter update": model parameter of the standard HMM were used, "mean and variance update": means and variances were updated). The state mean sequence of the standard HMM (the first coefficient in static part of $\boldsymbol{\mu}_{\boldsymbol{q}}$) and the trajectory extracted from one of the training data (the first coefficient in $\boldsymbol{c}$) are also plotted in the figure. It can be seen that the trajectory generated from the trained trajectory HMM is closer to the trajectory of the training data. It should be noted that the trajectory corresponding to a phoneme varies according to its neighboring phonemes (see phoneme /j/ in the figure). This means that the trajectory HMM has the capability to capture the coarticulation effects naturally.

Table 2 shows the phoneme recognition error rates for the trajectory HMM and the standard HMM. The trajectory HMM achieves a relative error reduction of 7% over the standard HMM. Given the fact that without parameter update the perfor-

mance of the trajectory HMM did not improve, model parameter training based on the "trajectory likelihood" is essential in the trajectory HMM approach.

5. Conclusion

This paper defined a new kind of trajectory model (trajectory HMM) for acoustic modeling of speech by reformulating the standard HMM whose observation vector consists of static and dynamic features. We also derived a Viterbi-type training algorithm for the trajectory model. It was shown that the model can work as a trajectory model while it maintains the basic structure of the standard HMM. In a preliminary speech recognition experiment, a relative error rate reduction of 7% over the standard HMM was achieved. One of the advantages of the trajectory HMM is that huge amounts of software resources for the standard HMM could be easily reused.

In the near future, results of a complete speech recognition experiment and application to Text-to-Speech synthesis will be presented. Future work also includes derivation of a fast algorithm for solving (27) and a Baum-Welch-type training algorithm.

6. Acknowledgment

Authors would like to thank Prof. Takao Kobayashi and Dr. Takashi Masuko for discussions.

7. References

[1] M. J. F. Gales and S. J. Young, "Segmental Hidden Markov Models," Proc. EUROSPEECH, pp.1579–1582, 1993.

[2] H. Gish and K. Ng, "Parametric trajectory models for speech recognition," Proc. ICSLP, pp.I-466–I-469, 1996.

[3] W. J. Holmes and M. J. Russell, "Speech recognition using a linear dynamic segmental HMM," Proc. ICASSP, pp.1611–1614, 1995.

[4] M. Ostendorf, V. Digalakis, and O. A. Kimball, "From HMMs to segment models: a unified view of stochastic modeling for speech recognition," IEEE Trans. on Speech and Audio Processing, vol.4, no.5, pp.360–378, 1996.

[5] L. Deng, M. Aksmanovic, X. Sun, J. Wu, "Speech recgnition using hiden Markov models with polynomial regression functions as nonstationary states," IEEE Trans. on Speech and Audio Processing, vol.2, no.4, pp.507–520, Oct. 1994.

[6] Y. Gong and J. P. Haton, "Stochastic trajectory modeling for speech recognition," Proc. ICASSP, vol.I, pp57–60, 1994.

[7] S. Takahashi, "Phoneme HMM's constrained by frame correlations," Proc. ICASSP, pp. 219-222, 1993.

[8] K. K. Paliwal, "Use of temporal correlation between successive frames in hidden Markov model based Speech recognizer," Proc. ICASSP, pp.215-218, 1993.

[9] M. Ostendorf and S.Roukos, "A stochastic segment model for phoneme-based continuous speech recognition," IEEE Trans. On Acoustics, Speech and Signal Processing, vol.37, no.12, pp.1857–1869, 1989.

[10] C. J. Wellekens, "Explicit correlation in hidden Markov model for Speech Recognition," Proc. ICASSP, pp.383-386, 1987.

[11] S. Furui, "Speaker independent isolated word rcognition using dynamic features of speech spectrum," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-34, pp.52–59, 1986.

[12] K. Tokuda, T. Kobayashi and S. Imai, "Speech parameter generation from HMM using dynamic features," Proc. ICASSP, pp.660–663, 1995.

[13] K. Tokuda, T. Masuko, T. Yamada, T. Kobayashi and S. Imai, "An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features," Proc. EUROSPEECH, pp.757–760, 1995.

[14] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," Proc. ICASSP, vol.3, pp.1315–1318, June 2000.

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

[16] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi and T. Kitamura, "Speaker Interpolation in HMM-Based Speech Synthesis System," Proc. EUROSPEECH, pp.2523–2526, 1997.

[17] M. Tamura, T. Masuko, K. Tokuda and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," Proc. ICASSP, vol.2, pp.805–808, May 2001.

[18] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, T. Kitamura, "Eigenvoices for HMM-based speech synthesis," Proc. ICSLP, pp.1269–1272, 2002.

[19] Y. Minami, E. McDermott, A. Nakamura, S. Katagiri, "A recognition method with parametric trajectory synthesized using direct relations between static and dynamic feature vector time series," Proc. ICASSP, vol.I, pp.957–960, 2002.

[20] K. S. Van Horn, "A Maximum-entropy solution to the frame-dependency problem in speech recognition," Tech. Rep., Dept. of Computer Science, North Dakota State University, Nov. 2001.

[21] K. Koishida, K. Tokuda, T. Masuko and T. Kobayashi, "Vector quantization of speech spectral parameters using statistics of dynamic features," Proc. ICSP, pp.247–252, 1997.

[22] T. Fukada, K. Tokuda, T. Kobayashi and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. ICASSP*, 1992, pp.137–140.

[23] A. Acero, "Formant analysis and synthesis using hidden Markov models," Proc. EUROSPEECH, pp.1047–1050, 1999.

A. Appendix

A.1. Maximizing $P(\boldsymbol{o} \,|\, \boldsymbol{q}, \lambda)$ with respect to $\boldsymbol{o}$ under $\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$

The logarithm of $P(\boldsymbol{o} \,|\, \boldsymbol{q}, \lambda)$ can be written as

$$\log P(\boldsymbol{o} \,|\, \boldsymbol{q}, \lambda) = -\frac{1}{2}\boldsymbol{o}^\top \boldsymbol{U}_{\boldsymbol{q}}^{-1}\boldsymbol{o} + \boldsymbol{o}^\top \boldsymbol{U}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} + \text{Const}, \quad (34)$$

where Const is independent of $\boldsymbol{o}$.

It is obvious that $P(\boldsymbol{o} \,|\, \boldsymbol{q}, \lambda)$ is maximized when $\boldsymbol{o} = \boldsymbol{\mu}_{\boldsymbol{q}}$ without the condition (4), that is, the speech parameter vector sequence becomes a sequence of the mean vectors. With the condition (4), maximizing $P(\boldsymbol{o} \,|\, \boldsymbol{q}, \lambda)$ with respect to $\boldsymbol{o}$ is equivalent to that with respect to $\boldsymbol{c}$. By setting

$$\frac{\partial \log P(\boldsymbol{W}\boldsymbol{c} \,|\, \boldsymbol{q}, \lambda)}{\partial \boldsymbol{c}} = \boldsymbol{0}, \quad (35)$$

we obtain a set of equations

$$\boldsymbol{R}_{\boldsymbol{q}}\boldsymbol{c} = \boldsymbol{r}_{\boldsymbol{q}}. \quad (36)$$

For direct solution of (36), we need $O(T^3M^3)$ operations[2] because $\boldsymbol{R}_{\boldsymbol{q}}$ is a $TM \times TM$ matrix. By utilizing the special structure of $\boldsymbol{R}_{\boldsymbol{q}}$, (36) can be solved by the Cholesky decomposition or the QR decomposition with $O(TM^3L^2)$ operations[3], where $L = \max_{n\in\{1,2\}, s\in\{-,+\}} L_s^{(n)}$. Equation (36) can also be solved by an algorithm derived in [12]–[14], which can operate in a time-recursive manner [21].

[2]When $\boldsymbol{U}_{q,i}$ is diagonal, it is reduced to $O(T^3M)$ since each of the M-dimensions can be calculated independently.

[3]When $\boldsymbol{U}_{q,i}$ is diagonal, it is reduced to $O(TML^2)$. Furthermore, when $L_-^{(1)} = -1$, $L_+^{(1)} = 0$, and $w^{(2)}(i) \equiv 0$, it is reduced to $O(TM)$ as described in [23].