SPEECH PARAMETER GENERATION FROM HMM USING DYNAMIC FEATURES

Keiichi Tokuda†, *Takao Kobayashi*†† *and Satoshi Imai*††

†Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Tokyo, 152 Japan
††Precision and Intelligence Laboratory, Tokyo Institute of Technology, Yokohama, 227 Japan

ABSTRACT

This paper proposes an algorithm for speech parameter generation from HMMs which include the dynamic features. The performance of speech recognition based on HMMs has been improved by introducing the dynamic features of speech. Thus we surmise that, if there is a method for speech parameter generation from HMMs which include the dynamic features, it will be useful for speech synthesis by rule. It is shown that the parameter generation from HMMs using the dynamic features results in searching for the optimum state sequence and solving a set of linear equations for each possible state sequence. We derive a fast algorithm for the solution by the analogy of the RLS algorithm for adaptive filtering. We also show the effect of incorporating the dynamic features by an example of speech parameter generation.

1. INTRODUCTION

The hidden Markov models (HMMs) can model sequences of speech spectra by well-defined algorithms, and have successfully been applied to speech recognition systems. From these facts, we surmise that, if there is a method for speech parameter generation from HMMs, it will be useful for speech synthesis by rule. For example, it is feasible to synthesize speech with various voice quality by using speaker adaptation technique in HMM-based speech recognition, and synthesis units can be selected automatically based on the model clustering and splitting methods used in HMM-based speech recognition. In addition, it is expected that the method is applicable to speech enhancement, voice quality conversion, etc.

From this point of view, this paper proposes an algorithm for speech parameter generation from continuous HMMs which include the dynamic features [1]. It is shown that the parameter generation from HMMs using the dynamic features results in searching for the optimum state sequence and solving a set of linear equations for each possible state sequence. We derive a fast algorithm for the solution by the analogy of the RLS algorithm [2] for adaptive filtering.

We show the effect of incorporating the dynamic features by an example of speech parameter generation in the case where the speech parameter consists of mel-cepstrum and delta mel-cepstrum. The performance of speech recognition based on HMMs is improved by introducing the dynamic features of speech. Similarly, it is shown that the dynamic feature plays an important role in speech parameter generation; without the dynamic features the generated parameter sequence becomes a sequence of the mean vectors independently of contexts.

2. PROBLEM

Let $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T\}$ be the vector sequence of speech parameter and $\mathbf{q} = \{q_1, q_2, \ldots, q_T\}$ be the state sequence of an HMM λ. In this paper, we assume that the vector of speech parameter $\mathbf{o}_t$ at frame t consists of the static feature vector $\mathbf{c}_t$ (e.g., cepstral coefficients) and the dynamic feature vector $\Delta\mathbf{c}_t$ (e.g., delta cepstral coefficients), that is,

$$\mathbf{o}_t = \{\mathbf{c}_t, \Delta\mathbf{c}_t\} \tag{1}$$

where

$$\mathbf{c}_t = [c(1), c(2), \ldots, c(M)]' \tag{2}$$

$$\Delta\mathbf{c}_t = [\Delta c(1), \Delta c(2), \ldots, \Delta c(M)]' \tag{3}$$

and $\Delta\mathbf{c}_t$ is defined as

$$\Delta\mathbf{c}_t = \sum_{i=-L}^{L} w(i)\, \mathbf{c}_{t+i}. \tag{4}$$

To simplify the discussion, we assume that $\mathbf{c}$ and $\Delta\mathbf{c}_t$ are statistically independent.

The problem is to determine the parameter sequence $\mathbf{c} = [\mathbf{c}_1', \mathbf{c}_2', \ldots, \mathbf{c}_T']'$ which maximizes

$$P[\mathbf{O} \,|\, \lambda] = \sum_{\text{all } \mathbf{q}} P[\mathbf{q}, \mathbf{O} \,|\, \lambda] \tag{5}$$

0-7803-2431-5/95 $4.00 © 1995 IEEE

for a given HMM λ. However, since the problem is difficult to solve, we consider the optimum sequence in a similar manner of the Viterbi algorithm, that is, we maximize

$$\overline{P}[\mathbf{O} \mid \lambda] = \max_{\mathbf{q}} P[\mathbf{q}, \mathbf{O} \mid \lambda] \tag{6}$$

with respect to $\mathbf{c}$. Since we have to determine $\mathbf{q}$ and $\mathbf{c}$ simultaneously, in contrast to the Viterbi algorithm the dynamic programming methods cannot be used.

3. SOLUTION OF THE PROBLEM

To solve the problem, first we consider maximizing $P[\mathbf{q}, \mathbf{O} \mid \lambda]$ for a given state sequence $\mathbf{q}$ with respect to $\mathbf{c}$. The probability $P[\mathbf{q}, \mathbf{O} \mid \lambda]$ is written as

$$P[\mathbf{q}, \mathbf{O} \mid \lambda] = P[\mathbf{q} \mid \lambda] \cdot P[\mathbf{O} \mid \mathbf{q}, \lambda]. \tag{7}$$

For given $\mathbf{q}$, maximizing $P[\mathbf{q}, \mathbf{O} \mid \lambda]$ with respect to $\mathbf{c}$ is equivalent to maximizing $P[\mathbf{O} \mid \mathbf{q}, \lambda]$ with respect to $\mathbf{c}$ because the probability $P[\mathbf{q} \mid \lambda]$ does not depend on $\mathbf{O}$. The probability $P[\mathbf{O} \mid \mathbf{q}, \lambda]$ is written by

$$P[\mathbf{O} \mid \mathbf{q}, \lambda] = b_{q_1}(\mathbf{o}_1)\, b_{q_2}(\mathbf{o}_2) \cdots b_{q_T}(\mathbf{o}_T). \tag{8}$$

Without loss of generality we assume that the distributions of $\{b_{q_t}(\mathbf{o}_t)\}$ are single mixture Gaussian because mixture components can be considered to be a special form of sub-state in which the transition probabilities are the mixture weitghts. Therefore, the output probability at state j is given by

$$b_j(\mathbf{o}_t) = \mathcal{N}(\mathbf{c}_t;\ \boldsymbol{\mu}_j, \mathbf{U}_j) \cdot \mathcal{N}(\Delta\mathbf{c}_t;\ \Delta\boldsymbol{\mu}_j, \Delta\mathbf{U}_j) \tag{9}$$

where $\boldsymbol{\mu}_j$ and $\mathbf{U}_j$ are the M-by-1 mean vector and the M-by-M covariance matrix of $\mathbf{c}_t$ at state j, respectively, $\Delta\boldsymbol{\mu}_j$ and $\Delta\mathbf{U}_j$ are those of $\Delta\mathbf{c}_t$, respectively, and $\mathcal{N}(\ \cdot\ ;\ \boldsymbol{\mu}, \mathbf{U})$ denotes the Gaussian distribution. Thus, the logarithm of $P[\mathbf{O} \mid \mathbf{q}, \lambda]$ is written as

$$\begin{aligned}\log P[\mathbf{O} \mid \mathbf{q}, \lambda] = &-\frac{1}{2}(\mathbf{c} - \boldsymbol{\mu})'\mathbf{U}^{-1}(\mathbf{c} - \boldsymbol{\mu}) \\ &-\frac{1}{2}(\mathbf{W}\mathbf{c} - \Delta\boldsymbol{\mu})'\Delta\mathbf{U}^{-1}(\mathbf{W}\mathbf{c} - \Delta\boldsymbol{\mu}) \\ -\frac{1}{2}\sum_{t=1}^{T}\log|\mathbf{U}_{q_t}| &- \frac{1}{2}\sum_{t=1}^{T}\log|\Delta\mathbf{U}_{q_t}| - TM\log 2\pi \end{aligned} \tag{10}$$

where

$$\boldsymbol{\mu} = [\boldsymbol{\mu}'_{q_1}, \boldsymbol{\mu}'_{q_2}, \ldots, \boldsymbol{\mu}'_{q_T}]' \tag{11}$$

$$\mathbf{U} = \mathrm{diag}[\mathbf{U}_{q_1}, \mathbf{U}_{q_2}, \ldots, \mathbf{U}_{q_T}] \tag{12}$$

$$\Delta\boldsymbol{\mu} = [\Delta\boldsymbol{\mu}'_{q_1}, \Delta\boldsymbol{\mu}'_{q_2}, \ldots, \Delta\boldsymbol{\mu}'_{q_T}]' \tag{13}$$

$$\Delta\mathbf{U} = \mathrm{diag}[\Delta\mathbf{U}_{q_1}, \Delta\mathbf{U}_{q_2}, \ldots, \Delta\mathbf{U}_{q_T}] \tag{14}$$

$$\mathbf{W} = \begin{bmatrix} w(0)\mathbf{I}_M & \cdots & w(L)\mathbf{I}_M & & \mathbf{0} \\ \vdots & w(0)\mathbf{I}_M & & \ddots & \\ w(L)\mathbf{I}_M & & \ddots & & w(L)\mathbf{I}_M \\ & \ddots & & \ddots & \vdots \\ \mathbf{0} & & w(L)\mathbf{I}_M & \cdots & w(0)\mathbf{I}_M \end{bmatrix} \tag{15}$$

and $\mathbf{I}_M$ denotes the M-by-M identity matrix. We assume that

$$\mathbf{c}_t = \mathbf{0}_M,\ \ t = -L, \ldots, -2, -1, T{+}1, T{+}2, \ldots, T{+}L. \tag{16}$$

where $\mathbf{0}_M$ denotes the M-by-1 zero vector.

To maximize (10) with respect to $\mathbf{c}$, by setting $\partial \log P[\mathbf{O} \mid \mathbf{q}, \lambda]/\partial\mathbf{c} = \mathbf{0}_{TM}$, we obtain a set of equations

$$\mathbf{R}\mathbf{c} = \mathbf{r} \tag{17}$$

where

$$\mathbf{R} = \mathbf{U}^{-1} + \mathbf{W}'\Delta\mathbf{U}^{-1}\mathbf{W} \tag{18}$$

$$\mathbf{r} = \mathbf{U}^{-1}\boldsymbol{\mu} + \mathbf{W}'\Delta\mathbf{U}^{-1}\Delta\boldsymbol{\mu}. \tag{19}$$

For direct solution of (17), we need $O(T^3M^3)$ operations on the assumption that $M, L \ll T$. When $\mathbf{U}_t$ and $\Delta\mathbf{U}_t$ are diagonal, it becomes $O(T^3M)$. To obtain $\mathbf{q}$ and $\mathbf{c}$ which maiximize $P[\mathbf{q}, \mathbf{O} \mid \lambda]$, we have to solve (17) for every possible state sequence. Fortunately, by using special properties of (17), we can derive a fast algorithm for determination of $\mathbf{q}$ and $\mathbf{c}$ as follows.

Let us consider replacing the values of the mean vector and covariance matrix, $\{\Delta\boldsymbol{\mu}_{q_t}, \Delta\mathbf{U}_{q_t}\}$, at a frame t with $\{\Delta\boldsymbol{\mu}_{\hat{q}_t}, \Delta\mathbf{U}_{\hat{q}_t}\}$. The corresponding set of equations can be written as

$$\hat{\mathbf{R}}\hat{\mathbf{c}} = \hat{\mathbf{r}} \tag{20}$$

where

$$\hat{\mathbf{R}} = \mathbf{R} + \mathbf{w}\mathbf{D}\mathbf{w}' \tag{21}$$

$$\hat{\mathbf{r}} = \mathbf{r} + \mathbf{w}\mathbf{d} \tag{22}$$

$$\mathbf{D} = \Delta\mathbf{U}_{\hat{q}_t}^{-1} - \Delta\mathbf{U}_{q_t}^{-1} \tag{23}$$

$$\mathbf{d} = \Delta\mathbf{U}_{\hat{q}_t}^{-1}\Delta\boldsymbol{\mu}_{\hat{q}_t} - \Delta\mathbf{U}_{q_t}^{-1}\Delta\boldsymbol{\mu}_{q_t} \tag{24}$$

$$\mathbf{w} = [\underset{1}{0}, \ldots, 0, \underset{t-L}{w(-L)}, \ldots, \underset{t}{w(0)}, \ldots, \underset{t+L}{w(L)}, 0, \ldots, \underset{T}{0}]'. \tag{25}$$

It can be seen that the relation of (21), (22) is similar to the time update property of the set of equations for the RLS adaptive filtering [2]. Consequently, we can derive a fast algorithm which obtaines $\hat{\mathbf{c}}$ from $\mathbf{c}$ recursively by the analogy of the derivation of the standard RLS algorithm, i.e., the application of the matrix inversion

Table 1: Summary of the proposed algorithm.

- Set $\mathbf{D}$, $\mathbf{d}$, $\mathbf{w}$ by (23)–(25) to replace $\{\Delta\boldsymbol{\mu}_{q_t}, \Delta\mathbf{U}_{q_t}\}$ with $\{\Delta\boldsymbol{\mu}_{\hat{q}_t}, \Delta\mathbf{U}_{\hat{q}_t}\}$
- Set $\mathbf{D}$, $\mathbf{d}$, $\mathbf{w}$ by (26)–(28) to replace $\{\boldsymbol{\mu}_{q_t}, \mathbf{U}_{q_t}\}$ with $\{\boldsymbol{\mu}_{\hat{q}_t}, \mathbf{U}_{\hat{q}_t}\}$.

Substitue $\hat{\mathbf{c}}$ and $\hat{\mathbf{P}}$ obtained by the previous iteration to $\mathbf{c}$ and $\mathbf{P}$, respectively, and calculate

$$\begin{aligned}
\boldsymbol{\pi} &= \mathbf{w}'\mathbf{P} \\
\boldsymbol{\kappa} &= \mathbf{D}^{-1} + \boldsymbol{\pi}\mathbf{w} \\
\mathbf{k} &= \mathbf{P}\mathbf{w}\boldsymbol{\kappa}^{-1} \\
\hat{\mathbf{P}} &= \mathbf{P} - \mathbf{k}\boldsymbol{\pi} \\
\hat{\mathbf{c}} &= \mathbf{c} + \mathbf{k}\left(\mathbf{D}^{-1}\mathbf{d} - \mathbf{w}'\mathbf{c}\right)
\end{aligned}$$

lemma. To replace $\{\boldsymbol{\mu}_{q_t}, \mathbf{U}_{q_t}\}$ with $\{\boldsymbol{\mu}_{\hat{q}_t}, \mathbf{U}_{\hat{q}_t}\}$ we can use following equations instead of equations (23)–(25):

$$\mathbf{D} = \mathbf{U}_{\hat{q}_t}^{-1} - \mathbf{U}_{q_t}^{-1} \tag{26}$$

$$\mathbf{d} = \mathbf{U}_{\hat{q}_t}^{-1}\boldsymbol{\mu}_{\hat{q}_t} - \mathbf{U}_{q_t}^{-1}\boldsymbol{\mu}_{q_t} \tag{27}$$

$$\mathbf{w} = [\underset{1}{0}, \ldots, 0, \underset{t}{1}, 0, \ldots, \underset{T}{0}]'. \tag{28}$$

The algorithm is summarized in Table 1. It is noted that $\mathbf{P} = \mathbf{R}^{-1}$. The computational complexity of the algorithm becomes $O(T^2M^3)$. When $\mathbf{U}_t$ and $\Delta\mathbf{U}_t$ are diagonal, the computational complexity is reduced to $O(T^2M)$. By using the recursive algorithm, we can search for the optimum state sequence keeping $\mathbf{c}$ optimal in the sense that $P[\mathbf{O}\,|\,\mathbf{q}, \lambda]$ is maximized with respect to $\mathbf{c}$. For each state sequence, we can use the recursive algorithm instead of solving (17) directly. Therefore the total computational complexity is significantly reduced.

The overall procedure for parameter generation from HMMs is summarized as follows:

1. Solve the set of equations (17) for an initial state sequence, and obtain $\mathbf{c}$ and $\mathbf{P}$.
2. Replace the state q_t of a frame t with $\hat{q}_t$ according to a certain strategy, and obtain $\hat{\mathbf{c}}$ and $\hat{\mathbf{P}}$ by using the algorithm shown in Table 1.
3. If the value of $\log P[\hat{\mathbf{q}}, \hat{\mathbf{O}}\,|\,\lambda]$ is not smaller than that of $\log P[\mathbf{q}, \mathbf{O}\,|\,\lambda]$, discard the replacement.
4. Repeat 2 and 3 until a certain condition is satisfied.

For the initail state sequence, (17) can be solved as follows. On the assumption that $\Delta\boldsymbol{\mu}_{q_t} = \mathbf{0}_M$ and $\Delta\mathbf{U}_{q_t} = \mathbf{0}_{M\times M}$ for $t = 1, 2, \ldots, T$, the solution of (17) is given as $\mathbf{c} = \boldsymbol{\mu}$ and $\mathbf{P} = \mathbf{U}$. Next, by putting the values of $\Delta\boldsymbol{\mu}_{q_t}$ and $\Delta\mathbf{U}_{q_t}$ back with the original value for $t = 1, 2, \ldots, T$ using the algorithm T times, we can obtain $\mathbf{c}$ and $\mathbf{P}$ for the initail state sequence.

The initial state sequence should be given appropriately; a reasonable way is to select the initial state sequence which maximizes $P[\mathbf{q}\,|\,\lambda]$. For a given T, we can obtain such an initial state sequence by using the Viterbi algorithm. Although the above procedure does not search every possible state sequence, since the initial state sequence should be close to the optimum state sequence, we can obtain an optimal or sub-optimal solution without a large number of iteration of the proposed algorithm.

The parameter generation procedure discussed in this paper can be straightforwardly extended to the version for mixture HMMs. It is also noted that we can easily extend the proposed algorithm to include delta-delta parameters and allow multiple data streams.

4. SPEECH SYNTHESIS BASED ON HMM

We suppose that mel-cepstrum is used as speech parameter. However the LPC-derived mel-cepstrum is not proper for synthesizing speech because it does not represent the original spectrum obtained by the LPC analysis. We have already proposed a speech analysis method [3] and a speech synthesis method [4] in which speech spectrum is represented by mel-cepstrum consistently. This analysis/synthesis method is suitable for the method proposed in this paper.

To synthesize speech, we require pitch information besides spectral information. Consequently, speech parameters in the proposed algorithm should include pitch information of speech signal. Although pitch contour generation methods based on HMMs have been proposed [5], [6], they have not used the dynamic features to generate pitch contours.

To generate high quality speech we have to investigate the issues such as the choice of model size (number of states), choice of output distribution (number of mixtures, diagonal or full covariance matrix), and choice of units of model (phoneme or syllable, context-dependent or independent). These choices must be made from the viewpoint that differs from that of speech recognition.

In the conventional HMMs the probability of state occupancy decreases exponentially with time. This type of state duration probability does not provide an adequate representation of the temporal structure of speech. To control temporal structure appropriately, we should use HMMs with state duration models.

5. EXAMPLE

A simple experiment of speech parameter generation was carried out using the ATR 5240 Japanese word data base. Only data from one speaker (speaker MAU) was used. We used 3 different phoneme models a, i, o plus an additional silence model sil, i.e., 4 models in all. Assuming an unreal language which consists of 3 phonemes /a/, /i/, /o/, we used only segments whose contexts are one of /a/, /i/, /o/, /sil/ for the training. The type of HMM used was a continuous Gaussian single mixture model with no explicit duration modeling. All models were 3-state left to right models with no skips. All feature vectors comprised of 13 mel-cepstral coefficients and 13 delta mel-cepstral coefficients. Both mel-cepstral and delta mel-cepstral coefficients included 0th coefficients. Mel-cepstral coefficients were obtained by the 12th order mel-cepstral analysis. The signal was windowed by a 25.6ms Blackman window with a 5ms shift. The state durations d_j at state j were estimated from self-transition probability $a_{j,j}$ as $d_j = 1/(1 - a_{j,j})$, and the search for the optimum state sequence was not carried out in this example.

Fig. 1 shows the spectra calculated from the mel-cepstral coefficients generated by an HMM, which is composed by concatenation of phoneme models; sil, a, i, sil. Without the dynamic features, the parameter sequence which maximizes $P[\mathrm{q}, \mathrm{O} \mid \lambda]$ becomes a sequence of the mean vectors; $\boldsymbol{\mu}_{q_1}, \boldsymbol{\mu}_{q_2}, \ldots, \boldsymbol{\mu}_{q_T}$ (see Fig. 1(a)). On the other hand, Fig. 1(b) shows that an appropriate parameter sequence is generated by using the static and dynamic features. At the first and last states of each phoneme model, the variances of static and dynamic features are relatively large, the generated speech spectra are modified appropriately according to the context of each model. At middle states, since static and dynamic features have small variances and the mean vectors of dynamic feature are nearly equal to zero, the generated speech spectra have forms corresponding to the mean vectors of static feature. From the example, it is seen that incorporating the dynamic features is essential to generate speech parameters from HMMs, and the algorithm is expected to be useful for concatenation of synthesis units in speech synthesis by rule.

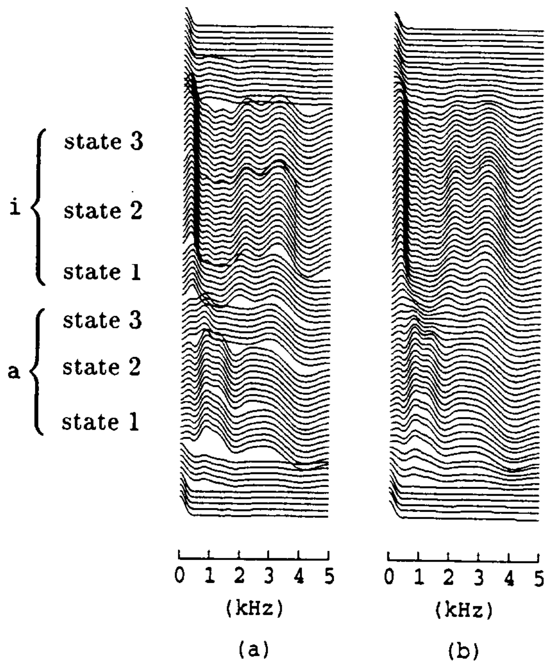

Figure 1: An example of parameter generation from an HMM composed by concatenation of phoneme models; sil, a, i, sil, (a) without dynamic feature, and (b) with dynamic feature.

6. CONCLUSION

We have proposed an algorithm for speech parameter generation from HMMs using the dynamic features. It is expected that the algorithm is useful for speech synthesis by rule, speech enhancement, voice conversion, etc. Implementation of a speech synthesis system is our future work. It can develop to a speech synthesis system by rule which can imitate various voice qualities and express emotions. Further reduction of computational complexity of the algorithm is also a future problem.

REFERENCES

[1] 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.

[2] S. Haykin, *Adaptive Filter Theory*, Englewood Cliffs, N.J.: Prentice-Hall, 1991.

[3] T. Fukada, K. Tokuda, T. Kobayashi and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. ICASSP-92*, 1992, pp.I-137–I-140.

[4] S. Imai, "Cepstral analysis synthesis on the mel frequency scale," in *Proc. ICASSP-83*, 1983, pp.93–96.

[5] T. Fukada, Y. Komori, T. Aso and Y. Ohara, "A study of pitch pattern generation using HMM-based statistical information," in *Proc. ICSLP-94*, 1994, pp.723–726.

[6] A. Ljolje and F. Fallside, "Synthesis of natural sounding pitch contous in isolated utterances using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp.1074–1080, 1986.