FIG. 4.6. *Plots of the first seven MFCC of the utterance.*

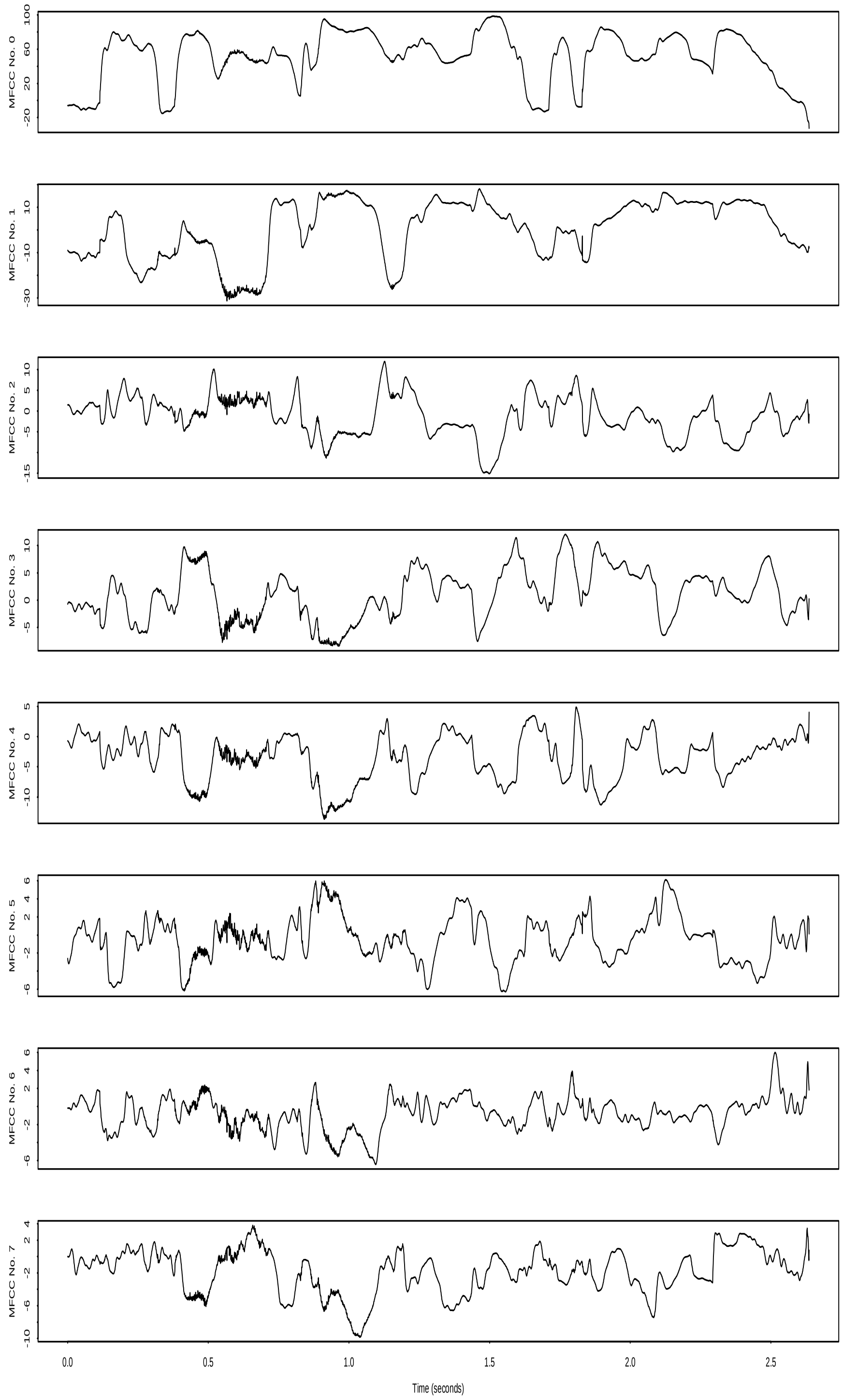


FIG. 4.5. *Spectrogram of utterance "sx361" from speaker "fsma0" with the transcription: "The speech symposium might begin Monday".*

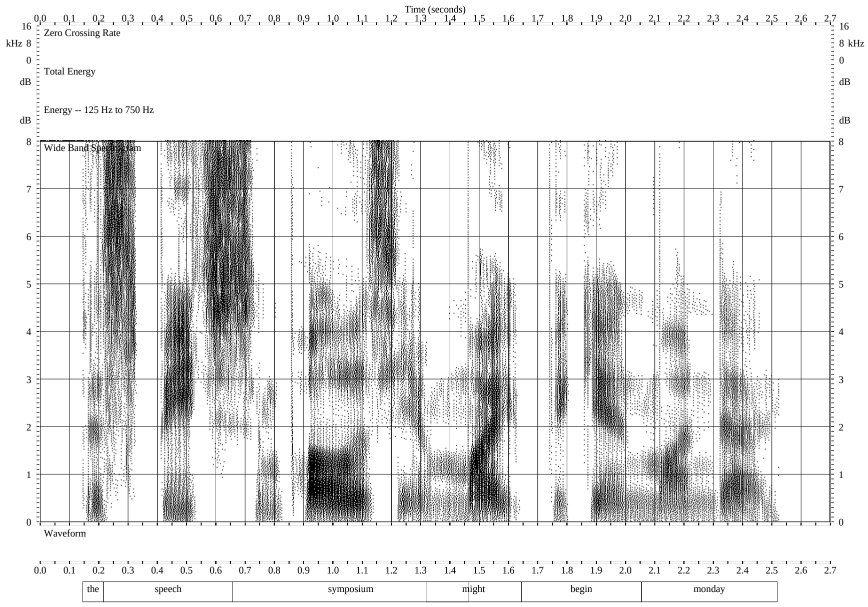


FIG. 4.4. *The same data fitting but using the non-stationary HMM with the state-dependent time warping mechanism incorporated.*

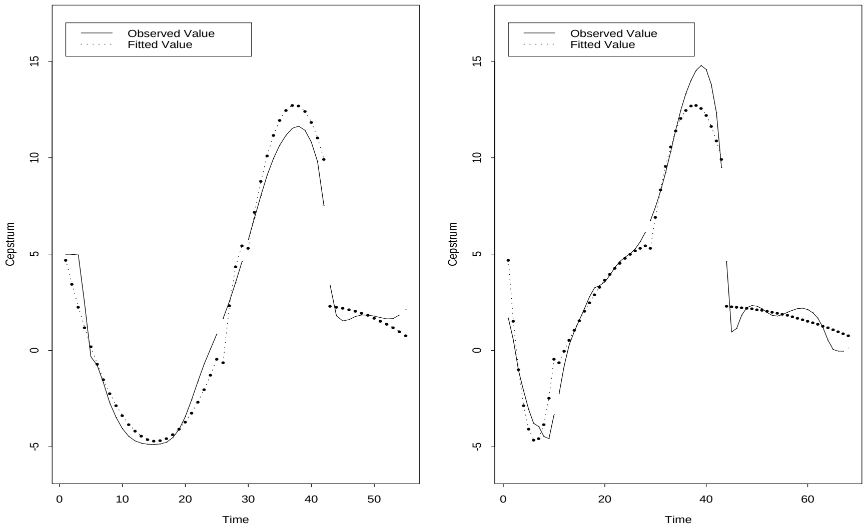


FIG. 4.2. *The same data fitting but using the non-stationary HMM with the state-dependent time warping mechanism incorporated.*

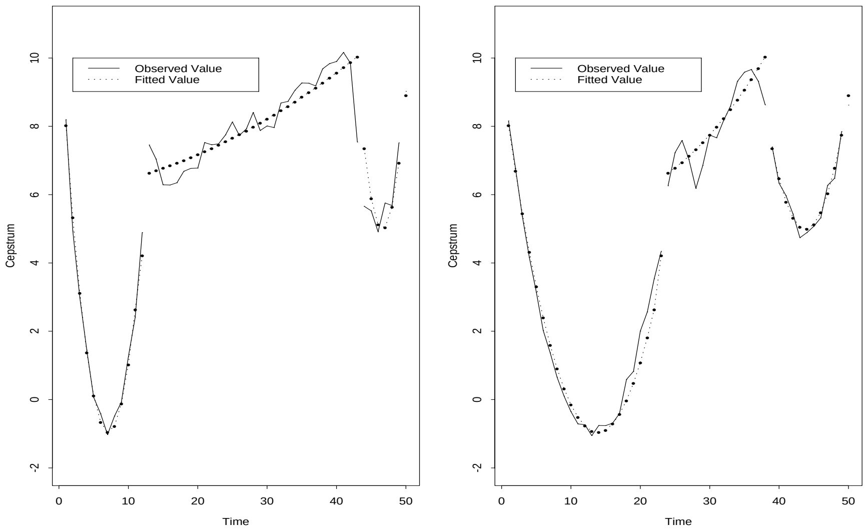


FIG. 4.3. *Fitting the non-stationary HMM with no time warping mechanism to C3 data sequences from two tokens of word bike excised from the TIMIT database.*

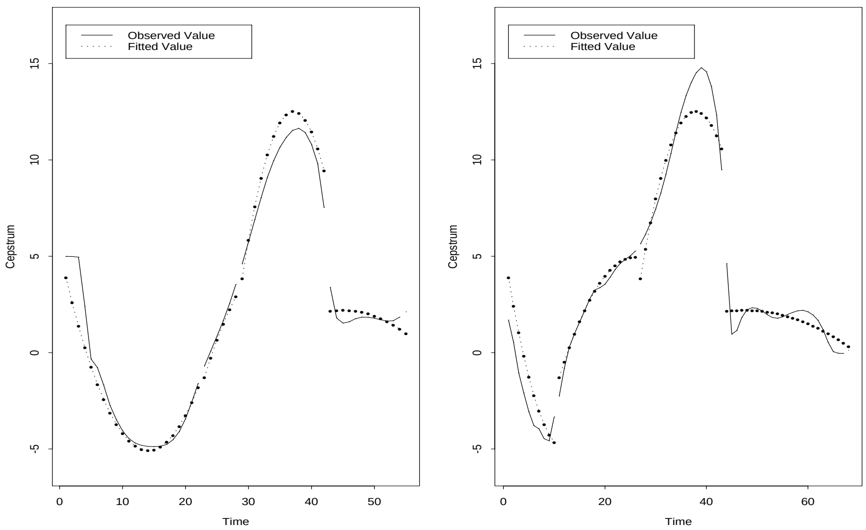
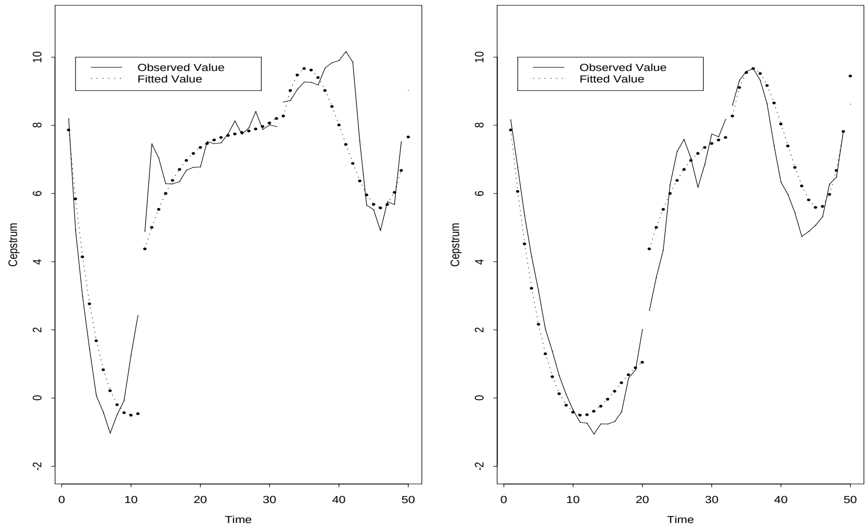


FIG. 4.1. *The solid lines in the two graphs are C1 data sequences from two tokens, respectively, uttered by the same speaker from the same word peek. Vertical axis is the magnitude of C1; horizontal axis is the frame number. The dotted lines superimposed on the graphs are three sequentially arranged state-dependent polynomial regression functions with the polynomial coefficients optimally trained from the two data tokens and with no time warping incorporated The point in time in each graph where the otherwise continuous regression line is broken is the frame at which optimal state transitions occur.*



To illustrate the data processing procedure, we provide an example of an utterance taken from TIMIT with the corresponding spectrogram plot and MFCC plots. We take the sentence "sx361" from speaker "fsma0" with the transcription: "The speech symposium might begin Monday" as the example here, and the plots are shown in Figure 4.5 and Figure 4.6.

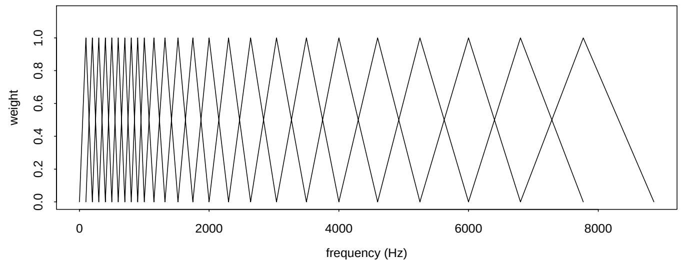
frequency triangular filter banks defined as

$$w_j(f) = \begin{cases} (f - f_{j-1})/(f_j - f_{j-1}), & \text{if} \quad f_{j-1} \leq f \leq f_j \\ (f_{j+1} - f)/(f_{j+1} - f_j), & \text{if} \quad f_j < f \leq f_{j+1} \\ 0 & else \end{cases}$$

for $j = 1, \cdots, J$, where J is the number of band-pass filters, and f_j are defined as follows.

$$f_j = \begin{cases} 100 * j & \text{if} \quad 0 \leq j \leq 10 \\ 1000 * (1.15)^{j-10} & \text{if} \quad j > 10 \end{cases}$$

For signal sampled at 16 kHz (i.e. f_{max} = 8000 Hz), we only need J=25 band-pass filters to cover the entire frequency range (0-8000 Hz). The mel-frequency filter banks are illustrated in the following graph.



To obtain the MFCC, we first define the log-channel energy vector by

$$X_j = log \left[\frac{\sum_{i=0}^{N} x_i \cdot w_j(\frac{i}{N} \cdot f_{max})}{\sum_{i=0}^{N} w_j(\frac{i}{N} \cdot f_{max})} \right], \qquad j = 1, \cdots, J.$$

Then, MFCC in computed as

$$Y_i = \sum_{j=1}^{J} X_j \;\cdot\; cos \left[i(j - \frac{1}{2})\frac{\pi}{J} \right], \quad i = 0, \cdots, M.$$

where M is called the order of the MFCC. From empirical study, the first few MFCC's capture most of the relevant information. The characteristics of the higher cepstrum coefficients largely depend on the speakers. In our experiments, only the first seven MFCC's are used.

The above description of the MFCC vectors is only for one FFT processing window, which moves along the time axis every 10 ms (1/100 seconds). Therefore, the resulting MFCC from one windowed speech waveform data can be viewed as a time-dependent features vector at a specific time point. Let $O_t, \quad t = 1, \cdots, T$ denote the time series of feature vector of speech sounds from time frame 1 to T.

[8] O. Ghitza and M. Sondhi. "Hidden Markov models with templates as non-stationary states: an application to speech recognition," *Computer Speech and Language*, Vol.7, No., 2, 1993, pages 101–119.
[9] B.H. Juang and L. R. Rabiner. "The segmental k-means algorithm for estimating parameters of hidden Markov models," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 38, 1990, pp.1639–1641.
[10] B.H. Juang and L. R. Rabiner. "Hidden Markov models for speech recognition," *Technometrics* , Vol. 33, No. 3, pp. 251–272, 1991.
[11] B.H. Juang and L. R. Rabiner. "Issues in using hidden Markov models for speech recognition." *Advances in Speech Signal Processing*, edited by S. Furui and M.M. Sondhi, Marcel Dekker, Inc., pp. 509–553, 1992.
[12] I. Lehiste. *Readings in Acoustic Phonetics*, MIT Press, Cambridge, MA., 1967.
[13] S.E. Levinson, "Continuously variable duration hidden Markov models for automatic speech recognition," *Computer, Sppech and language*, 1, 29–45, 1986.
[14] L.A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Information Theory*, Vol.28, pp. 729-734, 1982.
[15] K.V. Mardia, J.T. Kent, and J.M. Bibby, *Multivariate Analysis* , Academic Press, Harcourt Brace & Co., 1979.
[16] L.R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of the IEEE*, Vol. 77, February, 1989, pp. 257–285.
[17] D.X. Sun and L. Deng, "Nonstationary-State Hidden Markov Model with State-Dependent Time Warping: Application to Speech Recognition," *Proc. 1994 Int. Conf. on Spoken Language Processing*, September 18–22, 1994, Yokohama, Japan.
[18] V.W. Zue. "Speech Spectrogram Reading: An Acoustic Study of the English Language", Lecture Notes, M.I.T., Cambridge, MA., August, 1991.

Appendix A: Speech signal processing. In this appendix, we describe the basic speech signal processing methods for obtaining a time series of feature vectors, which is used for speech modeling, from a given speech waveform.

Speech signal is originally recorded as a sequence of acoustic waveform. It is usually sampled at a rate of 8–20 kHz (1 kHz = 1000 times/second) with 16-bit amplitude quantization (i.e., each waveform sample is represented by one of 2^{16}=65,536 values). Since most of the phonetic features of speech signal are based on the spectral properties, we convert the time domain signal into frequency domain for speech signal modeling. There are many spectral analysis methods including the standard discrete (fast) Fourier transform (FFT), linear prediction coding (LPC), and some auditory models that simulate the human auditory system. Since the data obtained from discrete Fourier transformation has very high dimensionality (typically 256-dimension), it is not directly suitable to speech modeling. The common approach is to transform the high dimensional FFT spectrum to a low dimensional feature space using the so called mel-frequency cepstral coefficients (MFCC). Following is a brief description of MFCC.

Let $x_i = FFT(\frac{i}{N} \cdot f_{max}), i = 0, \cdots, N$ denote the magnitude squared discrete FFT spectrum, where $f_{max} = f_{samp}/2$, N is the number of data points in one FFT processing window, and f_{samp} is the sampling frequency, which is 16 kHz as in most standard speech database (such as TIMIT). The magnitude squared FFT spectrum is then passed through the mel-

functions in the non-stationary HMM without time warping. The point in time in each graph where the regression line is broken is the frame at which the "optimal" state transition occurs. We note that the shapes of the data sequences of these two tokens are very similar except the initial portion of the data in token one (left graph) is nearly twice as fast as that in token two (right graph). (But token one slows down during the mid portion.) With no time warping mechanism built into the regression function, a single set of polynomial coefficients trained from the two tokens having varying state-durations are not able to fit closely to both the tokens. The polynomial coefficients were trained in such a way that the fitting accuracy is compromised between the two tokens.

In Figure 4.2, we show the results of fitting the same two C1 data sequences as in Figure 4.1 but using the new non-stationary HMM containing time warping parameters. Again, the same two tokens were used to train the model according to the algorithm described in Section 3.2. In contrast to the results in Figure 4.1, with use of the new model, the fitting accuracy is high for both of the two tokens despite their varying state durations. This simultaneous high accuracy is achieved through use of different values of the auxiliary time warping parameters for the two tokens.

As another example, Figure 4.3 and Figure 4.4 are analogous to Figure 4.1 and Figure 4.2 except for use of C3 data sequences of two tokens from the TIMIT database. Again, use of time warping parameters in the non-stationary HMM produces more accurate fitting to the two data tokens simultaneously than without use of them.

REFERENCES

[1] L.E. Baum. "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes", *Inequalities*, Vol. 3, pp. 1–8, 1972.

[2] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Stat.*, Vol. 41, No. 1, pp. 164–171, 1970.

[3] S. B. Davis and P. Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 28, No. 4, 1980, pp.357–365.

[4] A.P. Dempster, N.M. Laird, and D.B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, Vol.39, pp. 1-38, 1977.

[5] L. Deng, "A generalized hidden Markov model with state-conditioned trend functions of time for the speech signal," *Signal Processing*, Vol.27, No.1, April 1992, pp. 65-78.

[6] L. Deng, M. Aksmanovic, D.X. Sun, and C.F.J. Wu. "Speech recognition using hidden Markov models with polynomial regression functions as non-stationary states," *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 4, 1994, pp. 507–520.

[7] L. Deng, M. Lennig, and P. Mermelstein. "Use of vowel duration information in a large vocabulary word recognizer," *J. Acoust. Soc. Am.*, Vol. 86, August, 1989, pp. 540-548.

$$\frac{\partial^2}{\partial \Lambda^2} L(\Theta^{(n)}, \Lambda^{(n)}) - \frac{\partial^2}{\partial \Lambda^2} L(\Theta^{(n)}, \Lambda^{(n+1)}) \to 0, \quad \text{as } n \to \infty,$$

and $\partial^2 L(\Theta^{(n)}, \Lambda^{(n+1)})/\partial \Lambda^2$ is negative definite with eigenvalues bounded away from zero, we conclude that $\partial^2 L(\Theta^{(n)}, \Lambda^{(n)})/\partial \Lambda^2$ is also negative definite with eigenvalues bounded away from zero. On the other hand, from (3.2) it is obvious that

$$\frac{\partial^2}{\partial \Theta \partial \Lambda} L(\Theta^{(n)}, \Lambda^{(n)}) = 0 \ .$$

Putting together the above facts, we have that $\partial^2 L(\Theta^{(n)}, \Lambda^{(n)})/\partial(\Theta, \Lambda)^2$ converges to a negative definite matrix, which is $\partial^2 L(\Theta^*, \Lambda^*)/\partial(\Theta, \Lambda)^2$. □

In summary, Theorem 3.2 gives the convergence property of the two-step iterative algorithm, and Corollary 3.1, combined with Theorem 3.3, proves that the proposed algorithm indeed leads to maximum likelihood estimates.

4. Results on fitting non-stationary hidden Markov models to Speech Data. In this section, we apply the state-dependent time warping HMM to fit the acoustic-parameter sequences from different renditions of the same word which vary in the sequence duration.

The first set of speech data was taken from two tokens of the word *peek* /pi:k/ spoken by a native English speaker with intentionally different speaking rates. The second set of speech data were selected from the DARPA TIMIT acoustic-phonetic continuous speech corpus. The two tokens used for illustration were excised from the same word *bike* in the sentence "sx332" uttered by two male speakers from dialect region 2 and region 7.

The raw speech data was in the form of digitally sampled signal at 16 kHz. The mel-frequency cepstral coefficients [3] were computed from the raw data with a Hamming window of duration 25.6 msec and with a frame rate of 10 msec. Non-stationary HMMs with three states and with order three in the state-dependent polynomial regression functions on time are used to fit speech data from the word *peek*. For speech data from the word *bike* (in TIMIT), four-state non-stationary HMMs with order three in the regression functions are used. For the sake of space saving and for purposes of illustration, we show here only the data fitting results for the first-order cepstral coefficient C1 from word *peek* and for the third-order cepstral coefficient C3 from *bike*. (Similar results were obtained for other cepstral coefficients).

The solid, less smoothed lines in both graphs of Figure 4.1 are C1 data sequences of two tokens uttered by the same speaker from the same word *peek*. The vertical axis represents the magnitude of C1 data and the horizontal axis is the frame number (frame size 10 msec). Superimposed on the same graphs in Figure 4.1 as dotted, more smoothed lines are the three sequentially advanced state-dependent polynomial regression

Applying the identifiability condition, we obtain

$$\begin{aligned}\epsilon &> \eta \sum_{j=1}^{J} \| (\Theta^{(n+j)}, \Lambda^{(n+j)}) - (\Theta^{(n+j-1)}, \Lambda^{(n+j-1)}) \| \\ &\geq \eta \| \sum_{j=1}^{J} (\Theta^{(n+j)}, \Lambda^{(n+j)}) - (\Theta^{(n+j-1)}, \Lambda^{(n+j-1)}) \| \\ &\geq \eta \| (\Theta^{(n+J)}, \Lambda^{(n+J)}) - (\Theta^{(n)}, \Lambda^{(n)}) \| .\end{aligned}$$

This inequality implies that the sequence $(\Theta^{(n)}, \Lambda^{(n)})$ is a Cauchy sequence, and it must converge, say, to (Θ^*, Λ^*). Note that (Θ^*, Λ^*) would be the ultimate estimate to be obtained if we would let the algorithm iterate infinitely many times. ∎

COROLLARY 3.1. *(Θ^*, Λ^*) is a stationary point of the likelihood function, i.e.,*

$$\frac{\partial}{\partial \Theta} L(\Theta^*, \Lambda^*) = 0, \quad \textit{and} \quad \frac{\partial}{\partial \Lambda} L(\Theta^*, \Lambda^*) = 0,$$

where the derivatives are taken as

$$\frac{\partial}{\partial \Theta} L(\Theta^*, \Lambda^*) = \left. \frac{\partial}{\partial \Theta} L(\Theta, \Lambda^*) \right|_{\Theta = \Theta^*} .$$

Proof. **Step 1** and **Step 2** in Section 3.2 (page 12) assure that

$$\frac{\partial}{\partial \Theta} L(\Theta^{(n+1)}, \Lambda^{(n+1)}) = 0, \text{ and } \frac{\partial}{\partial \Lambda} L(\Theta^{(n)}, \Lambda^{(n+1)}) = 0. \tag{3.2}$$

By taking limits of both sides (let $n \to \infty$) and then applying Theorem 3.2, the corollary follows immediately. ∎

Our final task is to prove that (Θ^*, Λ^*) is in fact the maximum likelihood estimate. For this it suffices to show that the second-order derivative $\partial^2 L(\Theta^*, \Lambda^*)/\partial(\Theta, \Lambda)^2$ is negative definite.

THEOREM 3.3. *Suppose that*

$$\frac{\partial^2}{\partial \Theta^2} L(\Theta^{(n)}, \Lambda^{(n)}) \quad \textit{and} \quad \frac{\partial^2}{\partial \Lambda^2} L(\Theta^{(n)}, \Lambda^{(n+1)})$$

are both negative definite with eigenvalues bounded away from zero. Then

$$\frac{\partial^2}{\partial(\Theta, \Lambda)^2} L(\Theta^*, \Lambda^*)$$

is negative definite.

Proof. Since

$$\begin{aligned}&\frac{\partial^2}{\partial(\Theta, \Lambda)^2} L(\Theta^{(n)}, \Lambda^{(n)}) \\ &= \begin{pmatrix} \partial^2 L(\Theta^{(n)}, \Lambda^{(n)})/\partial \Theta^2 & \partial^2 L(\Theta^{(n)}, \Lambda^{(n)})/\partial \Theta \partial \Lambda \\ \partial^2 L(\Theta^{(n)}, \Lambda^{(n)})/\partial \Theta d \Lambda & \partial^2 L(\Theta^{(n)}, \Lambda^{(n)})/\partial \Lambda^2 \end{pmatrix},\end{aligned}$$

we have

$$Q_r(\lambda_r) = \sum_{m=0}^{2M} c_m (\frac{1}{\lambda_r})^m$$

where

$$c_m = I(m \leq M) \sum_{t=1}^{n_r} t^m u_{(m+1),t} + \sum_{i=\max(1,m-M+1)}^{\min(m+1,M+1)} \sum_{t=1}^{n_r} t^m v_{(m+2-i),i}.$$

Minimizing $Q_r(\lambda_r)$ with respect to λ_r is a one-dimensional optimization problem and it can be easily solved by standard routines.

3.3. Convergence properties of the algorithm. In this section, we prove the convergence of the proposed optimization algorithm and its equivalence to the maximum likelihood estimation procedure.

Let $\Theta = (B_0, \cdots, B_M)$, $\Lambda = (\lambda_1, \cdots, \lambda_R)$ and let the estimates at the n-th iteration of the algorithm be $\Theta^{(n)}$ and $\Lambda^{(n)}$.

THEOREM 3.1. *The likelihood function in (3.1) is non-decreasing over iterations of the algorithm, or*

$$L(\Theta^{(n+1)}, \Lambda^{(n+1)}) \geq L(\Theta^{(n)}, \Lambda^{(n)})$$

Proof. The difference between the likelihood functions at n-th and $(n+1)$-th iteration

$$\begin{aligned} & L(\Theta^{(n+1)}, \Lambda^{(n+1)}) - L(\Theta^{(n)}, \Lambda^{(n)}) \\ = \; & \underbrace{[L(\Theta^{(n+1)}, \Lambda^{(n+1)}) - L(\Theta^{(n)}, \Lambda^{(n+1)})]}_{(I)} + \underbrace{[L(\Theta^{(n)}, \Lambda^{(n+1)}) - L(\Theta^{(n)}, \Lambda^{(n)})]}_{(II)}. \end{aligned}$$

Since **Step 1** and **Step 2** in Section 3.2 (page 12) guarantee the non-negativeness of (I) and (II), the conclusion is obvious. ∎

THEOREM 3.2. *Suppose that the following "identifiability condition" is satisfied:*

$$\begin{aligned} & L(\Theta^{(n+1)}, \Lambda^{(n+1)}) - L(\Theta^{(n)}, \Lambda^{(n)}) \\ \geq \; & \eta \,\|\, (\Theta^{(n+1)}, \Lambda^{(n+1)}) - (\Theta^{(n)}, \Lambda^{(n)}) \,\| \end{aligned}$$

where η is a fixed constant and $\| \cdot \|$ is the Euclidean norm. Then $(\Theta^{(n)}, \Lambda^{(n)})$ converge to some (Θ^*, Λ^*) in the parameter space.

Proof. From Theorem 3.1 the sequence $L(\Theta^{(n)}, \Lambda^{(n)})$ must converge to some value $L^* < \infty$. Hence, for any $\epsilon > 0$, there exists an $n(\epsilon)$ such that, for all $n \geq n(\epsilon)$ and all $J \geq 1$,

$$\begin{aligned} & \sum_{j=1}^{J} [L(\Theta^{(n+j)}, \Lambda^{(n+j)}) - L(\Theta^{(n+j-1)}, \Lambda^{(n+j-1)})] \\ = \; & L(\Theta^{(n+J)}, \Lambda^{(n+J)}) - L(\Theta^{(n)}, \Lambda^{(n)}) < \epsilon. \end{aligned}$$

where $\mathbf{B}_{(M+1)\times D}$, $(\mathbf{Y}_r)_{n_r\times D}$, $(\mathbf{X}_r)_{n_r\times(M+1)}$ are as follows.

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_0' \\ \vdots \\ \mathbf{B}_M' \end{pmatrix}, \ \mathbf{Y}_r = \begin{pmatrix} (\mathbf{O}_1^{(r)})' \\ \vdots \\ (\mathbf{O}_{n_r}^{(r)})' \end{pmatrix}, \ \mathbf{X}_r = \begin{pmatrix} f_0(\frac{1}{\lambda_r}) & \cdots & f_M(\frac{1}{\lambda_r}) \\ \vdots & & \vdots \\ f_0(\frac{n_r}{\lambda_r}) & \cdots & f_M(\frac{n_r}{\lambda_r}) \end{pmatrix}$$

Direct maximization of the likelihood function with respect to $(\mathbf{B}, \mathbf{\Sigma}, \boldsymbol{\lambda})$ is quite complex due to the nonlinearity introduced by $\boldsymbol{\lambda}$ in the regression functions. Therefore, we propose an two-step iterative algorithm for this multi-dimensional nonlinear regression problem. The two steps involve multiple linear regression and one-dimensional polynomial root finding.

Step 1: For given $(\lambda_1, \cdots, \lambda_{n_r})$, the regression parameters $\mathbf{B}$ and $\mathbf{\Sigma}$ are estimated as follows.

$$\begin{aligned} \hat{\mathbf{B}}_{(M+1)\times D} &= \left[\sum_{r=1}^{R}(\mathbf{X}_r'\mathbf{X}_r)\right]^{-1}\left[\sum_{r=1}^{R}(\mathbf{X}_r'\mathbf{Y}_r)\right] \\ \hat{\mathbf{\Sigma}}_{D\times D} &= (\sum_{r=1}^{R} n_r)^{-1}\sum_{r=1}^{R}\left(\mathbf{Y}_r - \mathbf{X}_r\hat{\mathbf{B}}\right)'\left(\mathbf{Y}_r - \mathbf{X}_r\hat{\mathbf{B}}\right) \end{aligned}$$

Step 2: For given $\mathbf{B}$ and $\mathbf{\Sigma}$, let

$$\begin{aligned} Q_r(\lambda_r) &= tr((\mathbf{Y}_r - \mathbf{X}_r(\lambda_r)\mathbf{B})\mathbf{\Sigma}^{-1}(\mathbf{Y}_r - \mathbf{X}_r(\lambda_r)\mathbf{B})') - tr(\mathbf{Y}_r\mathbf{\Sigma}^{-1}\mathbf{Y}_r') \\ &= tr(\mathbf{X}_r U) + tr(\mathbf{X}_r V \mathbf{X}_r') \end{aligned}$$

where $U_{(M+1)\times n_r} = -2\mathbf{B}\mathbf{\Sigma}^{-1}\mathbf{Y}_r'$, $\quad V_{(M+1)\times(M+1)} = \mathbf{B}\mathbf{B}'$. Since

$$\begin{aligned} tr(\mathbf{X}_r U) &= \sum_{t=1}^{n_r}\sum_{j=1}^{M+1} f_{j-1}(\frac{t}{\lambda_r})u_{j,t} = \sum_{t=1}^{n_r}\sum_{j=1}^{M+1}(\frac{t}{\lambda_r})^{j-1}u_{j,t} \\ &= \sum_{m=0}^{M}\left[\sum_{t=1}^{n_r} t^m u_{(m+1),t}\right](\frac{1}{\lambda_r})^m \end{aligned}$$

and

$$\begin{aligned} tr(\mathbf{X}_r V \mathbf{X}_r') &= tr(\mathbf{X}_r'\mathbf{X}_r V) = \sum_{i=1}^{M+1}\sum_{j=1}^{M+1}\left[\sum_{t=1}^{n_r} f_{i-1}(\frac{t}{\lambda_r})f_{j-1}(\frac{t}{\lambda_r})\right]v_{j,i} \\ &= \sum_{i=1}^{M+1}\sum_{j=1}^{M+1}\left[\sum_{t=1}^{n_r} t^{(i+j-2)}(\frac{1}{\lambda_r})^{(i+j-2)}\right]v_{j,i} \\ &= \sum_{m=0}^{2M}\left[\sum_{i=\max(1,m-M+1)}^{\min(m+1,M+1)}\ \sum_{t=1}^{n_r} t^m v_{(m+2-i),i}\right](\frac{1}{\lambda_r})^m \end{aligned}$$

(2) Recursion:

$$
\begin{aligned}
\delta_{t+1}(j,d) &= I(d=0)\cdot \max_{i\neq j}^{N} \max_{\tau=0}^{t-1} \delta_t(i,\tau)\cdot a_{ij}\cdot b_j(O_{t+1}|t_0=t-d;\boldsymbol{\theta}) \\
&+ I(d>0)\cdot \delta_t(j,d-1)\cdot a_{jj}\cdot b_j(O_{t+1}|t_0=t-d;\boldsymbol{\theta}) \\
\psi_{t+1}(j,d) &= I(d=0)\cdot \arg\max_{i\neq j}^{N} \max_{\tau=0}^{t-1} \delta_t(i,\tau)\cdot a_{ij} \\
&+ I(d>0)\cdot (j,d-1)
\end{aligned}
$$

for $t=1,\cdots,T$, $j=1,\cdots,N$, and $d=0,\cdots,t$, where $b_j(O_t|t_0=t-d;\boldsymbol{\theta})$ is the probability density of observation O_t at state j with duration d:

$$
b_i(\mathbf{O}_t|t_0;\boldsymbol{\theta}) = (|2\pi\boldsymbol{\Sigma}_i|)^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{O}_t-\boldsymbol{\mu}_i(t,t_0))'\boldsymbol{\Sigma}_i^{-1}(\mathbf{O}_t-\boldsymbol{\mu}_i(t,t_0))\right]
$$

(3) Termination:

$$
\begin{aligned}
P^* &= \max_{i=1}^{N} \max_{d_T=0}^{T-1} [\ \delta_T(i,d_T)\ ] \\
(\ s_T^*, d_T^*\ ) &= \arg\max_{i=1}^{N} \max_{d_T=0}^{T-1} [\ \delta_T(i,d_T)\ ]
\end{aligned}
$$

(4) State sequence backtracking (including duration) :

$$
(\ s_t^*, d_t^*\ ) = \psi_{t+1}(\ s_{t+1}^*, d_{t+1}^*\ ), \quad \text{for } t=T-1,\cdots,1.
$$

The final result is the optimal state sequence: $(s_1^*, s_2^*, \cdots, s_T^*)$.

Once all the state boundaries are determined in the segmentation step, the entire process for parameter estimation of the non-stationary HMM is broken down into several independent and essentially identical processes for estimating the parameters associated with each individual state. Therefore, for notational simplicity, we hereafter drop the state label i and consider only the parameter estimation for one single state. Given R tokens that are associated with the current state from the Viterbi segmentation:

$$
\begin{aligned}
\text{token 1:} \qquad \mathbf{O}^{(1)} &= (\mathbf{O}_1^{(1)},\cdots,\mathbf{O}_{n_1}^{(1)}) \\
\vdots \qquad &\quad \vdots \\
\text{token R:} \qquad \mathbf{O}^{(R)} &= (\mathbf{O}_1^{(R)},\cdots,\mathbf{O}_{n_1}^{(R)}),
\end{aligned}
$$

we can obtain the likelihood function based on these tokens as follows.

$$
\begin{aligned}
(3.1) \qquad L(\mathbf{B},\boldsymbol{\Sigma},\boldsymbol{\lambda}) &= \sum_{r=1}^{R}\Big[-\frac{n_r}{2}\log(|2\pi\boldsymbol{\Sigma}|) \\
&- \frac{1}{2}tr((\mathbf{Y}_r-\mathbf{X}_r\mathbf{B})\boldsymbol{\Sigma}^{-1}(\mathbf{Y}_r-\mathbf{X}_r\mathbf{B})')\Big]
\end{aligned}
$$

data at time t. Note that $\lambda_i^{(r)}$ serves only as auxiliary parameter whose role is to adjust the length of each training token for obtaining accurate estimates of the intrinsic model parameters $\mathbf{B}'_{i,m}s$ and $\mathbf{\Sigma}'_i s$. In the speech recognition step, the auxiliary parameters for the unknown utterance are estimated independently so as to adjust the duration of this new token to its own optimal scale for matching the non-stationary HMM obtained in the training step. Since the presence of λ in the model introduces nonlinearity to the objective function for optimization, there is no explicit solution to the maximization step in the EM algorithm developed before. Therefore, we developed a generalized segmental K-means algorithm to simplify the computation in parameter estimation.

3.2. Generalized K-means algorithm. In this section, we present an algorithm for estimating the parameters in the non-stationary HMM containing the state-dependent time warping mechanism. This algorithm is embedded within each step of a global iteration, whose goal is to reduce the training complexity involving a multiple-state hidden Markov model to essentially that involving only one state. The idea is in the same spirit as the segmental K-means algorithm [9]. This global segmental K-means-like algorithm involves two iterative steps: the segmentation step and the optimization step. The parameters $\mathbf{A} = (a_{ij})$ and $t_0^{(r)}$'s are readily determined from the result of the segmentation step for fixed $\mathbf{B}_{i,m}$'s and $\lambda_i^{(r)}$'s. The regression and time-warping parameters $\mathbf{B}_{i,m}$'s and $\lambda_i^{(r)}$'s are estimated in the optimization step for fixed $\mathbf{A} = (a_{ij})$ and $t_0^{(r)}$'s.

The segmentation step can be carried out by a generalized Viterbi algorithm described in the following. Let $\mathbf{O} = \{\mathbf{O}_1, \mathbf{O}_2, \cdots, \mathbf{O}_T\}$ be the observation sequence of length T. Let $\mathbf{S} = \{s_1, s_2, \cdots, s_T\}$ be the state sequence for the entire observation sequence and $\mathbf{d} = \{d_1, d_2, \cdots, d_T\}$ be the sojourn time sequence where d_t denotes the sojourn time of observation $\mathbf{O}_t$ in state s_t (i.e., $d_t = d$ means that state $s_{t-d-1} \neq s_{t-d} = \cdots = s_t$). Then the largest probability along a single state-sequence path up to time t, with duration time d at state j can be expressed as

$$\delta_t(i,d) = \max_{s_1,\cdots,s_{t-d-1}} P(\mathbf{O}_1, \cdots, \mathbf{O}_t; i \in [t-d, t])$$

where $i \in [t-d, t]$ means that state i begins at $t-d$ and ends at t. To keep track of the optimal state sequence, we use $\psi_t(j,d)$ to trace the most likely state (state identity and state sojourn time) at time $t-1$ given that $s_t = j$ and $d_t = d$. The modified Viterbi algorithm involves four steps:
(1) Initialization:

$$\begin{aligned} \delta_1(i,d) &= I(d=0) \cdot \pi_i \cdot b_i(\mathbf{O}_1 | t_0 = 0; \boldsymbol{\theta}) \\ \psi_1(i,d) &= (0,0) \end{aligned}$$

for $i = 1, \cdots, N$, and $d = 0, \cdots, T$, where $\{\pi_i, \cdots, \pi_N\}$ is the initial probability of the Markov states.

3. Non-stationary hidden Markov model with time-warping. The non-stationary hidden Markov model introduced in Section 2 takes a significant leap from the standard HMM in its generality and in its economical use of model parameters for approximating highly dynamical patterns of the speech signal. But despite these desirable properties, it has introduced its own specific problem in speech modeling and recognition applications. That is, speaking rate variation from one speech data sequence (which we call "token") to another, given the same underlying phonetic representation for the HMM states, must be normalized. Because, unlike the standard stationary-state HMM, the polynomial trend function for each state of the non-stationary HMM is varying with time, significant variability is necessarily introduced when using the same, single trend function to describe speech data from multiple (state-segmented) tokens from the same word with varying token durations. The varying token durations are the results of speaking rate variations and of other temporal factors in speech.

To alleviate this difficulty, we have developed an algorithm which implements time warping in the state-dependent polynomial regression functions with token-dependent auxiliary time warping parameters. The time warping parameters are called auxiliary parameters because they are not considered as intrinsic parameters of the model. They are used only as a tool to improve the accuracy in estimating the intrinsic parameters — state-dependent polynomial coefficients $\mathbf{B}_{i,m}$ in (1.1). These auxiliary parameters linearly adjust the state duration using one separate "optimal" scale for each individual token. The sole function of the auxiliary parameters is to group all the training tokens in an optimal way such that the variability of the state duration does not affect the estimation of the intrinsic parameters of the model. The proposed algorithm is a two-step iterative optimization procedure where estimation of the auxiliary parameters and the polynomial regression parameters is carried out alternatively.

Next, we describe the model formulation and the generalized segmental K-means algorithm for parameter estimation.

3.1. Model formulation. The non-stationary hidden Markov model with time-warping can be expressed by the following model for $\mathbf{O}_t$.

$$\mathbf{O}_t = \boldsymbol{\mu}_i(t, t_0) + \boldsymbol{\epsilon}_t = \sum_{m=0}^{M} \mathbf{B}_{i,m} \cdot f_m\left(\frac{t - t_0}{\lambda_i}\right) + \boldsymbol{\epsilon}_t$$

where λ is the time-warping parameter.

For given R tokens in the training data, the model for the rth token in a given state i is

$$\mathbf{O}_t^{(r)} = \sum_{m=0}^{M} \mathbf{B}_{i,m} \cdot f_m\left(\frac{t - t_0^{(r)}}{\lambda_i^{(r)}}\right) + \boldsymbol{\epsilon}_t$$

$t = 1, \cdots, n_r$, $r = 1, \cdots, R$, where $\mathbf{O}_t^{(r)}$ denotes the rth token of the training

$$
\begin{aligned}
&= \sum_{\text{all } \mathbf{S}} P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}_0) \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{d=1}^{T} I(i \in [t+1, t+d]) \sum_{\tau=1}^{d} \log b_i(\mathbf{O}_{t+\tau}|t_0 = t; \boldsymbol{\theta}) \\
&= \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{d=1}^{T} \left[P(\mathbf{O}, i \in [t+1, t+d]|\boldsymbol{\theta}_0) \sum_{\tau=1}^{d} \log b_i(\mathbf{O}_{t+\tau}|t_0 = t; \boldsymbol{\theta}) \right] \\
&= \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{d=1}^{T} \xi(i, t, d) \sum_{\tau=1}^{d} \log b_i(\mathbf{O}_{t+\tau}|t_0 = t; \boldsymbol{\theta})
\end{aligned}
$$

where $i \in [t+1, t+d]$ means that state i begins at $t+1$ and ends at $t+d$, and

$$
\begin{aligned}
\xi(i, t, d) &= P(\mathbf{O}, i \in [t+1, t+d]|\boldsymbol{\theta}_0) \\
&= \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{\tau=1}^{d} b_i(\mathbf{O}_{t+\tau}|t_0 = t; \boldsymbol{\theta}_0).
\end{aligned}
$$

Let

$$
\begin{aligned}
Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}_0) &= \sum_{t=1}^{T} \sum_{d=1}^{T} \xi(i, t, d) \left[\sum_{\tau=1}^{d} \log b_i(\mathbf{O}_{t+\tau}|t_0 = t) \right] \\
&= \sum_{t=1}^{T} \sum_{d=1}^{T} \xi(i, t, d) \left[-\frac{d}{2} \cdot \log(|2\pi \boldsymbol{\Sigma}_i|) \right. \\
&\qquad \left. - \frac{1}{2} \cdot tr((\mathbf{Y}_{t,d} - \mathbf{X}_{t,d}\mathbf{B}_i) \boldsymbol{\Sigma}^{-1} (\mathbf{Y}_{t,d} - \mathbf{X}_{t,d}\mathbf{B}_i)') \right]
\end{aligned}
$$

where $tr(\mathbf{C}) = \sum_{i=1}^{n} c_{ii}$ is the trace of a squared matrix $\mathbf{C}_{n \times n} = [c_{ij}]$, and $(\mathbf{Y}_{t,d})_{d \times D}$, $(\mathbf{X}_{t,d})_{d \times (M+1)}$, $(\mathbf{B}_i)_{(M+1) \times D}$ are as follows.

$$
\mathbf{Y}_{t,d} = \begin{pmatrix} \mathbf{O}'_{t+1} \\ \vdots \\ \mathbf{O}'_{t+d} \end{pmatrix}, \ \mathbf{X}_{t,d} = \begin{pmatrix} f_0(1) & \cdots & f_M(1) \\ \vdots & & \vdots \\ f_0(d) & \cdots & f_M(d) \end{pmatrix}, \ \mathbf{B}_i = \begin{pmatrix} \mathbf{B}'_{i,0} \\ \vdots \\ \mathbf{B}'_{i,M} \end{pmatrix}
$$

To estimate the regression parameters for state i, we only need to maximize $Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}_0)$. Using the standard multivariate regression method (see [15]), we get the estimates as follows.

$$
\begin{aligned}
\hat{\mathbf{B}}_i &= \left[\sum_{t=1}^{T} \sum_{d=1}^{T} \xi(i, t, d) (\mathbf{X}'_{t,d} \mathbf{X}_{t,d}) \right]^{-1} \left[\sum_{t=1}^{T} \sum_{d=1}^{T} \xi(i, t, d) (\mathbf{X}'_{t,d} \mathbf{Y}_{t,d}) \right] \\
\hat{\boldsymbol{\Sigma}}_i &= \frac{\sum_{t=1}^{T} \sum_{d=1}^{T} \xi(i, t, d) \left(\mathbf{Y}_{t,d} - \mathbf{X}_{t,d} \hat{\mathbf{B}}_i \right)' \left(\mathbf{Y}_{t,d} - \mathbf{X}_{t,d} \hat{\mathbf{B}}_i \right)}{\sum_{t=1}^{T} \sum_{d=1}^{T} d \cdot \xi(i, t, d)}
\end{aligned}
$$

We adopt the convention that $a_{0j} = \pi_j$ is the initial probability of staying at state j and

$$\alpha_t(i) = \begin{cases} 1 & t = 0, i = 0 \\ 0 & t = 0, i = 1, \cdots, N \\ 0 & i = 0, t = 1, \cdots, T \end{cases}$$

Next, we will describe the estimation of transition probabilities and of regression model parameters separately.

2.1. Estimation of transition probabilities. The transition probabilities can be estimated by the Lagrange multiplier method. By setting

$$\frac{\partial (Q + \sum_{i=1}^{N} c_i (\sum_{j=1}^{N} a_{ij} - 1))}{\partial a_{ij}} = 0$$

$$\hat{a}_{ij} = \sum_{\text{all } \mathbf{s}} \left[P(\mathbf{O}, \mathbf{S} | \boldsymbol{\theta}_0) \cdot \sum_{l=1}^{r} I(s_{l-1} = i, s_l = j) \right] / c_i$$

where

$$c_i = \sum_{j=1}^{N} \sum_{\text{all } \mathbf{s}} [P(\mathbf{O}, \mathbf{S} | \boldsymbol{\theta}_0) \cdot \sum_{l=1}^{r} I(s_{l-1} = i, s_l = j)].$$

The summation over all possible state sequence $\mathbf{S}$ can be simplified as follows:

$$\begin{aligned}
& \sum_{\text{all } \mathbf{s}} \left[P(\mathbf{O}, \mathbf{S} | \boldsymbol{\theta}_0) \cdot \sum_{l=1}^{r} I(s_{l-1} = i, s_l = j) \right] \\
= & \sum_{\text{all } \mathbf{s}} \left[P(\mathbf{O}, \mathbf{S} | \boldsymbol{\theta}_0) \cdot \sum_{t=1}^{T} I(i \text{ ends at t}, j \text{ begins at } (t+1)) \right] \\
= & \sum_{t=1}^{T} P(\mathbf{O}_1, \cdots, \mathbf{O}_T;\ i \text{ ends at t}, j \text{ begins at } (t+1)\ | \boldsymbol{\theta}_0) \\
= & \sum_{t=1}^{T} \alpha_t(i) \cdot a_{ij} \cdot \beta_t^*(j)
\end{aligned}$$

2.2. Regression parameters. Consider the part of the objective function that is associated with the regression parameters.

$$\begin{aligned}
& Q_R(\boldsymbol{\theta} | \boldsymbol{\theta}_0) \\
= & \sum_{\text{all } \mathbf{s}} P(\mathbf{O}, \mathbf{S} | \boldsymbol{\theta}_0) \sum_{l=1}^{r} \sum_{\tau=1}^{d_l} \log b_{s_l}(\mathbf{O}_{\delta_{l-1}+\tau} | t_0 = \delta_{l-1}; \boldsymbol{\theta})
\end{aligned}$$

Therefore, the original optimization problem can be implemented by an iterative procedure consisting of two steps in each iteration. First, given an existing parameter vector $\boldsymbol{\theta}_0$, form the objective function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_0)$. Second, maximize $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_0)$ with respect to $\boldsymbol{\theta}$. For the next iteration, $\boldsymbol{\theta}_0$ is replaced by the new optimized parameter obtained in the second step. In the following, we describe the details of the second step in maximizing the objective function

$$
\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}_0) &= \sum_{\text{all } \mathbf{s}} P(\mathbf{O},\mathbf{S}|\boldsymbol{\theta}_0)\log P(\mathbf{O},\mathbf{S}|\boldsymbol{\theta}) \\
&= \sum_{\text{all } \mathbf{s}} P(\mathbf{O},\mathbf{S}|\boldsymbol{\theta}_0)\sum_{l=1}^{r}\log a_{s_{l-1},s_l} \\
&\quad +\sum_{\text{all } \mathbf{s}} P(\mathbf{O},\mathbf{S}|\boldsymbol{\theta}_0)\sum_{l=1}^{r}\sum_{\tau=1}^{d_l}\log b_{s_l}(\mathbf{O}_{\delta_{l-1}+\tau}|t_0=\delta_{l-1};\boldsymbol{\theta}) \\
&= Q_A(\boldsymbol{\theta}|\boldsymbol{\theta}_0)+Q_R(\boldsymbol{\theta}|\boldsymbol{\theta}_0)
\end{aligned}
$$

where $Q_A(\boldsymbol{\theta}|\boldsymbol{\theta}_0)$ is for the estimation of transition probabilities, and $Q_R(\boldsymbol{\theta}|\boldsymbol{\theta}_0)$ if for the estimation of regression parameters. To simplified the notation in the formulae for model parameters estimation, we use the four forward-backward variables as in [13,16] for explicit state duration HMM:

$$
\begin{aligned}
\alpha_t(i) &= P(\mathbf{O}_1,\cdots,\mathbf{O}_t;\ \text{state } i \text{ ends at } t|\boldsymbol{\theta}_0) \\
\alpha_t^*(i) &= P(\mathbf{O}_1,\cdots,\mathbf{O}_t;\ \text{state } i \text{ begins at } t+1|\boldsymbol{\theta}_0) \\
\beta_t(i) &= P(\mathbf{O}_{t+1},\cdots,\mathbf{O}_T|\ \text{state } i \text{ ends at } t;\boldsymbol{\theta}_0) \\
\beta_t^*(i) &= P(\mathbf{O}_{t+1},\cdots,\mathbf{O}_T|\ \text{state } i \text{ begins at } t+1;\boldsymbol{\theta}_0)
\end{aligned}
$$

These probabilities can be computed by the following recursive formulae.

$$
\begin{aligned}
\alpha_t^*(j) &= \sum_{i=0,i\neq j}^{N}\alpha_t(i)a_{ij} \\
\alpha_t(i) &= \sum_{d=1}^{t}\alpha_t^*(i)p_i(d)\prod_{\tau=t-d+1}^{t}b_i(\mathbf{O}_\tau|t_0=t-d;\boldsymbol{\theta}_0) \\
\beta_t(i) &= \sum_{j=1,j\neq i}^{N}a_{ij}\beta_t^*(j) \\
\beta_t^*(i) &= \sum_{d=1}^{T-t}\beta_t(i)p_i(d)\prod_{\tau=t+1}^{t+d}b_i(\mathbf{O}_\tau|t_0=t;\boldsymbol{\theta}_0)
\end{aligned}
$$

for $t=1,\cdot,T$, $i,j=1,\cdots,N$, where

$$p_i(d)=P(\text{ state } i \text{ begins at } t+1 \text{ and ends at } t+d\ )=(a_{ii})^{d-1}(1-a_{ii}).$$

variable temporal factors in speech, such as speaking rate, style, etc., appropriate procedure for time warping is needed for accurate estimation of the parameters in the regression models. To deal with this source of variation in multiple training tokens, we introduce the concept of auxiliary time-warping parameters in Section 3. In Section 4, we present some experimental results on fitting the non-stationary HMMs to speech data, which are different renditions of the same word but vary significantly over the speech token length. Comparisons between the data fitting results with and without using the time-warping mechanism illustrate the need for time warping in the non-stationary HMM and show effectiveness of the proposed algorithm for achieving the desired time warping.

2. Estimation of model parameters in non-stationary hidden Markov models. In principle, the parameters of the regression-based non-stationary HMM can be estimated by maximizing the likelihood function $P(\mathbf{O}|\boldsymbol{\theta})$ of the training data. However, the likelihood function, which involves the summation over all possible state sequence, is too complex to be maximized directly. As for the standard HMM, we adopt the paradigm of EM optimization procedure ([4]), also known as Baum-Welch algorithm ([2,1,16,10]). Instead of maximizing the original likelihood function $P(\mathbf{O}|\boldsymbol{\theta})$, we consider maximizing the following objective function:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}_0) = \sum_{\text{all } \mathbf{s}} P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}_0) \cdot \log P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta})$$

The relationship between the original likelihood function and the new objective function is illustrated in Theorem 2.1.

THEOREM 2.1. *If there exist $\boldsymbol{\theta}^*$ such that $Q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_0) > Q(\boldsymbol{\theta}_0|\boldsymbol{\theta}_0)$, then*

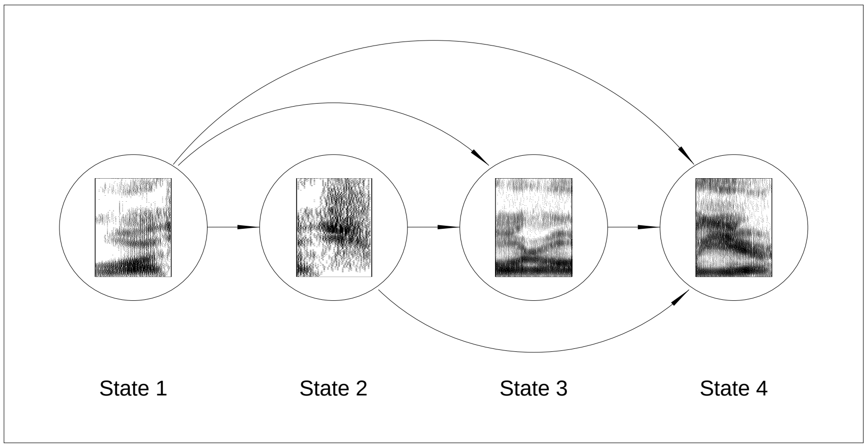
$$P(\mathbf{O}|\boldsymbol{\theta}^*) > P(\mathbf{O}|\boldsymbol{\theta}_0).$$

Proof.

$$\begin{aligned}
\log \frac{P(\mathbf{O}|\boldsymbol{\theta}^*)}{P(\mathbf{O}|\boldsymbol{\theta}_0)} &= \log \sum_{\text{all } \mathbf{s}} \left[\frac{P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}^*)}{P(\mathbf{O}|\boldsymbol{\theta}_0)} \right] \\
&= \log \sum_{\text{all } \mathbf{s}} \left[P(\mathbf{S}|\mathbf{O}; \boldsymbol{\theta}_0) \frac{P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}^*)}{P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}_0)} \right] = \log E \left(\frac{P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}^*)}{P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}_0)} \middle| \mathbf{O}, \boldsymbol{\theta}_0 \right) \\
&\geq E \left(\log \frac{P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}^*)}{P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}_0)} \middle| \mathbf{O}, \boldsymbol{\theta}_0 \right) \\
&= \sum_{\text{all } \mathbf{s}} [P(\mathbf{S}|\mathbf{O}; \boldsymbol{\theta}_0)(\log P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}^*) - \log P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}_0))] \\
&= \frac{Q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_0) - Q(\boldsymbol{\theta}_0|\boldsymbol{\theta}_0)}{P(\mathbf{O}|\boldsymbol{\theta}_0)} > 0
\end{aligned}$$

□

FIG. 1.2. *Illustration of HMM structure and the corresponding speech signals*



denote the r states by $s_1, \cdots, s_r$. The state duration associated with the r states are denoted by $d_1, \cdots, d_r$ and $\delta_l = \sum_{i=1}^{l} d_i$ is the cumulative duration up to state l. Hence the original state sequence can be described as $\mathbf{S} = (s_1, d_1; s_2, d_2; \cdots; s_r, d_r)$. With these notations, the likelihood function of the observation sequence can be expressed as

$$
\begin{aligned}
P(\mathbf{O}|\boldsymbol{\theta}) &= \sum_{\text{all } \mathbf{s}} P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}) = \sum_{\text{all } \mathbf{s}} P(\mathbf{O}|\mathbf{S}; \boldsymbol{\theta}) P(\mathbf{S}|\boldsymbol{\theta}) \\
&= \sum_{\text{all } \mathbf{s}} \prod_{l=1}^{r} \left[a_{s_{l-1}, s_l} \cdot \prod_{\tau=1}^{d_l} b_{s_l}(\mathbf{O}_{\delta_{l-1}+\tau} | t_0 = \delta_{l-1}; \boldsymbol{\theta}) \right]
\end{aligned}
$$

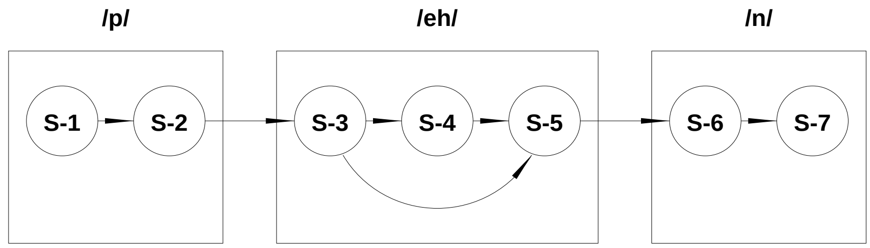
where

$$
b_i(\mathbf{O}_t | t_0) = (|2\pi \boldsymbol{\Sigma}_i|)^{-\frac{1}{2}} \exp\left[-\frac{1}{2} (\mathbf{O}_t - \boldsymbol{\mu}_i(t, t_0))' \boldsymbol{\Sigma}_i^{-1} (\mathbf{O}_t - \boldsymbol{\mu}_i(t, t_0)) \right]
$$

A related work for modeling the non-stationary features of speech signals is the dynamic-programming based template matching algorithm proposed in [8]. The major difference is that the proposed HMMs are parametric models while the dynamic-programming based template matching algorithm is non-parametric in nature.

To estimate the model parameters of the non-stationary HMMs, we propose a generalized EM algorithm or Baum-Welch algorithm which incorporates the state duration information in both forward-backward probability calculation and parameter re-estimation in Section 2. This method is related to the one for estimating model parameters in the explicit duration model [13]. When multiple training tokens are influenced by highly

FIG. 1.1. *Concatenation of three HMMs to a larger HMM for word "pen"*



In standard HMM, the observations in the same state are modeled by independent identically distributed random variables with Gaussian distributions or mixtures of Gaussian distributions. In other words, the observations within a state are assumed to be stationary. Therefore, the standard HMMs approximate the continuously varying speech signals in a piece-wise constant manner. Such approximations would be reasonably good when each state is intended to represent only a short portion of sonorant sounds. However, since the acoustic patterns of continuously spoken speech sounds are nearly never stationary in nature (e.g. [18]), it would be desirable to improve this rather poor piece-wise constant approximation in general. For example, Figure 1.2 shows a four-state HMM with the observation sequences within each state represented by segments of spectrogram, where the horizontal and vertical coordinates represent time and frequency, and darkness represents the spectral energy at each time-frequency point. It can be seen that the signals at different time points within a state are relatively similar to each other as compared with the signals in other states. However, there is also strong indication that the signals within one state are not quite stationary. The dynamic movements of the signals within a state can be observed in states 3 and state 4.

In this paper, we present a general framework of non-stationary HMM where the probability of an observation sequence within an HMM state is modeled by a parametric regression model. Consider the following model for observation $\mathbf{O}_t$ in a given state i:

$$(\mathbf{O}_t)_{D\times 1} = \boldsymbol{\mu}_i(t,t_0) + \boldsymbol{\epsilon}_t = \sum_{m=0}^{M} \mathbf{B}_{i,m} \cdot f_m(t-t_0) + \boldsymbol{\epsilon}_t \tag{1.1}$$

where i is the label of the state in the HMM, $\boldsymbol{\epsilon}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma}_i)$, D is the dimension of the speech feature vector, t_0 is the ending time of the previous state in the same HMM, and $t-t_0$ is the sojourn time in state i. The regression function $f_m(\cdot)$ is taken to be the mth order polynomial function in our following discussion, i.e., $f_m(t-t_0) = (t-t_0)^m$, although other parametrization may also be considered (e.g., [6]). For a given state sequence $\mathbf{S}$, let r be the total number of states that have been visited, and

Then joint probability of $\mathbf{O}$ and $\mathbf{S}$ is

$$P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}) = \prod_{t=1}^{T} [a_{s_{t-1},s_t} \cdot b_{s_t}(\mathbf{O}_t)],$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N; \mathbf{A})$, and $a_{s_0,s_1} = \pi_{s_1}$ is defined as the initial probability of staying at state s_1. When the state sequence is unknown, the likelihood function of the observation sequence $\mathbf{O}$ can be expressed as

$$P(\mathbf{O}|\boldsymbol{\theta}) = \sum_{\text{all } \mathbf{S}} P(\mathbf{O}, \mathbf{S}|\boldsymbol{\theta}) = \sum_{\text{all } \mathbf{S}} \prod_{t=1}^{T} [a_{s_{t-1},s_t} \cdot b_{s_t}(\mathbf{O}_t)]$$

where the summation is over all possible state sequences. Therefore, an HMM can be specified by the number of states N and the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N; \mathbf{A})$.

In speech recognition applications, HMMs are typically associated with speech sounds, such as words, syllables, phonemes or other phonetic units. The states of an HMM usually represent sub-phonemic speech units whose segmental boundaries in the acoustic signal are rather vague and are not known in the training data. The speech units for HMM modeling are usually selected according to the following three guidelines: First, the set of speech units should be capable of representing or synthesizing all the lexical entries in the vocabulary such that each word in the vocabulary can be specified and be distinguished from each other. Second, it is desirable to select speech units whose acoustic realizations are relatively invariant and consistent. Third, the inventory of the speech units should be sufficiently small so that the demand for the amount of training data is within reasonable range.

Several different HMMs can be concatenated to form a larger HMM by adjusting the transition probability from the final state of one HMM to the initial state of another HMM. In the example shown in Figure 1.1, there are three separate HMMs representing three phonemes /p/, /eh/, and /n/. For instance, the HMM for phoneme /eh/ consists of three states each representing certain sub-phonemic or microsegmental speech units. The arrow in the graph represent the feasible transitions between the states. The three HMMs can then be concatenated to form a new HMM for the word "pen".

The parameters of the HMMs can be estimated by maximizing the likelihood function $P(\mathbf{O}|\boldsymbol{\theta})$ based on the entire training data using the Baum-Welch or segmental K-means training algorithm ([1,9]). During speech recognition, the probability of an incoming utterance $\mathbf{O}_{\text{new}}$ (i.e., the new observation sequence) is evaluated using all the HMMs in the inventory, and the one with the largest likelihood is recognized as the speech sound or the sequence of sounds that generates $\mathbf{O}_{\text{new}}$. In other words,

$$\mathbf{O}_{\text{new}} \in \text{ Class}(i) \quad \text{if} \quad P(\mathbf{O}_{\text{new}}|\boldsymbol{\theta}_i) = \max_j P(\mathbf{O}_{\text{new}}|\boldsymbol{\theta}_j)$$

NON-STATIONARY HIDDEN MARKOV MODELS FOR SPEECH RECOGNITION

DON X. SUN[1] AND LI DENG[2] *

Key words: *Baum-Welch algorithm, Hidden Markov models, Regression models, Segmental K-means algorithm, Time warping, Viterbi algorithm.*

Abstract. The standard hidden Markov models (HMM) assume local or state-conditioned stationarity of the signals being modeled. In this article, we present some recent development in generalizing the standard HMM to incorporate the local dynamic patterns as well as the global non-stationarity for speech signal modeling. The major component of the proposed non-stationary HMMs is the parametric regression models for individual HMM states. The regression functions are intended for characterizing the dynamic movements of the signals within a HMM state. Both the EM algorithm (or Baum-Welch algorithm) and the segmental K-means algorithms are generalized to accommodate the complex state duration information needed for the estimation of regression parameters. To allow for the flexibility of linear time warping in individual HMM states, an efficient algorithm is developed with the use of token-dependent auxiliary parameters. Although the auxiliary parameters are of no interest in themselves for modeling speech sound patterns, they provide an intermediate tool for achieving maximal accuracy in estimating the parameters of the regression models.

1. Introduction. Hidden Markov models (HMM) have been widely used in automatic speech recognition systems for modeling non-stationary time-varying signals [16,10,11]. (A brief description of speech signal representation is provided in Appendix A.) The basic element of an N-state HMM is a Markov chain structure and a probabilistic model of the observation vectors for a given state in the Markov chain ([1,14,10]).

To describe the standard HMM, let s_t denote the state of the system at time t. The Markov chain structure is specified by a state transition probability matrix $\mathbf{A} = [a_{ij}]$, where $a_{ij} = P(s_t = j | s_{t-1} = i), \quad 1 \le i, j \le N$. The probability of the observation vector $\mathbf{O}_t$ in a given state i is denoted by $b_i(\mathbf{O}_t, \boldsymbol{\theta}_i)$. Assuming that all the observation vectors are independent, for a given state sequence $\mathbf{S} = (s_1, \cdots, s_T)$, the probability of the entire observation sequence $\mathbf{O} = (\mathbf{O}_1, \cdots, \mathbf{O}_T)$ is

$$P(\mathbf{O}|\mathbf{S}; \boldsymbol{\theta}) = \prod_{t=1}^{T} b_{s_t}(\mathbf{O}_t).$$

* [1] Prof. Sun is with the Dept. of Applied Mathematics and Statistics at the State University of New York at Stony Brook, NY 11794-3600; [2] Prof. Deng is with the Dept. of Electrical and Computer Engineering, University of Waterloo, N2L 3G1, Canada. The support for this work is provided by University of Waterloo Interdisciplinary Grants, Natural Sciences and Engineering Research Council of Canada and by National Science Foundation Grant Number DMS-9057429.