Speech Trajectory Discrimination Using the Minimum Classification Error Learning

Rathinavelu Chengalvarayan, *Member, IEEE*, and Li Deng, *Senior Member, IEEE*

***Abstract*— In this paper, we extend the maximum likelihood (ML) training algorithm to the minimum classification error (MCE) training algorithm for discriminatively estimating the state-dependent polynomial coefficients in the stochastic trajectory model or the trended hidden Markov model (HMM) originally proposed in [2]. The main motivation of this extension is the new model space for smoothness-constrained, state-bound speech trajectories associated with the trended HMM, contrasting the conventional, stationary-state HMM, which describes only the piecewise-constant "degraded trajectories" in the observation data. The discriminative training implemented for the trended HMM has the potential to utilize this new, constrained model space, thereby providing stronger power to disambiguate the observational trajectories generated from nonstationary sources corresponding to different speech classes. Phonetic classification results are reported which demonstrate consistent performance improvements with use of the MCE-trained trended HMM both over the regular ML-trained trended HMM and over the MCE-trained stationary-state HMM.**

***Index Terms*—Discrimination, MCE training, mixture trended HMM, phonetic classification, trajectory.**

I. Introduction

THE formulation of the trended hidden Markov model (HMM), also called the parametric nonstationary-state HMM or parametric stochastic trajectory model, has been used in speech recognition applications for the past few years by a number of research groups [2], [4], [6]–[8], [13]. The main motivation for using the trajectory model is its advantage of capturing smoothed temporal variations ubiquitously observed in the spectral aspects of speech data. This leads to an effective means of succinct parameterization of the segment-bound temporal correlation structure of speech which cannot be modeled by the conventional HMM. The model parameters of the trended HMM, especially the state-dependent time-varying Gaussian means, used in the past were trained by a modified Viterbi algorithm based on the joint-state maximum likelihood (ML) principle [4]. The method of ML, however, is generally not optimal in terms of minimizing classification error rate in classification tasks in which the observation is assumed to be produced by one of the many source classes [1]. Only the in-class information is available to train each model when the ML approach is used; that is, a separate model is constructed for each class (a phone or a word, for example) and is trained on tokens of that class only. This type of ML-based training is not discriminative, since each model is built independently and one intends only to model the acoustic observations representative of that class. Discrimination can be improved if out-of-class information is jointly used in training the models. Discriminative training methods do not aim at construction of the best model of observation data for each class, but instead attempt to predict whether a given observation belongs to one class or another. Since such methods focus on the use of parameters on the decision surface among different classes and not on the distribution of observations themselves, they have theoretical advantages over the ML method in term of classification performance. An example of discriminative training is the minimum classification error (MCE) training algorithm, which has been implemented in various forms (e.g., [1], [9], [12]).

In the study reported in this paper, the MCE algorithm is extended from the earlier formulation that applies to the conventional or stationary-state HMM to the trended HMM. In particular, the MCE algorithm is used to discriminatively estimate the state-dependent and mixture-dependent polynomial coefficients in the trended HMM based on a gradient-descent method. The properties of the MCE formulation for training the trended HMM are analyzed by examining goodness-of-fit of the raw speech data to the polynomial trajectories in the model, and comparative experimental results on phonetic classification are reported which demonstrated the effectiveness of the MCE algorithm for the trended HMM. All our experimental results have substantiated our theoretical reasoning and motivation for the application of the MCE algorithm to the trended HMM. That is, given that the trended HMM tracks the stochastic trajectories of the speech data, new degree of freedom in the space of the modeled trajectories associated with the trended HMM, together with the constraint forcing the modeled trajectory to be a (state-bound) smoothed function of time, should allow discriminative training to exploit interactions between the new model space and the constraint. This should then allow discriminative training to gain more power to disambiguate the different trajectories associated with different speech classes.

This paper is organized as follows. The formulation of a mixture version of the trended HMM including the mixture-dependent and state-dependent polynomial coefficients is provided in Section II. In Section III, the basic principle of the MCE training is summarized, and the training procedure

Manuscript received November 7, 1996; revised December 18, 1997. This work was supported by the Natural Sciences and Engineering Research Council of Canada. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kuldip K. Paliwal.

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ont., Canada N2L 3G1 (e-mail: deng@crg3.uwaterloo.ca).

Publisher Item Identifier S 1063-6676(98)07786-4.

applied specifically to the optimization of the state-dependent polynomial functions and the variances in the trended HMM is described in detail. Experimental results on fitting the trended HMM to real speech data are provided in Section IV. These results, presented as comparisons between use of the ML training algorithm and use of the MCE training algorithm, illustrate the need for discriminative training in the trended HMM and show effectiveness of the algorithm in achieving enhanced discrimination ability. In Section V, we report phonetic classification results obtained using the TIMIT data base. These results demonstrate consistent performance improvements with use of the MCE-trained trended HMM over the regular ML-trained trended HMM and over the MCE-trained conventional stationary-state HMM. Concluding remarks are finally given in Section VI.

II. Mixture Trended HMM

The trended HMM is of a data-generative type and can be described in the following equation for the generation of the acoustic observation sequence:

$$\begin{aligned}\mathcal{O}_t &= F_t(i,m) + R_t(\Sigma_{i,m}),\\ &= \sum_{p=0}^{P} B_{i,m}(p)(t-\tau_i)^p + R_t(\Sigma_{i,m}),\\ &\qquad m = 1,2,\cdots,M; i = 1,2,\cdots,N \end{aligned} \tag{1}$$

where $\mathcal{O}_t, t = 1,2,\cdots,T$ is a modeled observation data sequence of length T, within the HMM state indexed by $i; B_{i,m}(p)$ are mixture-dependent and state-dependent polynomial regression coefficients of order P indexed by mixture component m and by state i; and the term R_t is the stationary residual (after the data-fitting by the first term F_t) assumed to be independent and identically distributed (i.i.d.) and zero-mean Gaussian source characterized by state (i)-dependent, mixture (m)-dependent, but time-invariant diagonal covariance matrix $\Sigma_{i,m}$.

In the conventional, stationary-state HMM [15], the first term in (1) is only a function of state i, not a function of time t. Note also that the polynomials for each state depend not only on the coefficients $B_{i,m}(p)$, but also on the time-shift parameter τ_i. The term $t-\tau_i$ represents the sojourn time in state i at time t, where τ_i registers the time when state i in the HMM is just entered before regression on time takes place. Polynomial coefficients $B_{i,m}(p)$ (for state i and mixture component m) are considered as true model parameters and τ_i is merely an auxiliary parameter for the purpose of obtaining maximal accuracy in estimating $B_{i,m}(p)$. In the recognition step, τ_i is again estimated as the auxiliary parameter so as to achieve a maximal score in matching the model to the unknown utterance over all possible τ_i values.

In summary, a mixture trended HMM consists of the following parameter quadruple $[A, B, \Sigma, W]$.

1) $A = [a_{i,j}], i,j = 1,2,\cdots,N$ is the transition probability matrix of the underlying Markov chain with a total of N states.
2) $B = [B_{i,m}(p)], i = 1,2,\cdots,N, m = 1,2,\cdots,M,$ and $p = 1,2,\cdots,P$ are the polynomial coefficients, of order P and associated with state i and mixture m, in the state-dependent deterministic regression function of time. (Dimensionality of vector $B_{i,m}(p)$ is n, the same as that of feature vector O_t.)
3) $\Sigma = [\Sigma_{i,m}], i = 1,2,\cdots,N$ and $m = 1,2,\cdots,M$ are the time-invariant covariance matrices (dimensionality of $n \times n$) of the zero-mean Gaussian i.i.d. residual signals $R_t(\Sigma_{i,m})$. (These matrices are also state and mixture dependent.)
4) $W = [w_{i,m}], i = 1,2,\cdots,N$ and $m = 1,2,\cdots,M$ are the mixture weights.

III. Discriminative Training for Mixture Trended HMM

One major purpose of this study is to develop and implement the MCE-based discriminative training paradigm in the context of the trended HMM for achieving optimal estimation of the state-dependent polynomial coefficients. Let $\Phi_j, j = 1,2,\cdots,\mathcal{K}$, denote the parameter set characterizing the trended HMM for the jth class, where $\mathcal{K}$ is the total number of classes. The classifier based on these $\mathcal{K}$ class models can be characterized by $\Phi = \{\Phi_1, \Phi_2, \cdots, \Phi_{\mathcal{K}}\}$. The purpose of the MCE-based discriminative training is to find the parameter set Φ such that the probability of misclassifying all the training tokens is minimized.

Let $g_j(\mathcal{O}, \Phi)$ denote the log-likelihood associated with the optimal state sequence Θ for the input token $\mathcal{O}$, obtained by applying the Viterbi algorithm using model Φ_j for the jth class. Then, for the utterance $\mathcal{O}$ (from class c), the misclassification measure $d_c(\mathcal{O}, \Phi)$ is determined by

$$d_c(\mathcal{O}, \Phi) = -g_c(\mathcal{O}, \Phi) + g_\chi(\mathcal{O}, \Phi) \tag{2}$$

where χ denotes the incorrect model with the highest log-likelihood (i.e., the most confusible class). In this definition, a negative value of $d_c(\mathcal{O}, \Phi)$ corresponds to a correct classification. The definition in (2) focuses on the comparison between the true model and only the closest-competing wrong model, an approximation which we adopt in this study for computation efficiency. (A more general form of the misclassification measure using the log-likelihoods from all models can be found in [1] and [16]). A loss function with respect to the input token is defined in terms of the misclassification measure given by

$$\Upsilon(\mathcal{O}, \Phi) = \frac{1}{1 + e^{-d_c(\mathcal{O}, \Phi)}} \tag{3}$$

which projects $d_c(\mathcal{O}, \Phi)$ into the interval [0,1]. Note that the loss function $\Upsilon(\mathcal{O}, \Phi)$ is directly related to the classification error rate and is first-order differentiable with respect to all the model parameters of $\Phi_j, j = 1,2,\cdots,\mathcal{K}$. Once the objective function in (3) is determined, the MCE-based discriminative training is reduced to finding the gradient of the objective function with respect to all the model parameters and to using the computed gradient to update the model parameters in an iterative manner.

A. Gradient Descent Method

Let ϕ be a parameter in the model Φ. Provided that $\Upsilon(\mathcal{O},\Phi)$ is differentiable with respect to ϕ, that parameter is adjusted in the gradient descent method according to

$$\hat{\phi} = \phi - \epsilon \frac{\partial \Upsilon(\mathcal{O},\Phi)}{\partial \phi}, \quad \text{or}$$
$$\hat{\phi} = \phi - \epsilon \underbrace{\Upsilon(\mathcal{O},\Phi)(1-\Upsilon(\mathcal{O},\Phi))}_{\psi} \frac{\partial d_c(\mathcal{O},\Phi)}{\partial \phi}. \tag{4}$$

In (4), $\hat{\phi}$ is the new estimate of the parameter and ϵ is a small positive constant that monotonically decreases as the iteration number increases. This gradient descent method is iteratively applied to all training tokens in a sequential manner (for all model parameters) to minimize the loss function during the training process.

Some intuitive explanations for (4) are given here. In the case of near error-free classification (i.e., $\Upsilon(\mathcal{O},\Phi) \approx 0$), or in the case of a complete loss (very poor classification; i.e., $\Upsilon(\mathcal{O},\Phi) \approx 1$), the magnitude of ψ in (4) would be close to zero and therefore the change of ϕ would become very small. On the other hand, if $\Upsilon(\mathcal{O},\Phi) \approx 0.5$ (i.e., the likelihoods for the correct and the best wrong model about the same, then the magnitude of ψ would reach a maximum. Therefore, the training procedure as described in (4) will focus on input tokens which are likely to be misclassified but can be classified correctly after proper adjustment of the model parameters.

In order to determine $\partial d_c(\mathcal{O},\Phi)/\partial\phi$ in (4), we note that in the trended HMM, each mixture of each state is characterized by a multivariate time-varying Gaussian density function in the form of

$$b_{i,m}(\mathcal{O}_t|\tau_i) = \frac{(2\pi)^{-n/2}}{|\Sigma_i|^{1/2}} \cdot \exp\left(\frac{-1}{2}\left[\mathcal{O}_t - \sum_{p=0}^{P} B_{i,m}(p)(t-\tau_i)^p\right]^{Tr} 1 \cdot \Sigma_{i,m}^{-1}\left[\mathcal{O}_t - \sum_{p=0}^{P} B_{i,m}(p)(t-r_i)^p\right]\right) \tag{5}$$

where $B_{i,m}(p)$ and $\Sigma_{i,m}$ denote the polynomial coefficients for the time-varying Gaussian mean and the covariance matrix associated with the mth mixture of ith state, respectively; $(t-\tau_i)$ is the sojourn time[1] in state i at time t, and n is the dimensionality of the observation vector $\mathcal{O}_t$. Superscripts Tr and -1, and the symbol $|.|$ denote matrix transposition, inversion, and determinant, respectively. Based on the trended HMM for speech class j, the optimal state sequence $\Theta^j = \theta_1^j, \theta_2^j, \cdots, \theta_T^j$ and the corresponding mixture sequence $\mathcal{M}^j = m_1^j, m_2^j, \cdots, m_T^j$ for an input token $\mathcal{O} = \mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_T$ (T frames in total) is obtained by means of the Viterbi algorithm, with modification by incorporating an additional optimization loop for the state sojourn time (see Appendix for details). Then, the log-likelihood is given by

$$g_j(\mathcal{O},\Phi) = \sum_{t=1}^{T} \log b_{\theta_t^j, m_t^j}(\mathcal{O}_t|\tau_{\theta_t^j}) \tag{6}$$

which will be used to compute the gradient $\partial d_c(\mathcal{O},\Phi)/\partial\phi$ in (4) for model parameters in the mixture trended HMM to be described in later portions of this section.

B. Initializing Model Parameters

Once all the state boundaries and the optimal mixture components along the optimal state sequence are determined via the modified Viterbi segmentation step (see Appendix for detail), determining the time-varying mean parameters in the trended HMM reduces essentially to the problem of polynomial regression according to the ML method, which we adopt for model initialization. Here, we present the general solution for the regression problem involving multiple observation tokens where each token can be a subsequence of a training utterance that has been segmented and assigned to a given state. In the remainder of this subsection, class index j will be omitted since in-class information is used in the ML method and hence each class' model is built independently of another.

Let $\mathcal{O} = \{\mathcal{O}^1, \mathcal{O}^2, \cdots, \mathcal{O}^L\}$ denote a set of L feature vector sequences (i.e., a total of L variable-length tokens), and let $\mathcal{O}^l = \{\mathcal{O}_1^l, \mathcal{O}_2^l, \cdots, \mathcal{O}_{T^l}^l\}$ denote the lth sequence which has a total of T^l frames in length. Define

$$\mathcal{X}_t(i) = [(t-\tau_i)^0 \quad (t-\tau_i)^1 \quad \cdots \quad (t-\tau_i)^P]^{Tr}$$

as a $(P+1)$-dimensional vector of explanatory variables with $(t-\tau_i)$ representing the sojourn time in state i. Then the ML estimate for the polynomial coefficients becomes the solution to the regression equation

$$\mathcal{U}_{i,m}[\hat{\mathcal{B}}_{i,m}(0) \quad \hat{\mathcal{B}}_{i,m}(1) \quad \cdots \quad \hat{\mathcal{B}}_{i,m}(P)]^{Tr} = \mathcal{V}_{i,m}$$

where $\mathcal{U}_{i,m}$ and $\mathcal{V}_{i,m}$ are computed according to

$$\mathcal{U}_{i,m} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T^l}\gamma_t(i,m)\mathcal{X}_t(i)[\mathcal{X}_t(i)]^{Tr}}{\sum_{l=1}^{L}\sum_{t=1}^{T^l}\gamma_t(i,m)}$$
$$\mathcal{V}_{i,m} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T^l}\gamma_t(i,m)\mathcal{X}_t(i)[\mathcal{O}_t^l]^{Tr}}{\sum_{l=1}^{L}\sum_{t=1}^{T^l}\gamma_t(i,m)}.$$

In the above equation, the quantity $\gamma_t(i,m)$ is set to one if the model stays in mixture m of state i, and is set to zero, otherwise. The covariance matrix is determined according to the equation shown at the bottom of the next page, and the

[1] In this work we have not used duration-normalized time because of its implementational complexity. In our earlier work we show advantages of duration normalization in performance for an ML-based system [19] but to rigorously (not heuristically as other groups have done) carry out duration normalization for the current MCE-based system requires substantial new efforts that we have not taken in the current work.

formula for mixture weight parameters is

$$\hat{w}_{i,m} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T^l}\gamma_t(i,m)}{\sum_{v=1}^{M}\sum_{l=1}^{L}\sum_{t=1}^{T^l}\gamma_t(i,v)}$$

for $i = 1, 2, \cdots, N$ and $m = 1, 2, \cdots, M$. The observation density assumes the following form in our current model implementation:

$$b_i(\mathcal{O}_t|\tau_i) = \max_{m=1,2,\cdots,M} w_{i,m} b_{i,m}(\mathcal{O}_t|\tau_i).$$

That is, only the most likely mixture component is chosen as the observation density for each HMM state.

C. Gradient Computation for the Mixture-Dependent Polynomials

By substituting (2), (5), and (6) in (4), the gradient calculation of the mth mixture of ith state parameter, $B_{i,m}^{(j)}(l)$, $l = 0, 1, \cdots, P$, for the jth model becomes

$$\begin{aligned}
\frac{\partial \Upsilon(\mathcal{O},\Phi)}{\partial B_{i,m}^{(j)}(l)} &= \psi \frac{\partial d_c(\mathcal{O},\Phi)}{\partial B_{i,m}^{(j)}(l)} \\
&= \psi \frac{\partial}{\partial B_{i,m}^{(j)}(l)}(-g_c(\mathcal{O},\Phi) + g_\chi(\mathcal{O},\Phi)) \\
&= \psi \frac{\partial}{\partial B_{i,m}^{(j)}(l)}\left(-\sum_{t=1}^{T}\log b_{\theta_t^c, m_t^c}(\mathcal{O}_t|\tau_{\theta_t^c})\right. \\
&\quad \left. + \sum_{t=1}^{T}\log b_{\theta_t^\chi, m_t^\chi}(\mathcal{O}_t|\tau_{\theta_t^\chi})\right) \\
&= \psi_j \sum_{t\in T_{i,m}(j)} \frac{\partial}{\partial B_{i,m}^{(j)}(l)}\left(-\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_{i,m}^{(j)}|\right. \\
&\quad - \frac{1}{2}\left[\mathcal{O}_t - \sum_{p=0}^{P} B_{i,m}^{(j)}(p)(t-\tau_i)^p\right]^{Tr} \Sigma_{i,m}^{(j)^{-1}} \\
&\quad \left. \cdot \left[\mathcal{O}_t - \sum_{p=0}^{P} B_{i,m}^{(j)}(p)(t-\tau_i)^p\right]\right) \\
&= \psi_j \sum_{t\in T_{i,m}(j)} \Sigma_{i,m}^{(j)^{-1}} \\
&\quad \cdot \left[\mathcal{O}_t - \sum_{p=0}^{P} B_{i,m}^{(j)}(p)(t-\tau_i)^p\right](t-\tau_i)^l \qquad (7)
\end{aligned}$$

where the variable ψ_j is defined as

$$\psi_j = \begin{cases} \psi, & \text{if } j = c \text{ (correct-class)} \\ -\psi, & \text{if } j = \chi \text{ (closest-competing-class)} \\ 0, & \text{otherwise} \end{cases}$$

and the set $T_{i,m}(j)$ includes all the time indices such that mixture m and state i are in the optimal Viterbi path determined using the j-class model; that is,

$$T_{i,m}(j) = \{t|\theta_t^j = i, m_t^j = m\} \qquad 1 \le i \le N, \quad 1 \le m \le M.$$

D. Gradient Computation for the State and Mixture-Dependent Variances

Similarly, the gradient formula for covariance matrices can be derived (cf. [16]), which has the following final form:

$$\begin{aligned}
\frac{\partial \Upsilon(\mathcal{O},\Phi)}{\partial \tilde{\Sigma}_{i,m}^{(j)}} &= 0.5\psi_j \sum_{t\in T_{i,m}(j)} \\
&\quad \cdot \left[\mathcal{O}_t - \sum_{p=0}^{P} B_{i,m}^{(j)}(p)(t-\tau_i)^p\right] \\
&\quad \cdot \left[\mathcal{O}_t - \sum_{p=0}^{P} B_{i,m}^{(j)}(p)(t-\tau_i)^p\right]^{Tr} \Sigma_{i,m}^{(j)^{-1}} - I
\end{aligned} \qquad (8)$$

where I indicates the $n \times n$ unity matrix and $\tilde{\Sigma}_{i,m}$ is the log-transformed diagonal covariance matrices to automatically impose the constraint that the variances always remain positive definite during training.

IV. Experiments on Fitting Models to Speech Data

The problem of speech classification can be viewed as a statistical data-fitting problem, where relative closeness in fitting an array of speech models to the unknown speech data sequence provides the classification decision. In order to provide insights into the advantages of the MCE training on the trended HMM, we, in this section, report results of data-fitting experiments where both the conventional HMM and the trended HMM, trained with ML and with MCE, respectively, are used to fit the acoustic observation data.

The procedure and the criterion for the data-fitting experiments are discussed here first. Once the structure of the trended HMM is determined, the ML and MCE training algorithms discussed in Section III are used to estimate the trended HMM model parameters using a given set of training data. After the parameters are estimated, diagnostic analysis is carried out to examine the residuals measuring closeness of

$$\hat{\Sigma}_{i,m} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T^l}\gamma_t(i,m)\left[\mathcal{O}_t - \sum_{p=0}^{P} B_{i,m}(p)(t-\tau_i)^p\right]^{Tr}\left[\mathcal{O}_t - \sum_{p=0}^{P} B_{i,m}(p)(t-\tau_i)^p\right]}{\sum_{l=1}^{L}\sum_{t=1}^{T^l}\gamma_t(i,m)}$$

the model fitting to the data. To do this, the modified Viterbi algorithm as described in Appendix is used first to find the optimal state sequence $(\Theta = \theta_1, \theta_2, \cdots, \theta_T)$ associated with the given speech data, from which the model fitting error is then computed according to the data-fitting criterion described below.

Given the parameters of the mixture trended HMM, the model-generated observation sequence O_t is given by

$$O_t = \sum_{p=0}^{P} B_{i,m}(p)(t-\tau_i)^p + R_t(\Sigma_{i,m}) \tag{9}$$

where state i at given time t is determined by the state sequence Θ, and $\tau_i, i = 1, 2, \cdots, N$ are the Viterbi-segmentation boundaries of states. By setting the fitting function

$$F_t(i,m) = \sum_{p=0}^{P} B_{i,m}(p)(t-\tau_i)^p,$$

$R_t(\Sigma_{i,m})$ can be computed according to

$$R_t(\Sigma_{i,m}) = O_t - F_t(i,m).$$

The overall model data-fitting error is then computed by a linear summation of the residual squares over the states and over the state-bound time frames; that is,

$$\text{Error} = \sum_{i=1}^{N} \min_{m=1,2,\cdots,M} \left\{ \sum_{t=\tau_{i-1}}^{\tau_i} R_t^2(\Sigma_{i,m}) \right\}. \tag{10}$$

The smaller this error is, the better we consider the data-fitting would be (zero error indicates perfect fitting).

The test data sequence from phone ***aa***, for which we show the data-fitting results in this section, is selected from a female speaker of dialect region one of the TIMIT speech corpus. The raw speech data is in the form of a digitally sampled signal at 16 kHz. The conventional mel-frequency cepstral coefficients (MFCC's) are computed with a frame rate of 10ms. For illustration purposes, we show the data-fitting results only for the second-order MFCC C_2. The C_2 contains acoustic information about summation of log energies of low- and high-frequency channels subtracting those of mid-frequency channels. Other orders of MFCC's give similar results which will not be plotted due to space limitation.

Figs. 1 and 2 show the results of fitting the acoustic data (C2) of ***aa*** using the "correct" aa-model and using the "wrong" ae-model, respectively. The top two subplots in each figure show the data-fitting results for the ML-trained stationary-state HMM (left) (polynomial order $P = 0$), and for the trended HMM (right) with a linear trend function (polynomial order $P = 1$), respectively. The bottom two subplots in each figure show the data-fitting results using the MCE-trained stationary-state HMM and the trended HMM, respectively. In all the plots, the solid lines are the speech data, $\mathcal{O}_t$, of the C2 sequence from the test token (i.e., not used in training the models). The vertical axis represents the magnitude of C2 and the horizontal time axis is expressed in terms of the frame number. Note that the C2 data sequence is far from stationary. (In analyzing statistical properties of

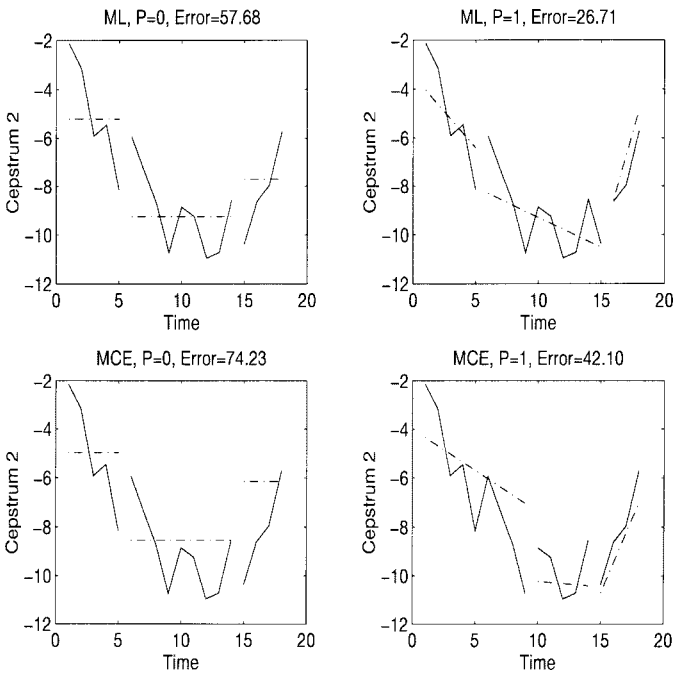

Fig. 1. Fitting three-state "correct" aa-models to an ***aa*** data C2 sequence.

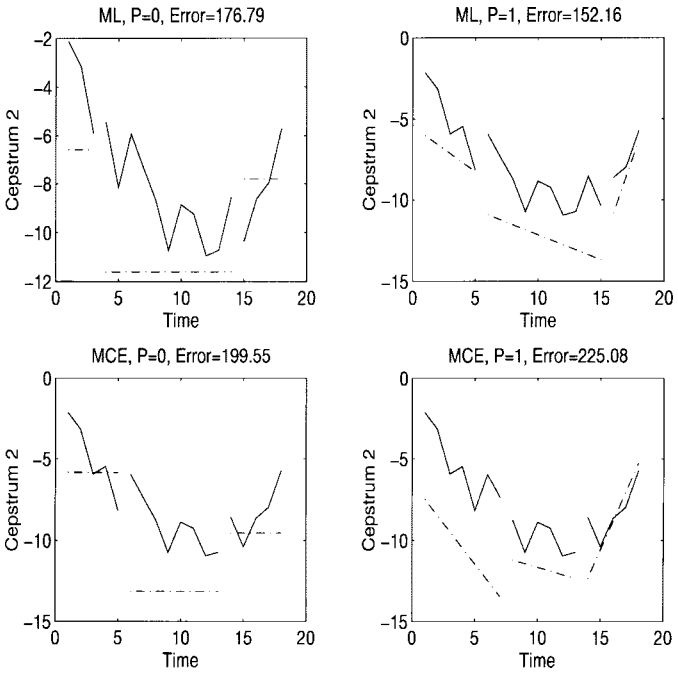

Fig. 2. Fitting three-state "wrong" ae-models to the ***aa*** data C2 sequence.

the MFCC data in TIMIT, we found that not only glides, liquids, diphthongs, and transition regions between phones reveal the most notable nonstationary nature in speech, but even vowels contain virtually no stationary portions. This has been consistent with the spectrographic studies of continuous speech [20].)

In all our data-fitting experiments, a left-to-right, three-state topology (with no skip) is used for both stationary-state HMM's and trended HMM's. Only one Gaussian is used per mixture for both types of the HMM's. For each subplot of Figs. 1 and 2, the two break-points in the otherwise continuous

solid lines correspond to the frames at which the optimal state transitions occur from state one to state two, and from state two to state three, respectively. The dash-dot lines in all subplots of Figs. 1 and 2 are the four different fitting functions, F_t, varying in the order ($P = 0$ or $P = 1$) of the trend function and in the training procedure (ML or MCE). These labels are shown at the head of each subplot, together with the data-fitting error computed according to (10).

Several observations are made in examining the example data-fitting results shown in Figs. 1 and 2. First, the MCE training consistently produces greater data-fitting errors than the ML counterparts. This is equivalent to reduced likelihoods from the ML to MCE training, and is consistent with the two respective training objectives. Second, for data fitting using the "correct" model (Fig. 1), the trended HMM's consistently gives lower fitting errors than the stationary-state HMM's, regardless of ML or MCE training. (This may not be true when using the "wrong" model for data fitting as shown in Fig. 2). Third, and importantly, despite the increased data-fitting errors in going from the ML to MCE training (for both cases of the "correct" and the "wrong" models), the *difference* between the data-fitting errors associated with use of the "correct" model and with use of the "wrong" model is much greater for the MCE training than for the ML training. That is, use of the "wrong" model produces greater errors in the data fitting than the "correct" model (this accounts for the differential likelihood scores necessary for identifying the correct speech class and discriminating against the wrong class), and this difference margin is significantly enhanced in going from the ML training to the MCE training.[2] More specifically, the enhancement of the error-difference margin discussed above is greater for the trended HMM than that for the stationary-state HMM. Examining the results of Figs. 1 and 2, in the case of MCE training, this enhancement of the error-difference margin for the trended HMM is $225.08 - 42.10 = 182.98$, significantly bigger than that for the stationary-state HMM ($199.55 - 74.23 = 125.32$). This difference shows a greater degree of freedom of the modeled trajectory space offered by the trended HMM, which should therefore endow the discriminative training with more power to distinguish the trajectories generated from different speech classes. The corresponding enhancement values for error-difference margin in the case of ML training is $152.16 - 26.71 = 125.71$ and $176.79 - 57.18 = 119.61$, for the trended HMM and the stationary-state HMM, respectively. The small difference between these two values (in comparison with that for the MCE-training case) suggests that with use of the ML training, the improvement of speech discrimination would be relatively slight in going from the stationary-state HMM to the trended HMM.

We should note at this point that although it is the temporally cumulative state likelihood (5) that determines the recognition score (rather than the data-fitting error measure [(10)], the difference between the two measures lies only in the variances which could weigh the frame errors differently with different variances. So long as the variances do not differ substantially (which we checked is the case for our data-fitting example shown in Figs. 1 and 2), the conclusions drawn from the above data-fitting results are valid. Nevertheless, the data-fitting results are able to illustrate a number of phenomena that cannot be shown by using the cumulative state likelihood. Specifically, the data-fitting results enable us to visualize detailed behaviors regarding how the trajectory from the model matches the actual data trajectory. Use of the cumulative state likelihood shows only the final score of the comparison between the modeled trajectory and the actual data trajectory, and does not show details of such a comparison which is important to understand and thereby to improve the model. Further, data-fitting results shown in Figs. 1 and 2 allow us to understand the underlying structure of the mixture trended HMM in terms of the important constraint that each linear trajectory is not allowed to jump across different mixture components within each state. This cannot be appreciated by using the cumulative state likelihood as the measure alone.

The above example, representative of many other TIMIT examples we have examined in this study, is only intended to illustrate the general, largely qualitative analysis of the discriminative mechanisms for both the stationary-state HMM and the trended HMM. The quantitative behavior in terms of relative effectiveness under varying modeling structures and varying training criteria can be assessed only in a large scale experiment, from which some meaningful statistics are extracted. The average classification error rate appears to be such a meaningful statistic which also has the advantage of being simple to compute and to illustrate. The experiments we have conducted to acquire such a statistic are described next.

V. Phonetic Classification Experiments

In this section we report the results from empirical studies, using the TIMIT data base, on the convergence property of the MCE training procedure described in Section III, and on phonetic classification performance achieved by applying this procedure. The TIMIT data base with a total of 462 different speakers is divided into a training set and a test set with no overlapping speakers. The training set consists of 442 speakers with a total 3536 sentences and the test set consists of 160 sentences spoken by the 20 remaining speakers. These speech materials contain a total of 129 743 phone tokens in the training set and 5775 phone tokens in the test set. In these data sets, only "sx" and "si" sentences were used. The experiments described in this section are aiming at classifying (i.e., using the phone segment information provided in the TIMIT data base) the 61 TIMIT labels defined in the TIMIT data base. In keeping with the convention adopted by many other speech recognition researchers, we folded 22 phone labels into the remaining 39 classes in determining classification accuracy.

The acoustic analysis used a 21-channel filterbank with approximates mel-spaced filters at a frame rate of 10 ms per

[2] We note here that although the MCE criterion does not directly aim at enhancing error differences between data fitting using the correct model and that using wrong models, this enhancement can be easily understood as a natural consequence of the MCE criterion. Such enhancement is highly desirable for robust speech recognition.

frame. Following is the analysis condition under which the static speech features are computed.

Sampling rate:	16 kHz
Frame size:	10 ms (160 samples)
Window type:	Hamming
Window length:	32 ms (512 samples)
Window overlap:	22 ms (352 samples)
Analysis:	Short-time spectrum analysis
Features:	Mel-frequency cepstrum coefficients (MFCC's)

For the computation of MFCC's, 21 triangular bandpass filters are simulated, spaced linearly from 0 to 1 kHz and exponentially from 1–8.86 kHz, with the adjacent filters overlapped in the frequency range by 50%. The fast Fourier transform (FFT) power spectral points are combined using a weighted sum to obtain the output of the triangular filter. The MFCC's (static features) are then computed according to (11)

$$\mathrm{MFCC}(p) = \sum_{r=1}^{21} S_r \cos\left(p \times [r - 0.5] \times \frac{\pi}{21}\right), \quad 0 \le p \le 12$$

where S_r is the log-energy output of the rth mel-filter. A twelve-component static feature vector is extracted every 10 ms throughout the signal. Thus the augmented feature vector is represented by a vector of 25 components, with 12 cepstrum coefficients, 12 delta cepstra[3] and the delta log energy. Each phone is a left-to-right (with only self and forward state transitions), three-state HMM with mixture Gaussian state observation densities (time invariant or time-varying). The covariance matrices in all the states of all the models are diagonal and are not tied. In the testing phase, the acoustic data of each test phone is scored with all phone models by applying the modified Viterbi algorithm, and the model with the highest likelihood score is treated as the recognized phone (i.e., ML decoding).

In Fig. 3, we show empirical results on the behavior of the MCE training procedure for the 39-phone context-independent phonetic classification task. (We first initialized the trainable parameters of the trended HMM's described in Section III-B before performing the MCE training.) Some fixed parameters of the models that we used to obtain the results of Fig. 3 are $N = 3$ as the number of HMM states for each phone, and $M = 5$ as the number of Gaussian mixtures in each HMM state. The upper graph of Fig. 3 shows the classification rates as a function of the epoch (a complete pass through the entire training data set is called an epoch) of the MCE training algorithm for the testing data. The solid lines are associated with MCE-trained conventional HMM ($P = 0$), and the dotted lines with trended HMM ($P = 1$). The lower graph of Fig. 3 shows the average loss for the entire training data set as a function of the training epoch. The convergence behavior of the MCE training which we expected from general theoretical considerations is confirmed by the results shown in Fig. 3; that is, the classification rate monotonically increases with the training epoch, and the average loss monotonically decreases, both reaching their respective asymptotic values after five epochs of the training. Note that the decreasing values of the average loss with the training epoch follow the same tendency, in a qualitative manner, as those of the classification error rate. The average loss decreases faster for the trended HMM than for the conventional HMM, indicating the effectiveness of the newly trained trended HMM. Similar characteristics in the classification performance are also observed. This indicates that the original objective set out for minimizing the classification error via the MCE training is accomplished and that the MCE training may be more effective for the trended HMM than the conventional HMM. In the remaining of this section we will report full detail of the phonetic classification results, focusing on the comparative performances of the MCE-trained trended HMM versus the conventional HMM.

For the MCE approach, the initial trended HMM's are obtained using the ML objective criterion with five iterations of the modified Viterbi algorithm as described in Section III-B and in the Appendix. The polynomial coefficients and diagonal covariances of the trended HMM's are further trained employing the MCE optimization procedure. A total of five epochs are performed and only the best-incorrect-class is used in the misclassification measure. Further, for both the stationary-state HMM and the trended HMM, we have explored both context-independent (CI) and context-dependent (CD) versions of the phonetic model. For the CI model, a total of 39 models ($39 \times 3 = 117$ states) are constructed, one for each of the 39 classes intended for the classification task. A CD phonetic model that we used in this study is the one that is made dependent on both the left and the right neighboring phone classes. The phone classes used are the same as those described in our earlier work [16], which result in a total of 1209 states for the folded 39 CD phones in TIMIT.

Several sets of experiments are run to evaluate the phonetic classifiers constructed using two types of HMM's (stationary-state and trended) and two types of training (ML and MCE). The overall performance of the phonetic classifiers, organized as the classification rate as a function of the polynomial trend function order ($P = 0$ for stationary-state HMM's and $P = 1$ for linearly trended HMM's) and of the mixture number ($M = 1$ or $M = 5$) in each HMM state, is summarized in Table I for the case of ML training, and in Table II for the

[3]Use of delta parameters in our trajectory model is motivated by our earlier empirical finding, based on empirical comparisons between stationary-state HMM's with delta parameters and linearly trended HMM's without delta parameters, that delta parameters and trajectory modeling partially complement each other in capturing true dynamic properties in speech data sequences [3]. Mixed use of delta parameters and trajectory modeling is admittedly heuristic, as we proceed in this study, but it nevertheless jointly utilizes different ways in which the two separate approaches exploit the dynamic properties of the speech data. Technically, the difference between use of delta parameters in the stationary-state HMM setup and use of the linear trended HMM without delta parameters can be seen as follow: the former takes fixed temporal differences of static MFCC's in the preprocessor (and finally mixing the results back with the static MFCC's in calculating likelihoods), while the latter trains the model parameters related to the dynamics (i.e. polynomial coefficients or slopes in the linear case) specific for each HMM state and for each speech class.

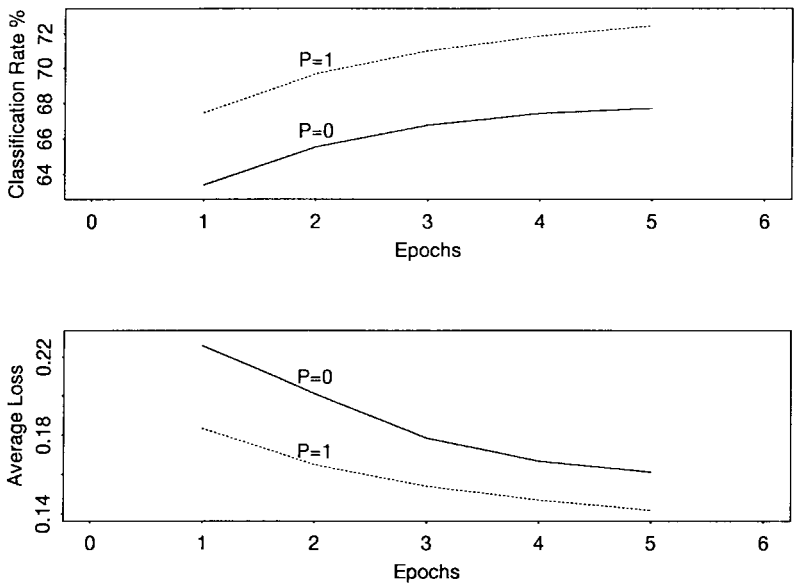

Fig. 3. Convergence characteristics of the MCE training procedure. Top graph shows the context-independent phonetic classification rates for the test set; bottom graph shows the average loss as a function of the training epoch.

TABLE I
TIMIT 39-PHONE CONTEXT INDEPENDENT (LEFT) AND CONTEXT DEPENDENT (RIGHT) CLASSIFICATION RATES USING THE ML TRAINING

Type of Model	Context-Independent Classification Rate		Context-Dependent Classification Rate	
	1 Mixture	5 Mixtures	1 Mixture	5 Mixtures
Stationary-State HMM (P=0)	53.07%	57.76%	76.62%	78.60%
Linearly Trended HMM (P=1)	54.11%	59.43%	77.07%	78.79%

TABLE II
TIMIT 39-PHONE CONTEXT INDEPENDENT (LEFT) AND CONTEXT DEPENDENT (RIGHT) CLASSIFICATION RATES USING THE MCE TRAINING

Type of Model	Context-Independent Classification Rate		Context-Dependent Classification Rate	
	1 Mixture	5 Mixtures	1 Mixture	5 Mixtures
Stationary-State HMM (P=0)	63.98%	67.72%	79.08%	80.19%
Linearly Trended HMM (P=1)	69.33%	72.42%	82.89%	83.48%

case of MCE training, respectively. Both CI and CD results are shown. The results shown in Tables I and II can be elaborated as follows. First, under all conditions, the MCE training is superior to the ML training; the MCE-based classifier achieves an average of 25% classification error rate reduction, uniformly across all types of speech models (both CI and CD ones, both stationary-state and trended HMM's), over the ML-based classifier. Second, for the ML-based classifier (Table II), the trended HMM is slightly superior to the stationary-state HMM, consistent with our earlier finding based on a different evaluation task [4]. Third, for the MCE-based classifier (Table II), superiority of the trended HMM over the stationary-state HMM becomes significantly greater than the ML case; this is true even at a better baseline performance level, and true for both the CI and CD models. Finally, the improvement in the classification rate in going from the ML to the MCE training with use of the trended HMM is higher than that with the stationary-state HMM. This shows that the behavior exhibited in Figs. 1 and 2 in our data-fitting experiments is a dominant one, testifying to our conjecture that the MCE training should be particularly effective for the trended HMM because of the new constrained degree of freedom existing in the modeled speech data sequence to allow for trajectory discrimination.

We conclude from the above phonetic classification results that the difference in performance between the stationary-state HMM and the trended HMM becomes more significant when MCE training is used than when ML training is used. The best result is achieved by using a combination of the trended HMM and the MCE training algorithm, which produces an error rate reduction from 25–33% in moving from the ML training to the MCE training.

VI. Summary and Discussion

In this study, the MCE training algorithm using gradient descent is derived, implemented and evaluated for optimally estimating the state-dependent polynomial coefficients in the trended HMM. Development of this new training approach is motivated by our recognition of the poor discriminative ability of the conventional ML training paradigm, particularly in view of the additional constrained degree of freedom in modeling speech data trajectories offered by the trended HMM, which we developed in the past. This degree of freedom is more limited in the conventional stationary-state HMM (which models piecewise constant "trajectories" rather than continuous trajectories as exhibited in most real speech data), and hence we infer that the discriminative training should be more powerful when applied to the trended HMM than to the stationary-state HMM which has already been demonstrated with some degrees of success by other research groups [1], [12], [10].

Our expectation for superiority of the MCE-trained trended HMM has been confirmed, as reported in Sections IV and V in this paper, both by data-fitting experiments and by phonetic classification experiments. We have observed consistently from the data-fitting experiments that use of a "wrong" model to fit test speech utterances generally produces greater data-fitting errors than the errors with use of the "correct" model, and that such error differentials ("wrong" model versus "correct" model) are the greatest with the MCE-trained trended HMM, followed by the MCE-trained conventional HMM, then by the ML-trained trended HMM, then by the ML-trained conventional HMM. These observations have been corroborated by the independent set of conclusions drawn from the phonetic classification experiments. The results summarized in Tables I and II demonstrate the best classification performance achieved with use of the MCE-trained trended HMM (classification rate of 83.48%), followed by the MCE-trained conventional HMM (classification rate of 80.19%), then by the ML-trained trended HMM, then by the ML-trained conventional HMM[4].

The results we have reported in this paper are promising, but at first glance may be striking in light of the opposite behavior in performance improvement from ML training to MCE training observed in comparing the following two scenarios of increasing model parameters: adding more mixtures versus adding linear trends.[5] For the former, the gain of MCE training is reduced moving from one Gaussian to five Gaussians per mixture (79.08%–76.62% versus 80.19%–78.60%), while for the latter, the gain is enhanced (79.08%–76.62% versus 82.89% –77.07%) (cf. Tables I and II). It is easy to explain the mixture case. With the conventional HMM using one Gaussian per mixture for speaker-independent data (such as TIMIT reported in this paper), the model does not have enough degrees of freedom in representing the true data distribution. Therefore, ML training with this highly limited model is very poor in finding good decision boundaries, where the MCE training becomes comparatively more powerful. When the degree of freedom is increased as more Gaussians per mixtures are added, the problems with the ML training are subdued. The MCE training in this case will still perform better but not as much in comparison with the one-Gaussian-per-mixture case.

The behavior opposite to the above in performance improvement from ML to MCE with use of the trended HMM (which, like adding more Gaussians per mixture, also increases model parameters), however, requires more careful explanations. It is clear that although the trended HMM and the mixture HMM both increase model parameters, the manner in which they increase the degree of freedom in the model space is completely different. We conjecture that the greater (rather than reduced) difference in performance between ML and MCE trainings with use of the trended HMM is attributed to the interactions between the increased model space due to the new parameters in the trend functions and the constraint in the model which forces the model to produce a smoothed trajectory within each HMM state. (The constraint is implemented in the modified Viterbi algorithm which forbids the trend function from jumping across different mixture components within each state.) This constraint balances the increased degree of freedom due to addition of the regression parameters, and allows the MCE training to work more effectively. Therefore, it is not just the size of model parameters that matters the most. Rather, it is how these parameters are structured and constrained which determines the relative effectiveness of various training algorithms in the recognition performance achievable by the models.

In summary, the work presented in this paper has provided evidence for the superior performance of the trended HMM, as a parametric stochastic trajectory model for speech acoustics, to the conventional HMM when the parameters characterizing the modeled trajectories are trained discriminatively. Mechanisms for such superiority are investigated through data-fitting experiments, which shed light on the role of speech trajectory discrimination in speech recognition. Because the trajectory model for speech acoustics as studied in this work is only a primitive and highly simplified model intended to describe the hierarchically structured dynamic process in speech production, we conclude that the discriminative analysis and learning will also play significant roles in more advanced and realistic dynamic models of speech production for use in speech recognition. In this context, the role of discriminative learning may be identified as one of the two critical components which shape and define the goal of speech production in a recently proposed phonetic theory of speech production [11], [14].

[4] At this point, we should point out that quantitative comparisons of our results reported on Section V with other published results on the TIMIT task are inherently difficult because different authors tend to use different test sets and different HMM setups. The performance of the benchmark ML-trained HMM for the CI task we reported (57.76% phone classification rate) is not far from other similar classifiers; for example, a somewhat higher rate on the same task (62.3%) is reported in [18], which uses 32 Gaussians per mixture in the HMM (we used five Gaussians per mixture only), and 39-dimensional feature vectors (we used 25-dimensional vectors), and tested on 112 male speakers (we tested on 20 mixed female and male speakers).

[5] The authors thank Dr. E. McDermott of ATR, Japan who pointed out this sharp observation to us and offered valuable discussions.

$$\delta_{t+1}(j,m,d)=\begin{cases}\max\limits_{i<j}\ \max\limits_{1\le v\le M}\ \max\limits_{0\le\tau\le t-1}\ \delta_t(i,v,\tau)a_{i,j}w_{j,m}b_{j,m}(O_{t+1}|d) & d=0\\ \delta_t(j,m,d-1)a_{j,j}b_{j,m}(O_{t+1}|d) & d>0,\end{cases}$$

$$\psi_{t+1}(j,m,d)=\begin{cases}\arg\ \max\limits_{i<j}\ \max\limits_{1\le v\le M}\ \max\limits_{0\le\tau\le t-1}\ \delta_t(i,v,\tau)a_{i,j} & d=0\\ (j,m,d-1) & d>0\end{cases}$$

APPENDIX
MODIFIED VITERBI SEGMENTATION ALGORITHM

This appendix describes the modified Viterbi segmentation algorithm that we developed in this study for automatic training of the parameters, notably the time-varying polynomial coefficients, in the trended HMM. Let $\Theta=\theta_1,\theta_2,\cdots,\theta_T$ be the state sequence and $\mathcal{O}=\{\mathcal{O}_1,\mathcal{O}_2,\cdots,\mathcal{O}_T\}$ be the given vector-valued observation sequence of length T with dimension n. Define a duration sequence $d_1,d_2,\cdots,d_T$ where d_t indicates the sojurn time in state θ_t (the time spent in the current state θ_t since the last state transition). Then the largest probability along a single state-sequence path up to time t, with duration d at state j can be expressed as

$$\delta_t(j,m,d)=\max_{\theta_1,\theta_2,\cdots,\theta_{t-1}}\mathrm{P}\{\theta_1,\theta_2,\cdots,\theta_t=j,\ d_t=d,\mathcal{O}_1,\mathcal{O}_2,\cdots,\mathcal{O}_t|\Phi\}$$

where Φ is the parameter ensemble of the model.

Given the above notations and definitions, the following four operations are a complete description of the segmentation step, where $\delta_t(j,m,d)$ is efficiently computed via recursion, and $\psi_t(j,m,d)$ is used to store the most likely state and mixture information at time $t-1$, given that $\theta_t=j,m_t=m$ and $d_t=d$.

1) Initialization:

$$\delta_1(j,m,d)=\begin{cases}\pi_j w_{j,m}b_{j,m}(\mathcal{O}_1|d), & d=0,1\le j\le N,\\ & 1\le m\le M\\ 0, & \text{otherwise}\end{cases}$$

$$\psi_1(j,d)=(0,0)d=0,1\le j\le N,1\le m\le M$$

with π_j being the initial probability distribution of Markov states.

2) Forward Recursion: See the formula at the top of the page, for $1\le j\le N,1\le m\le M$ and $1\le t<T$.

3) Termination of Recursion:

$$P^*=\max_i\max_m\max_{d_T}[\delta_T(i,m,d_T)]$$

$$(\theta_T^*,m_T^*,d_T^*)=\arg\max_i\max_m\max_{d_T}[\delta_T(i,m,d_T)]$$

4) Backtracking for Optimal Path:

$$(\theta_t^*,m_t^*,d_t^*)=\psi_{t+1}(\theta_{t+1}^*,m_{t+1}^*,d_{t+1}^*),\quad t=T-1,T-2,\cdots,1.$$

ACKNOWLEDGMENT

The authors thank Dr. E. McDermott of ATR, Japan, for valuable discussions, and thank two anonymous reviewers for their constructive suggestions on improving the technical content and presentation of this paper. A preliminary version of this work was presented in [17].

REFERENCES

[1] W. Chou, C. Lee, B. Juang, and F. Soong, "A minimum error rate pattern recognition approach to speech recognition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 8, pp. 5–31, 1994.

[2] L. Deng, "A generalized hidden Markov model with state-conditioned trend functions of time for the speech signal," *Signal Process.*, vol. 27, pp. 65–78, 1992.

[3] L. Deng and M. Aksmanovic, "Speaker-independent phonetic classification using hidden Markov models with state-conditioned mixtures of trend functions," *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 319–324, July 1997.

[4] L. Deng, M. Aksmanovic, D. Sun, and C. F. J. Wu, "Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states," *IEEE Trans. Speech Audio Processing*, vol. 2, no. 4, pp. 507–520, 1994.

[5] L. Deng, M. Lennig, and P. Mermelstein, "Modeling microsegments of stop consonants in a hidden Markov model based word recognizer," *J. Acoust. Soc. Amer.*, vol. 87, pp. 2738–2747, June 1990.

[6] T. Fukada, Y. Sagisaka, and K. Paliwal, "Model parameter estimation for mixture density polynomial segment models," in *Proc. ICASSP*, 1997, vol. 2, pp. 1403–1406.

[7] H. Gish and K. Ng, "Parametric trajectory models for speech recognition," in *Proc. ICSLP*, 1996, vol. 1, pp. 466–469.

[8] W. Holmes and M. Russell, "Linear trajectory segmental HMM's," *IEEE Signal Processing Lett.*, vol. 4, pp. 72–74, 1997.

[9] B. Juang and S. Katagiri, "Discriminative learning for minimum error rate training," *IEEE Trans. Signal Processing*, vol. 40, pp. 3043–3054, 1992.

[10] S. Kapadia, V. Valtchev, and S. J. Young, "MMI training for continuous phoneme recognition on the TIMIT database," in *IEEE Proc. ICASSP*, 1993, vol. 2, pp. 491–494.

[11] B. Lindblom, "Explaining phonetic variation: A sketch of the H and H theory," in *Speech Production and Speech Modeling*, W. Hardcastle and A. Maral, Eds. Boston, MA: Kluwer, 1990, pp. 403–439.

[12] E. McDermott and S. Katagiri, "Prototype-based minimum classification error/generalized probabilistic descent training for various speech units," *Comput. Speech Lang.*, vol. 8, pp. 351–368, 1994.

[13] M. Ostendorf, "From HMM's to segment models," in *Automatic Speech and Speaker Recognition—Advanced Topics*, C. Lee, F. Soong, and K. Paliwal, Eds. Boston, MA: Kluwer, 1996, pp. 185–210.

[14] J. Perkell, M. Matthies, M. Svirsky, and M. Jordan, "Goal-based speech motor control: A theoretical framework and some preliminary data," *J. Phonet.*, vol. 23, pp. 23–35, 1995.

[15] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–285, 1989.

[16] C. Rathinavelu and L. Deng, "Use of generalized dynamic feature parameters for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 232–242, 1997.

[17] ———, "The trended HMM with discriminative training for phonetic classification," in *Proc. ICSLP*, vol. 2, pp. 1049–1052, 1996.

[18] S. Sandhu, O. Ghitza, and C. H. Lee " A comparative study of mel-cepstra and EIH for phone classification under adverse conditions," in *IEEE Proc. ICASSP*, 1995, pp. 409–412.

[19] D. Sun, L. Deng, and C. F. J. Wu, "State-dependent time warping in the trended hidden Markov model," *Signal Process.*, vol. 39, pp. 263–275, 1994.
[20] V. Zue, "Speech spectrogram reading: An acoustic study of American English," Lecture Notes, Mass. Inst. Technol., Cambridge, Aug. 1991.

Rathinavelu Chengalvarayan (S'92–M'96) was born in Kadambathur, India. He received the B.E. degree in electronics and communications and the M.E. degree in communication systems engineering from Anna University, Guindy, Chennai, India, and the M.S. and Ph.D. degrees in electrical and computer engineering from University of Waterloo, Waterloo, Ont., Canada, in 1992 and 1995, respectively. His Ph.D. dissertation involved research on the integrated design of preprocessing and modeling components of a speech recognition system.

From March 1986 to December 1990, he was a Deputy Engineer at Bharat Electronics, Bangalore, India, involved in a number of projects ranging from digital telephone switching systems to wireless equipments, with special emphasis on speech signal processing. He served as a Post-Doctoral Fellow at the University of Waterloo from January 1996 to August 1996. He is currently a Member of Technical Staff in the Speech Processing Group at Bell Laboratories, Lucent Technologies, Naperville, IL. His current research interests include speech trajectory modeling, robust speech recognition in wireless environment, large vocabulary continuous speech recognition, speaker adaptation and verification, as well as model-based discriminative feature extraction. His recent research has focused on importing the speech recognition algorithms on advanced public and mobile switched telecommunication networks.

Dr. Chengalvarayan is an Associate Member of the Acoustical Society of America on speech communication and a member of European Speech Communication Association and Australian Speech Science and Technology Association on automatic speech recognition.

Li Deng (S'83–M'86–SM'91) received the B.S. degree in biophysics from University of Science and Technology of China in 1982, and the M.S. and Ph.D. degrees from the University of Wisconsin, Madison, both in electrical engineering, in 1984 and 1986, respectively.

He worked on large vocabulary automatic speech recognition at INRS-Telecommunications, Montreal, P.Q., Canada, from 1986 to 1989. Since 1989, he has been with Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ont., Canada, where he is currently Full Professor. From 1992 to 1993, he conducted sabbatical research at the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, working on statistical models of speech production and the related speech recognition algorithms. His research interests include acoustic-phonetic modeling of speech, speech recognition, synthesis, and enhancement, speech production and perception, statistical methods for signal analysis and modeling, nonlinear signal processing, neural network algorithms, computational phonetics and phonology for the world's languages, and auditory speech processing.