HMM-Based Speech Recognition Using State-Dependent, Discriminatively Derived Transforms on Mel-Warped DFT Features

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

***Abstract*— In the study reported in this paper, we investigate interactions of front-end feature extraction and back-end classification techniques in hidden Markov model-based (HMM-based) speech recognition. The proposed model focuses on dimensionality reduction of the mel-warped discrete fourier transform (DFT) feature space subject to maximal preservation of speech classification information, and aims at finding an optimal linear transformation on the mel-warped DFT according to the minimum classification error (MCE) criterion. This linear transformation, along with the HMM parameters, are automatically trained using the gradient descent method to minimize a measure of overall empirical error counts. A further generalization of the model allows integration of the discriminatively derived state-dependent transformation with the construction of dynamic feature parameters. Experimental results show that state-dependent transformation on mel-warped DFT features is superior in performance to the mel-frequency cepstral coefficients (MFCC's). An error rate reduction of 15% is obtained on a standard 39-class TIMIT phone classification task, in comparison with the conventional MCE-trained HMM using MFCC's that have not been subject to optimization during training.**

I. Introduction

THE STRUCTURE of many successful systems for speech recognition typically consists of a feature analysis-extraction procedure (i.e., signal preprocessing "front-end") followed by a "back-end" statistical pattern classifier. The operation of the back-end classifier has been virtually independent of the front-end feature extractor, although the performance of the recognizer has been known to be clearly affected by the information extracted from the input speech data and accessed by the classifier [2]. Previous studies showed that the signal processing and classification techniques interact with each other to affect phonetic classification (e.g., [16]). The recent advent of discriminative feature extraction showed that improved recognition results can be obtained by using an integrated optimization of both the preprocessing and classification stages [15]. Various techniques including use of filterbank, lifter, and dynamic feature design have been proposed for examining the interactions between the preprocessing stage and the classification stage [1], [7], [24]. In this paper, we report our recent comprehensive investigation on the integration of front-end preprocessing and back-end classification techniques in the context of model-based discriminative feature extraction, which has generalized all the previous techniques in a principled way. The techniques we developed have been evaluated on a phonetic classification task showing promising results.

Filterbank modeling of speech signals has been widely used in speech recognition tasks, and psychoacoustic studies have confirmed the importance of the critical band or mel-frequency scale in auditory functions [28]. Mel-filter bank (MFB) log channel energies are calculated directly from the discrete fourier transform (DFT), and effectively compress the linguistically relevant speech information contained within the raw DFT's. Because of the well-established psychoacoustic evidence for the mel-frequency scale and its (first-stage) data compression role, all the signal transformations developed in this study will be on sequences of MFB log channel energies, which we also call mel-warped DFT features because of the directness of computing log channel energies from DFT's.

The conventional, model-independent speech features, called *mel-frequency cepstral coefficients* (MFCC's), use discrete cosine transform (DCT) as a linear operator to map mel-warped DFT (in the form of MFB log channel energies) into a lower dimensional feature space [6], [13]. Despite the empirical superiority of MFCC's over many other types of signal processing techniques, there are no theoretical reasons why the linear transformation associated with DCT, which is fixed *a priori* and independent of HMM states and of speech classes, on MFB log channel energies is an optimal one as far as the speech recognition performance is concerned. To construct theoretically optimal transformation, we have in this study developed a new statistical model of speech, called optimum-transformed HMM (THMM), with the optimality of the transformation defined according to the MCE criterion. The state-dependent transformation on the mel-warped DFT, together with the HMM parameters, is automatically trained using the gradient descent method, resulting in minimization of a measure of an overall empirical error count.

Manuscript received March 25, 1996; revised October 14, 1996. The work of C. Rathinavelu was supported by a Commonwealth Scholarship. 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 Dr. Mazin Rahim.

R. Chengalvarayan was with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ont., Canada N2L 3G1. He is now with the Speech Processing Group, Bell Laboratories, Lucent Technologies, Naperville, IL 60566 USA.

L. Deng is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ont., Canada N2L 3G1 (e-mail: deng@crg5.waterloo.edu).

Publisher Item Identifier S 1063-6676(97)03184-2.

Two versions of the optimum transformed HMM have been developed in this work. Version 1, which we call THMM-1, performs state-dependent linear transformation on the static mel-warped DFT, independently of the successive frames (unless, of course, there occurs a state transition among the successive frames). In contrast, Version 2 of the THMM, which we call THMM-2, performs the transformation jointly on the static and dynamic parameters based on mel-warped DFT's. To keep the generality of our approach, we constructed the THMM-2 with generalized dynamic parameters; i.e., linear filtering of static parameters with filter weights made as state dependent and trainable [7], [23].

The remainder of the paper is organized as follows. Section II describes the mathematical formulation of both versions of the THMM, and shows the construction of input discriminative feature sets closely tied to the speech model. In Section III, we develop a discriminative training algorithm for both THMM-1 and THMM-2. In particular, the gradient calculation for the newly introduced model parameters, which is a critical step in the training algorithm, is presented in detail. In Section IV, we present experimental results and report the comparative performance of the THMM with benchmark systems in a standard TIMIT 39-class phonetic classification task. We summarize our findings in Section V, and in Figs. 6–9, we provide details of phonetic confusion matrices obtained in our experiments.

II. A Statistical Model of Speech Embedding Input Features

Let $\mathcal{F} = \{\mathcal{F}^1, \mathcal{F}^2, \ldots, \mathcal{F}^L\}$ denote a set of L mel-filterbank log-energy (mel-warped DFT) n-dimensional vector-valued sequences, and let $\mathcal{F}^l = \{\mathcal{F}^l_1, \mathcal{F}^l_2, \cdots, \mathcal{F}^l_{T^l}\}$ denote the lth sequence having a length of T^l frames.

A. Construction of State-Dependent Transforms for Static Input Features

The THMM-1 described in this paper integrates the input features (mel-warped DFT's or MFB log channel energies) into the modeling process using a set of state-dependent transformation matrices as trainable parameters of the model. The new, transformed static feature vector $\mathcal{X}^l_t$ at time frame t (lth token) is a state (i) dependent, mixture (m) dependent, linear combination of each row of transformation matrix with each element of the MFB log channel energy vector at time t according to

$$\mathcal{X}^l_{p,t} = \sum_{q=1}^{n} \mathcal{B}_{p,q,i,m} \mathcal{F}^l_{q,t} \quad p = 1, 2, \cdots, d, \; t = 1, 2, \cdots, T^l \tag{1}$$

In the matrix form, (1) can be written as

$$\begin{pmatrix} \mathcal{X}^l_{1,t} \\ \mathcal{X}^l_{2,t} \\ \vdots \\ \mathcal{X}^l_{d,t} \end{pmatrix} = \begin{pmatrix} \mathcal{B}_{1,1,i,m} & \mathcal{B}_{1,2,i,m} & \cdots & \mathcal{B}_{1,n,i,m} \\ \mathcal{B}_{2,1,i,m} & \mathcal{B}_{2,2,i,m} & \cdots & \mathcal{B}_{2,n,i,m} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{B}_{d,1,i,m} & \mathcal{B}_{d,2,i,m} & \cdots & \mathcal{B}_{d,n,i,m} \end{pmatrix} \begin{pmatrix} \mathcal{F}^l_{1,t} \\ \mathcal{F}^l_{2,t} \\ \vdots \\ \mathcal{F}^l_{n,t} \end{pmatrix}$$

or

$$\mathcal{X}^l_t = \mathcal{B}_{i,m} \mathcal{F}^l_t$$

where $\mathcal{B}_{p,q,i,m}$ is the pqth element of the transformation matrix $\mathcal{B}_{i,m}$ associated with the mth mixture residing in the Markov state i, n is the number of MFB log channel energies for each frame, and d is the vector size of the transformed static feature. Note that the transformed static features constructed above can be interpreted as the output from a slowly time-varying (due to state dependence of the transformation) linear filter with the MFB log energy vector sequence as the input.

Given the transformed static features as described above, the dynamic feature vectors $\mathcal{Y}^l_t$ (for frame t of lth token) are constructed in a conventional way (i.e., independent of the HMM state and not jointly with the transformation on the MFB log channel energies) by taking the difference between two frame forward and two frame backward of the related static features according to [8]

$$\begin{aligned} \mathcal{Y}^l_t = \mathcal{X}^l_{t+2} - \mathcal{X}^l_{t-2} &= \mathcal{B}_{i,m} \mathcal{F}^l_{t+2} - \mathcal{B}_{i,m} \mathcal{F}^l_{t-2} \\ &= \mathcal{B}_{i,m} \left[\mathcal{F}^l_{t+2} - \mathcal{F}^l_{t-2} \right]. \end{aligned} \tag{2}$$

This fixed window length of four (equivalent to 40 ms) appears to have reasonably well captured the slope of the spectral envelope. Note that in THMM-1, the dynamic features at frame t are extracted by taking the linear transformation of time differenced MFB log channel energy vectors at time $t+2$ and at time $t-2$ using the transformation matrix $\mathcal{B}_{i,m}$ derived from the static feature optimization only.

B. Construction of State-Dependent Joint Transforms for Static and Dynamic Features

For THMM-2, state-dependent linear transforms for static features and those for generalized dynamic features [7], [23] are integrated in a single model to obtain the optimal combined advantages of individual sets of features. (The generalized dynamic parameter technique discussed here includes the conventional use of the dynamic parameters developed in [11] and [12] as special cases.) As described above, the static features are obtained by a linear transformation of an n-dimensional input space for the MFB log channel energies, represented by the vector $\mathcal{F}^l_t$, to a transformed d-dimensional feature space according to (1). Instead of taking the temporal difference of the transformed static features fixed *a priori* in THMM-1, the dynamic feature vector $\mathcal{Y}^l_t$ at frame t in THMM-2 is constructed as additional state-dependent, trainable linear combinations of the static features stretching over the interval f frames forward and b frames backward according to

$$\mathcal{Y}^l_t = \sum_{k=-b}^{f} w_{k,i,m} \mathcal{X}^l_{t+k}, \quad 1 \le l \le L, \; 1 \le t \le T^l \tag{3}$$

where $w_{k,i,m}$ is the kth scalar weighting coefficient associated with the mth mixture residing in the Markov state i. (Note that in this THMM-2, $w_{k,i,m}$ is trainable, in contrast to THMM-1 where weights are prefixed). In the matrix form, (3) can be

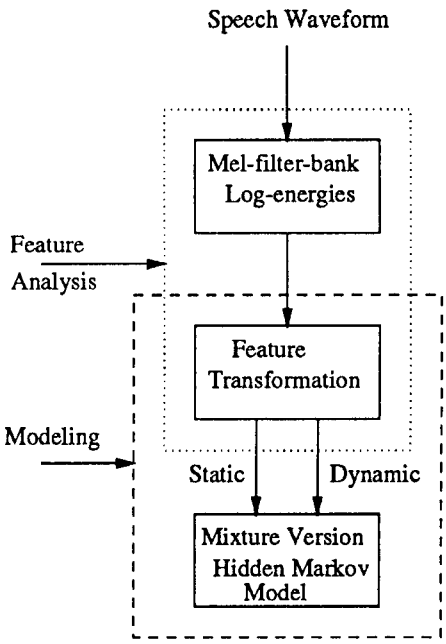

Fig. 1. Block diagram of showing the integration of feature analysis and recognizer design in Version 2 of the optimum-transformed HMM (THMM-2).

written as

$$\begin{pmatrix} \mathcal{Y}^l_{1,t} \\ \mathcal{Y}^l_{2,t} \\ \vdots \\ \mathcal{Y}^l_{d,t} \end{pmatrix} = \begin{pmatrix} \mathcal{X}^l_{1,t-b} & \cdots & \mathcal{X}^l_{1,t} & \cdots & \mathcal{X}^l_{1,t+f} \\ \mathcal{X}^l_{2,t-b} & \cdots & \mathcal{X}^l_{2,t} & \cdots & \mathcal{X}^l_{2,t+f} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{X}^l_{d,t-b} & \cdots & \mathcal{X}^l_{d,t} & \cdots & \mathcal{X}^l_{d,t+f} \end{pmatrix} \times \begin{pmatrix} w_{-b,i,m} \\ w_{-b+1,i,m} \\ \vdots \\ w_{f,i,m} \end{pmatrix}$$

where subscript $1, 2, \cdots, d$ denotes the individual element in the feature vector. The static feature matrix above has the dimensionality $d \times (f+b+1)$, with d being the dimension of the feature vectors. Using (1), we rewrite (3) as

$$\mathcal{Y}^l_t = \sum_{k=-b}^{f} w_{k,i,m} \mathcal{B}_{i,m} \mathcal{F}^l_{t+k} = \mathcal{B}_{i,m} \sum_{k=-b}^{f} w_{k,i,m} \mathcal{F}^l_{t+k}. \quad (4)$$

According to the definition of (3), the dynamic features can be interpreted as the output from a slowly and stepwise time-varying linear filter with the (optimally transformed) static feature vector sequence serving as the input to the filter. The time-varying filter coefficients are evolving slowly according to the Markov chain in the underlying HMM. In this THMM-2, the jointly transformed static and dynamic features are provided as data input into the modeling stage of the speech recognizer constructed as a mixture continuous density HMM. The THMM-2's integration of the feature analysis, as exemplified by the top two blocks, and the modeling process, as exemplified by the bottom two blocks (overlapping in the feature transformation block), is depicted in Fig. 1, where both static and dynamic features are subject to joint optimization which shares between feature analysis and model construction.

C. Output Distributions of THMM and Full Set of Model Parameters

A mixture Gaussian density associated with each state i (a total of N states) is used in the model, which assumes the form

$$b_i(\mathcal{O}^l_t) = b_i(\mathcal{X}^l_t, \mathcal{Y}^l_t) = \sum_{m=1}^{M} c_{i,m} b_{i,m}(\mathcal{X}^l_t) b_{i,m}(\mathcal{Y}^l_t), \quad 1 \le i \le N \quad (5)$$

where $\mathcal{O}^l_t$ is the augmented feature vector (including both static and dynamic features) of the lth token at frame t, M is the total number of Gaussian mixtures in the HMM's output distribution, and $c_{i,m}$ is the mixture weight for the mth mixture in state i. In (5), $b_{i,m}(\mathcal{X}_t)$, and $b_{i,m}(\mathcal{Y}_t)$ are d-dimensional unimodal Gaussian densities for static and dynamic features, respectively, as

$$b_{i,m}(\mathcal{X}^l_t) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_{x,i,m}|^{\frac{1}{2}}} \times \exp\left(\frac{-1}{2} [\mathcal{X}^l_t - \mu_{x,i,m}]^{\mathrm{Tr}} \Sigma^{-1}_{x,i,m} [\mathcal{X}^l_t - \mu_{x,i,m}] \right)$$

$$b_{i,m}(\mathcal{Y}^l_t) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_{y,i,m}|^{\frac{1}{2}}} \times \exp\left(\frac{-1}{2} [\mathcal{Y}^l_t - \mu_{y,i,m}]^{\mathrm{Tr}} \Sigma^{-1}_{y,i,m} [\mathcal{Y}^l_t - \mu_{y,i,m}] \right)$$

where variables $\mathcal{X}$ and $\mathcal{Y}$ indicate the static and the dynamic features, respectively. Superscripts Tr and -1 denote vector transposition and matrix inversion. The mixture weights $c_{i,m}$ in (5) satisfy the stochastic constraint

$$\sum_{m=1}^{M} c_{i,m} = 1, \quad 1 \le i \le N$$
$$c_{i,m} \ge 0, \quad 1 \le i \le N, \ 1 \le m \le M.$$

The classic or conventional HMM [17], THMM-1, and the HMM incorporating the generalized dynamic parameters described in [7] and [23], can all be considered as special cases of the THMM-2 presented here. The THMM-2 can be reduced to

- THMM-1, by removing state-dependent optimization of the dynamic features;
- the model of [7], by removing state-dependent optimization of the static features;
- the classic HMM, by removing state-dependent optimization on both static and dynamic features.

The full set of the parameters associated with the most general model THMM-2 are summarized as follows:

- transition probabilities $a_{i,j}$: $i, \ j = 1, 2, \cdots, N$ of the N-state Markov chain;
- The state-dependent mixture Gaussian mean vectors $\{\mu_{x,i,m}, \mu_{y,i,m}\}$;
- State and mixture dependent Covariance matrices $\{\Sigma_{x,i,m}, \Sigma_{y,i,m}\}$;

- Transformation matrices defining static features for each state i and for each mixture m $\mathcal{B}_{i,m},\ 1 \le i \le N,\ 1 \le m \le M$;
- Weighting coefficients defining dynamic feature parameters for each state i and for each mixture m $w_{k,i,m},\ 1 \le i \le N,\ 1 \le m \le M,\ -b \le k \le f$.

The subscripts x and y stand for the static and dynamic features, respectively.

III. Discriminative Parameter Estimation

Discriminative training or parameter estimation by the minimum classification error (MCE) criterion has been successfully used by several researchers in speech and speaker recognition tasks to improve upon the maximum likelihood (ML) criterion (e.g., [14], [18], [4], [19]). In-class information is used in the ML training and out-of-class information is used in the MCE-based training. In this section, we describe the application of the MCE-based training for the new model, THMM, developed in this study and detailed in Section II. The ML trained model [17] is used as the initial model for the ensuing MCE training step. In the supervised training mode, each training token $\mathcal{O}^l$ is known to belong to one of $\mathcal{K}$ speech classes $\{C^j\}_{j=1}^{\mathcal{K}}$. The speech recognition process is based on the classifier parameter set, $\Phi = \{\Phi^j\}_{j=1}^{\mathcal{K}}$, derived from the training process. The goal of the training is to reduce the number of misclassifications occurring over the training set through minimization of the overall loss function $\Upsilon(\mathcal{O}^l, \Phi)$, closely related to the classification error. In the THMM, the classifier parameter set consists of all the state-dependent, mixture-dependent transformation matrices $\mathcal{B}_{i,m}$ (both THMM-1 and THMM-2), weighting functions $w_{k,i,m}$ (only THMM-2), together with the conventional HMM parameters including mixture weights $c_{i,m}$, mixture Gaussian mean vectors $(\mu_{x,i,m}, \mu_{y,i,m})$, and mixture Gaussian covariance matrices $(\Sigma_{x,i,m}, \Sigma_{y,i,m})$, for all the models $(j = 1, 2, \cdots \mathcal{K})$ each representing a distinctive class of the speech sounds to be classified.

A. The MCE Training Procedure

The overall loss function is constructed and minimized through the following steps.

1) *Discriminant function*: The log-likelihood score of the input utterance $\mathcal{O}^l$ along the optimal state sequence $\Theta^\kappa = \{\theta_1^\kappa, \theta_2^\kappa \cdots, \theta_{T^l}^\kappa\}$ for the model associated with the κth class Φ^κ can be written as

$$g_\kappa(\mathcal{O}^l, \Phi) = \sum_{t=1}^{T^l} \log b_{\theta_t^\kappa}(\mathcal{O}_t^l)$$

where $b_{\theta_t^\kappa}(\mathcal{O}_t^l)$ is the probability of generating the feature vector $\mathcal{O}_t^l$ at time t in state θ_t^κ by the model for κth class. The implied decision rule for classification is defined as

$$C(\mathcal{O}^l) = C^\kappa, \quad \text{if } g_\kappa(\mathcal{O}^l, \Phi) = \max_j\ g_j(\mathcal{O}^l, \Phi).$$

2) *Misclassification measure:* Given a discriminant function, the misclassification measure for an input training utterance $\mathcal{O}^l$ from class κ becomes

$$\begin{aligned} d_\kappa(\mathcal{O}^l, \Phi) &= -g_\kappa(\mathcal{O}^l, \Phi) + \max_{j \ne \kappa} g_j(\mathcal{O}^l, \Phi) \\ &= -g_\kappa(\mathcal{O}^l, \Phi) + g_\chi(\mathcal{O}^l, \Phi) \end{aligned}$$

where C^χ is the most confusable class. Clearly, $d_\kappa(\mathcal{O}^l, \Phi) > 0$ implies misclassification and $d_\kappa(\mathcal{O}^l, \Phi) \le 0$ means correct classification.

3) *Loss function:* The loss function is defined as a sigmoid, nondecreasing function of d_κ

$$\Upsilon_\kappa(\mathcal{O}^l, \Phi) = \frac{1}{1 + e^{-d_\kappa(\mathcal{O}^l, \Phi)}}$$

which approximates the classification error count.

4) *Overall loss function:* The overall loss function for the entire classifier is defined for each class as

$$\Upsilon(\mathcal{O}^l, \Phi) = \sum_{\kappa=1}^{\mathcal{K}} \Upsilon_\kappa(\mathcal{O}^l, \Phi)\delta[\mathcal{O}^l \in C^\kappa] \tag{6}$$

where $\delta[\xi]$ is the Kronecker indicator function of a logic expression ξ that gives value 1 if the value of ξ is true and value 0, otherwise. The average loss (or error probability) for the entire training data set is defined as

$$\mathcal{L}(\Phi) = \frac{1}{L} \sum_{l=1}^{L} \Upsilon(\mathcal{O}^l, \Phi) \tag{7}$$

where L is the total number of training tokens.

5) *Minimization:* The loss function $\Upsilon(\mathcal{O}^l, \Phi)$ is minimized, each time a training token $\mathcal{O}^l$ is presented, by adaptively adjusting the parameter set Φ according to

$$\Phi_{l+1} = \Phi_l - \epsilon \nabla \Upsilon\big(\mathcal{O}^l, \Phi_l\big) \tag{8}$$

where Φ_l is the parameter set at the lth iteration, $\nabla\Upsilon(\mathcal{O}^l, \Phi_l)$ is the gradient of the loss function for training sample $\mathcal{O}^l$, and ϵ is a small positive learning constant.

B. Gradient Calculation

The THMM model parameters are adaptively adjusted to reduce the overall loss function along a gradient descent direction. The gradient is obtained by computing the partial derivatives of $\Upsilon(\mathcal{O}^l, \Phi)$ with respect to each THMM parameter for a given training token $\mathcal{O}^l$ belonging to class κ. For the sake of keeping our presentation simple, we describe the gradient calculation only for the newly introduced model parameters. Let $\phi_{i,m}^j$ denote a feature extraction parameter associated with model j, then in the case of token-by-token[1] training, we can write the gradient as

$$\begin{aligned} \frac{\partial \Upsilon(\mathcal{O}^l, \Phi)}{\partial \phi_{i,m}^j} &= \frac{\partial}{\partial \phi_{i,m}^j} \left(\sum_{\kappa'=1}^{\mathcal{K}} \Upsilon_{\kappa'}(\mathcal{O}^l, \Phi)\delta\big[\mathcal{O}^l \in C^{\kappa'}\big] \right) \\ &= \frac{\partial}{\partial \phi_{i,m}^j} \Upsilon_\kappa(\mathcal{O}^l, \Phi) \\ &= \frac{\partial \Upsilon_\kappa(\mathcal{O}^l, \Phi)}{\partial d_\kappa(\mathcal{O}^l, \Phi)} \frac{\partial d_\kappa(\mathcal{O}^l, \Phi)}{\partial g_j(\mathcal{O}^l, \Phi)} \frac{\partial g_j(\mathcal{O}^l, \Phi)}{\partial \phi_{i,m}^j}. \end{aligned} \tag{9}$$

[1] This type of sequential optimization in general is referred to as *stochastic descent* [10].

The first factor in the right-hand-side of (9) can be simplified to

$$\begin{aligned}\frac{\partial \Upsilon_\kappa(\mathcal{O}^l,\Phi)}{\partial d_\kappa(\mathcal{O}^l,\Phi)} &= \frac{\partial}{\partial d_\kappa(\mathcal{O}^l,\Phi)}\left(\frac{1}{1+e^{-d_\kappa(\mathcal{O}^l,\Phi)}}\right)\\ &= \frac{e^{-d_\kappa(\mathcal{O}^l,\Phi)}}{\left(1+e^{-d_\kappa(\mathcal{O}^l,\Phi)}\right)^2} = e^{-d_\kappa(\mathcal{O}^l,\Phi)}\Upsilon_\kappa^2(\mathcal{O}^l,\Phi)\\ &= \left\{\frac{1}{\Upsilon_\kappa(\mathcal{O}^l,\Phi)}-1\right\}\Upsilon_\kappa^2(\mathcal{O}^l,\Phi)\\ &= \Upsilon_\kappa(\mathcal{O}^l,\Phi)\big[1-\Upsilon_\kappa(\mathcal{O}^l,\Phi)\big]. \end{aligned} \tag{10}$$

The second factor of the right-hand-side of (9) can be simplified as follows:

$$\begin{aligned}\frac{\partial d_\kappa(\mathcal{O}^l,\Phi)}{\partial g_j(\mathcal{O}^l,\Phi)} &= \frac{\partial}{\partial g_\chi(\mathcal{O}^l,\Phi)}(-g_\kappa(\mathcal{O}^l,\Phi)+g_\psi(\mathcal{O}^l,\Phi))\\ &= \begin{cases}-1, & \text{if } j=\kappa\\ 1, & \text{if } j=\chi.\end{cases}\end{aligned} \tag{11}$$

The third factor of the right-hand-side of (9) can be modified to

$$\begin{aligned}&\frac{\partial g_j(\mathcal{O}^l,\Phi)}{\partial \phi_{i,m}^j}\\ &= \frac{\partial}{\partial \phi_{i,m}^j}\sum_{t=1}^{T^l}\log b_{\theta_t^j}(\mathcal{O}_t^l)\\ &= \sum_{t\in T_i^l}\frac{1}{b_i^j(\mathcal{O}_t^l)}\frac{\partial}{\partial \phi_{i,m}^j}\sum_{m'=1}^{M}c_{i,m'}^j b_{i,m'}^j(\mathcal{X}_t^l)b_{i,m'}^j(\mathcal{Y}_t^l)\\ &= \sum_{t\in T_i^l}\frac{c_{i,m}^j b_{i,m}^j(\mathcal{X}_t^l)b_{i,m}^j(\mathcal{Y}_t^l)}{b_i^j(\mathcal{O}_t^l)}\frac{\partial}{\partial \phi_{i,m}^j}\frac{-1}{2}\\ &\quad\times\big([\mathcal{X}_t^l-\mu_{x,i,m}^j]^{\mathrm{Tr}}\Sigma_{x,i,m}^{-1}(j)[\mathcal{X}_t^l-\mu_{x,i,m}^j]\\ &\quad+[\mathcal{Y}_t^l-\mu_{y,i,m}^j]^{\mathrm{Tr}}\Sigma_{y,i,m}^{-1}(j)[\mathcal{Y}_t^l-\mu_{y,i,m}^j]\big)\end{aligned} \tag{12}$$

where the set T_i^l includes all the time indices such that the state index of the state sequence at time t belongs to state i in the Markov chain, i.e.,

$$T_i^l=\{t\mid \theta_t=i\},\quad 1\le i\le N,\ 1\le t\le T^l.$$

Define the *a posteriori* probability as

$$\gamma_{i,m}^j(t)=\frac{c_{i,m}b_{i,m}(\mathcal{X}_t^l)b_{i,m}(\mathcal{Y}_t^l)}{b_i(\mathcal{O}_t^l)}.$$

Then, using (10)–(12), (9) can be rewritten as

$$\begin{aligned}\frac{\partial \Upsilon(\mathcal{O}^l,\Phi)}{\partial \phi_{i,m}^j} &= \psi_j\sum_{t\in T_i^l}\gamma_{i,m}^j(t)\frac{\partial}{\partial \phi_{i,m}^j}\frac{-1}{2}\big([\mathcal{X}_t^l-\mu_{x,i,m}^j]^{\mathrm{Tr}}\\ &\quad\times\Sigma_{x,i,m}^{-1}(j)[\mathcal{X}_t^l-\mu_{x,i,m}^j]+[\mathcal{Y}_t^l-\mu_{y,i,m}^j]^{\mathrm{Tr}}\\ &\quad\times\Sigma_{y,i,m}^{-1}(j)[\mathcal{Y}_t^l-\mu_{y,i,m}^j]\big)\end{aligned} \tag{13}$$

with the adaptive step size defined as

$$\psi_j=\begin{cases}\Upsilon_\kappa(\mathcal{O}^l,\Phi)\big[\Upsilon_\kappa(\mathcal{O}^l,\Phi)-1\big], & \text{if } j=\kappa\\ \Upsilon_\kappa(\mathcal{O}^l,\Phi)\big[1-\Upsilon_\kappa(\mathcal{O}^l,\Phi)\big], & \text{if } j=\chi.\end{cases}$$

In the remainder of this section, class index j will be omitted for clarity of presentation.

1) Gradient Computation of $\mathcal{B}_{i,m}$ for THMM-1: By substituting (1) and (2) into (13), the gradient calculation of $\mathcal{B}_{i,m}^j$ becomes

$$\begin{aligned}\frac{\partial \Upsilon(\mathcal{O}^l,\Phi)}{\partial \mathcal{B}_{i,m}} &= \psi\sum_{t\in T_i^l}\gamma_{i,m}(t)\frac{\partial}{\partial \mathcal{B}_{i,m}}-\frac{1}{2}\big([\mathcal{B}_{i,m}\mathcal{F}_t^l-\mu_{x,i,m}]^{\mathrm{Tr}}\\ &\quad\times\Sigma_{x,i,m}^{-1}[\mathcal{B}_{i,m}\mathcal{F}_t^l-\mu_{x,i,m}]\\ &\quad+[\mathcal{B}_{i,m}(\mathcal{F}_{t+2}^l-\mathcal{F}_{t-2}^l)-\mu_{y,i,m}]^{\mathrm{Tr}}\\ &\quad\times\Sigma_{y,i,m}^{-1}[\mathcal{B}_{i,m}(\mathcal{F}_{t+2}^l-\mathcal{F}_{t-2}^l)-\mu_{y,i,m}]\big)\\ &= -\psi\sum_{t\in T_i^l}\gamma_{i,m}(t)\big(\Sigma_{x,i,m}^{-1}[\mathcal{X}_t^l-\mu_{x,i,m}][\mathcal{F}_t^l]^{\mathrm{Tr}}\\ &\quad+\Sigma_{y,i,m}^{-1}[\mathcal{Y}_t^l-\mu_{y,i,m}][\mathcal{F}_{t+2}^l-\mathcal{F}_{t-2}^l]^{\mathrm{Tr}}\big).\end{aligned} \tag{14}$$

To reduce the computational complexity as well as the model complexity, we tied all the mixtures for feature transformation matrices $\mathcal{B}_{i,m}^j$ to a single state parameter $\mathcal{B}_i^j$ in our experiments. For this special case, the gradient is given by

$$\begin{aligned}\frac{\partial \Upsilon(\mathcal{O}^l,\Phi)}{\partial \mathcal{B}_i} &= -\psi\sum_{t\in T_i^l}\sum_{m=1}^{M}\gamma_{i,m}(t)\big(\Sigma_{x,i,m}^{-1}[\mathcal{X}_t^l-\mu_{x,i,m}]\\ &\quad[\mathcal{F}_t^l]^{\mathrm{Tr}}+\Sigma_{y,i,m}^{-1}[\mathcal{Y}_t^l-\mu_{y,i,m}][\mathcal{F}_{t+2}^l-\mathcal{F}_{t-2}^l]^{\mathrm{Tr}}\big).\end{aligned}$$

2) Gradient Computation of Both $w_{k,i,m}$ and $\mathcal{B}_{i,m}$ for THMM-2: Substitution of (1) and (4) in (13) yields

$$\begin{aligned}\frac{\partial \Upsilon(\mathcal{O}^l,\Phi)}{\partial \mathcal{B}_{i,m}} &= \psi\sum_{t\in T_i^l}\gamma_{i,m}(t)\frac{\partial}{\partial \mathcal{B}_{i,m}}\frac{-1}{2}\Bigg([\mathcal{B}_{i,m}\mathcal{F}_t^l-\mu_{x,i,m}]^{\mathrm{Tr}}\\ &\quad\times\Sigma_{x,i,m}^{-1}[\mathcal{B}_{i,m}\mathcal{F}_t^l-\mu_{x,i,m}]\\ &\quad+\left[\mathcal{B}_{i,m}\sum_{k'=-b}^{f}w_{k',i,m}\mathcal{F}_{t+k'}^l-\mu_{y,i,m}\right]^{\mathrm{Tr}}\\ &\quad\times\Sigma_{y,i,m}^{-1}\left[\mathcal{B}_{i,m}\sum_{k'=-b}^{f}w_{k',i,m}\mathcal{F}_{t+k'}^l-\mu_{y,i,m}\right]\Bigg)\\ &= -\psi\sum_{t\in T_i^l}\gamma_{i,m}(t)\Bigg(\Sigma_{x,i,m}^{-1}[\mathcal{X}_t^l-\mu_{x,i,m}][\mathcal{F}_t^l]^{\mathrm{Tr}}\\ &\quad+\Sigma_{y,i,m}^{-1}[\mathcal{Y}_t^l-\mu_{y,i,m}]\left[\sum_{k'=-b}^{f}w_{k',i,m}\mathcal{F}_{t+k'}^l\right]^{\mathrm{Tr}}\Bigg)\end{aligned} \tag{15}$$

$$\begin{aligned}\frac{\partial \Upsilon(\mathcal{O}^l,\Phi)}{\partial w_{k,i,m}} &= \psi\sum_{t\in T_i^l}\gamma_{i,m}(t)\frac{\partial}{\partial w_{k,i,m}}\frac{-1}{2}\\ &\quad\times\left(\left[\sum_{k'=-b}^{f}w_{k',i,m}\mathcal{X}_{t+k'}^l-\mu_{y,i,m}\right]^{\mathrm{Tr}}\right.\\ &\quad\left.\times\Sigma_{y,i,m}^{-1}\left[\sum_{k'=-b}^{f}w_{k',i,m}\mathcal{X}_{t+k'}^l-\mu_{y,i,m}\right]\right)\\ &= -\psi\sum_{t\in T_i^l}\gamma_{i,m}(t)[\mathcal{Y}_t^l-\mu_{y,i,m}]^{\mathrm{Tr}}\\ &\quad\times\Sigma_{y,i,m}^{-1}[\mathcal{B}_{i,m}\mathcal{F}_{t+k}^l].\end{aligned} \tag{16}$$

The gradient computation for the remaining model parameters is similar to those for the conventional HMM. For keeping this paper self contained, we list these formulas below without derivation:

$$\frac{\partial \Upsilon(\mathcal{O}^l,\Phi)}{\partial \mu_{x,i,m}} = \psi \sum_{t\in T_i^l} \gamma_{i,m}(t)\Sigma_{x,i,m}^{-1}\Lambda_{x,i,m}(t) \tag{17}$$

$$\frac{\partial \Upsilon(\mathcal{O}^l,\Phi)}{\partial \mu_{y,i,m}} = \psi \sum_{t\in T_i^l} \gamma_{i,m}(t)\Sigma_{y,i,m}^{-1}\Lambda_{y,i,m}(t) \tag{18}$$

$$\frac{\partial \Upsilon(\mathcal{O}^l,\Phi)}{\partial \tilde{\Sigma}_{x,i,m}} = 0.5\psi \sum_{t\in T_i^l} \gamma_{i,m}(t)\left[\Lambda_{x,i,m}^{\mathrm{Tr}}(t)\Sigma_{x,i,m}^{-1}\Lambda_{x,i,m} - I\right] \tag{19}$$

$$\frac{\partial \Upsilon(\mathcal{O}^l,\Phi)}{\partial \tilde{\Sigma}_{y,i,m}} = 0.5\psi \sum_{t\in T_i^l} \gamma_{i,m}(t)\left[\Lambda_{y,i,m}^{\mathrm{Tr}}\Sigma_{y,i,m}^{-1}\Lambda_{y,i,m} - I\right] \tag{20}$$

$$\frac{\partial \Upsilon(\mathcal{O}^l,\Phi)}{\partial \tilde{c}_{i,m}} = \psi \sum_{t\in T_i^l(j)} \gamma_{i,m,t}(j) - c_{i,m}(j) \tag{21}$$

where the quantities $\Lambda_{x,i,m}(t)$ and $\Lambda_{y,i,m}(t)$ are defined as

$$\Lambda_{x,i,m}(t) = \mathcal{B}_{i,m}\mathcal{F}_t^l - \mu_{x,i,m}$$

$$\Lambda_{y,i,m}(t) = \mathcal{B}_{i,m}\sum_{k=-b}^{f} w_{k,i,m}\mathcal{F}_{t+k}^l - \mu_{y,i,m}$$

and $\tilde{\Sigma}_{y,i,m}$ are the log-transformed covariance matrices for implementation simplicity [5].

IV. Experiments and Model Evaluation Results

The proposed new model, THMM, and the associated discriminative training described in Sections II and III have been evaluated on the phonetically rich, speaker-independent TIMIT database. Several phonetic classification experiments were conducted to study the characteristics of the MCE-based training for the new model and for demonstration of the superiority of the new model over the traditional HMM.

The TIMIT database with a total of 462 different speakers is divided into a training set and a test set with no overlapping speakers. Out of the ten sentences per speaker, two **sa** sentences are common to all speakers and were removed from both training and test sets in order to avoid possible biasness. The training set consists of 442 speakers with a total 3536 sentences, and the test set consists of 160 sentences spoken by 20 disjoint speakers. These speech materials contain a total of 129 743 phone tokens in the training set and 5775 phone tokens in the test set. The experiments described in this section are aiming at classifying the 61 TIMIT labels defined in the TIMIT database. In keeping with the convention adopted by many speech recognition researchers, we folded 22 phone labels into the remaining 39 classes in determining classification accuracy.

For the computation of MFB log channel energies that serve as data input to feature transformation, 21 triangular filters are used in our experiments, which are spaced linearly from 0 to 500 Hz, and exponentially from 500 Hz to 8500 Hz.

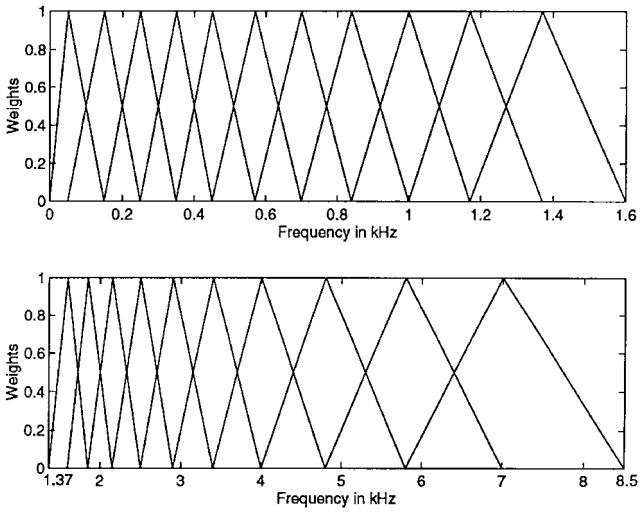

Fig. 2. Spacing of the 21 triangular filters used in the experiments for generating the mel-filterbank log channel energies.

The adjacent filters are overlapped by 50% in the frequency axis, shown in Fig. 2. The frequency components below 70 Hz are treated as noise and are removed in the filtering process. The raw speech waveforms in TIMIT are sampled at 16 kHz, and are blocked into 512 samples to form 10-ms frames. An overlap of 352 samples between two adjacent data blocks is used in the analysis. Each frame is then passed through a 512-point Hamming window, and a 512-point fast fourier transform (FFT) is applied to the frame to produce a 256-point power spectrum. The FFT power spectra are combined using a weighted sum, shaped by the triangular filter, to obtain the filter output. Logarithms of the 21 outputs are then calculated, arriving at 21 MFB log channel energies for each speech frame. For the THMM, only these MFB log channel energy vectors are used as the raw data to the recognizer. All the feature parameters are automatically constructed within the recognizer.

In our experiments, each phone defined in TIMIT is represented by a simple left-right (i.e., with only self and forward transition) three-state HMM with mixture Gaussian densities. The covariance matrices in all the states of all the models are diagonal and are not tied. To avoid singularities caused by an underestimation of the variance, we assigned the minimum variance (typically a value of 0.1) in covariance matrices. All transition probabilities are uniformly set to 0.5 (all transitions from a state are considered equally likely) and are not trained, since they are found to play a minor role in the forward-backward probability scoring.

For the MCE approach, the initial model is trained using the ML criterion [17], [8]. The state dependent transformation matrix is initialized by the DCT matrix

$$\mathcal{A}_{p,q} = \sum_{q=1}^{n} \cos\left[p(q-0.5)\frac{\pi}{n}\right]$$
$$p = 1,2,\cdots,d,\ t = 1,2,\cdots,T^l$$

where $\mathcal{A}$ denotes the $d \times n$ DCT matrix, and d is the dimensionality of the static feature vector. Similarly, the state-dependent dynamic weighting coefficients are fixed to a first

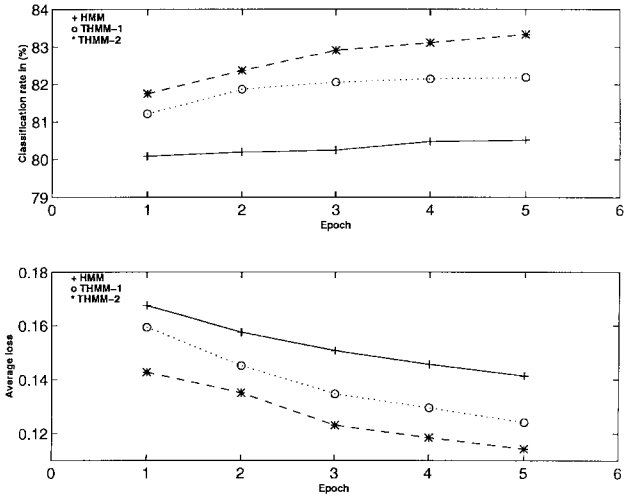

Fig. 3. Convergence characteristics of the MCE training procedure. Top graph: phonetic classification rates on the test set for three types of classifiers as a function of the MCE training epoch number. Bottom graph: average loss computed from training set for three types of classifiers as a function of the MCE training epoch number.

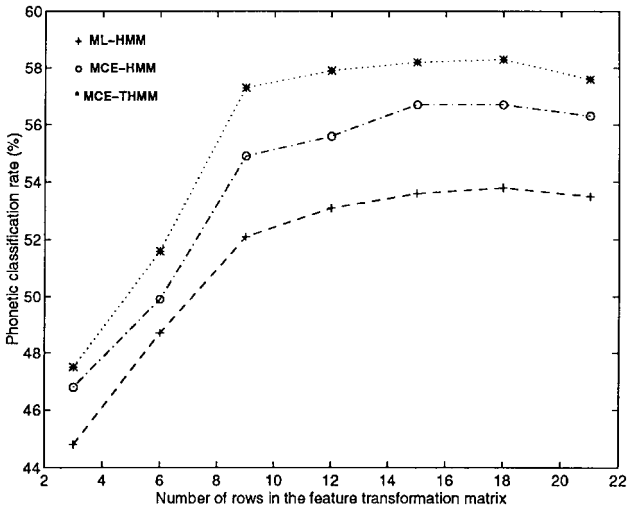

Fig. 4. Results of a fast experiment intented to find the optimal dimension of the feature transformation matrix. Phonetic classification rates for three types of classifiers are plotted as a function of the number of rows in the transformation matrix.

difference condition: $f = b = 2, w_{2,i,m} = 1.0$ and $w_{-2,i,m} = -1.0$. Note that the above initialization of the transformation matrix by DCT matrix without further training gives rise to the traditional MFCC feature parameters.

During the model training phase, we call one complete pass through the training data set as an epoch. For the case of token-by-token training, model parameters are updated several times over an epoch. Additionally, in each epoch we also decrease the step size monotonically $\epsilon_k = \epsilon(1 - \frac{\epsilon_k}{\tau})$, where k is the epoch number, τ is the limit for the number of epochs and $\epsilon = 0.01$ is a small positive learning constant. The classification performance often peaked after about four or five epochs and then again varied randomly within about one percent. In training, we perform a total of five epochs and only the best-incorrect-class[2] is used in the misclassification measure. Phonetic classification is performed directly from the standard Viterbi score calculation.

Context-independent (CI) phone models assume that speech is produced as a sequence of concatenated phones which are unaffected by context. For the CI experiments, a total of 39 models (with $39 \times 3 = 117$ states) were constructed, one for each of the 39 classes intended for the classification task. A context-dependent (CD) phone model is one that is dependent on the left and right neighboring phone. With 39 phone classes, there are potentially $39 \times 39 \times 39 = 59\,319$ phone models constituting 177 957 states, which are impractical to train given the limited amount of training data. As an alternative, the procedure outlined in [24] has been adopted to create CD models, resulting in approximately a total of 1209 states.

Before we present the full set of phonetic classification results, we first present the convergence characteristics of the MCE training. Fig. 3 shows the classification rate of the test set (top plot) and the average loss (bottom plot), defined by (7), of the training set for three different classifiers[3] as a function of the training epoch number. In Fig. 3, the solid lines are associated with MCE-trained conventional HMM, dotted lines with THMM-1, and dashed lines with THMM-2. The classification rates and the average loss are evaluated at the end of every epoch. As shown in Fig. 3, the classification rates are monotonically increased and the average loss is monotonically decreased as the training progresses. The average loss decreases faster for THMM-1 and THMM-2 than the conventional HMM, indicating the effectiveness of the new THMM models. Similar trends in the classification performance are also observed. This suggests that the original objective of minimizing the misclassification error using the MCE training is indeed achieved and that the MCE training may be more effective for the THMM than the conventional HMM.

In our phonetic classification experiments, we use feature transformation to reduce the dimensionality of the raw data. It converts the partially preprocessed speech data to a suitable form (feature vectors) for use as the input to the HMM for modeling and classification. Obviously, the transformed feature vector $\mathcal{X}_t^l$ must have a smaller dimension than that of the MFB log channel energies (i.e., $d < n = 21$). To determine the best dimension, d, for use in phonetic classification tasks, a series of fast experiments are conducted using a subset of training-set, consists of 320 sentences from each of 40 speakers, and test set with unimodal Gaussian CI phone models. The results expressed as phonetic classification rate as a function of dimension d are plotted in Fig. 4, with the ML-trained conventional HMM (i.e., using MFCC's[4]) plotted as the dashed-dashed line, MCE-trained conventional HMM as the dashed-dotted line, and MCE-trained THMM-1 as the dotted-dotted line. (The ML-trained HMM with state-dependent DCT matrices, or MFCC's, is provided as the initial

[2]This is a computationally efficient way of pruning the search in the discriminative training.

[3]The results are obtained with use of fivey-mixtures CD phone models.

[4]The feature ordering for MFCC's was performed by selecting lowest order MFCC's first.

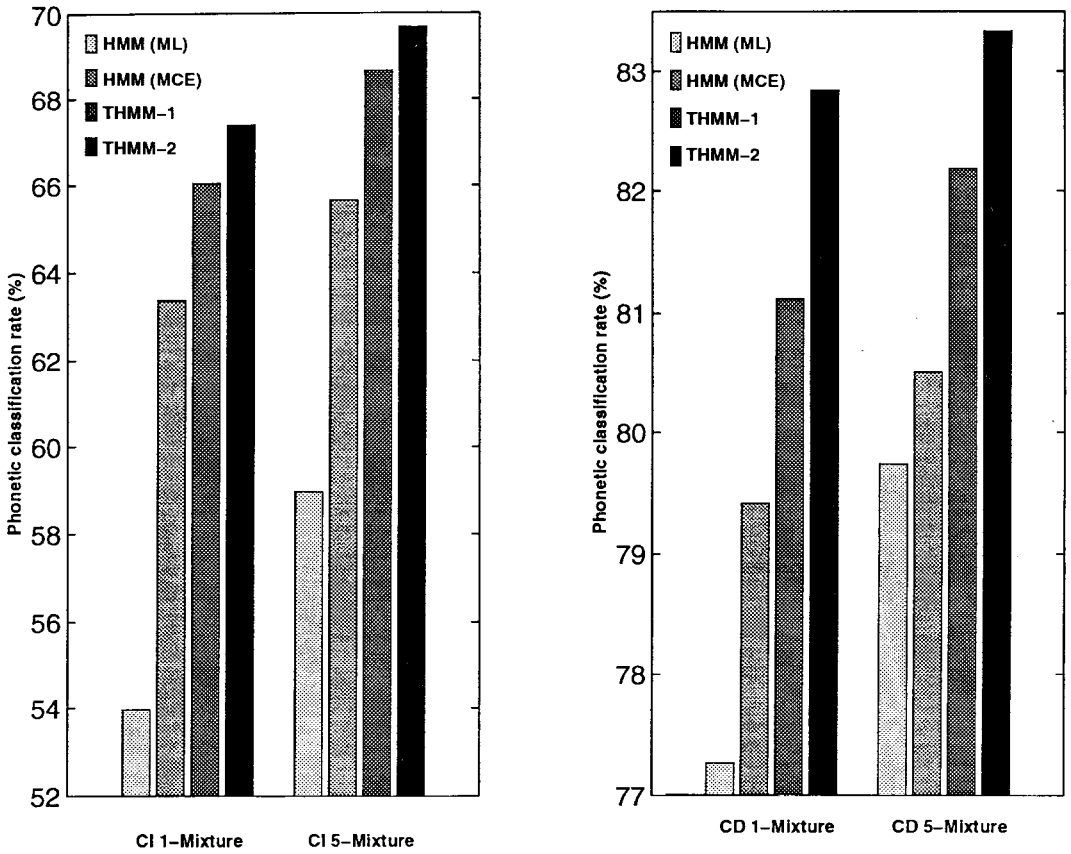

Fig. 5. TIMIT 39-phone context-independent (left) and context-dependent (right) classification rates as a function of the classifier types and of the number of Gaussian mixtures in the model state. Four types of classifiers are evaluated and compared: two benchmarks based on ML-trained and MCE-trained conventional HMM's using MFCC's, and two versions of the THMM's, both trained by MCE.

model for the MCE training of THMM, and in these fast experiments, all states of all models use an identical number of features.) From the results shown in Fig. 3, we observe that the classifier performance remains fairly constant after d reaches twelve. Therefore, in our following formal, more comprehensive experiments, we choose $d = 12$, making the dimensions of the linear transformation matrix to be 12×21. This gives the total feature vector consisting of 26 elements: one normalized log energy, 12 transformed MFB log channel energies, one delta log energy, and 12 delta transformed MFB log channel energies.

A series of comparative experiments have been carried out, using full sets of training and test data, to examine the effectiveness of the MCE training on the proposed THMM-1 and THMM-2 as described in Sections II and III. The experimental results are summarized in Fig. 5. For performance comparison with benchmarks, conventional HMM's with use of static and dynamic MFCC features are first implemented. The static MFCC features are obtained by taking DCT of the MFB log channel energies and the dynamic features are calculated as the difference between the static feature vectors for two-frame ahead and two-frame behind each current time. The left and right plots of Fig. 5 give the performance comparison (among four types of classifiers) for CI and CD phones, respectively. For both CI and CD cases, we evaluate unimodal Gaussian HMM's ($M = 1$) and mixture Gaussian HMM ($M = 5$) separately. The four types of classifiers are as follows. The first classifier, denoted by HMM (ML) in Fig. 5, is designed with conventional HMM's as a benchmark using MFCC and delta-MFCC features and being trained with five iterations of Baum–Welch (ML) algorithm. This HMM (ML) is comparable in performance with other similar classifiers (e.g., [25]), and gives 65.7% phone classification rate using five-mixture CI models (about the same as 66.2% reported in [25] with 32 mixtures). The second classifier, denoted by HMM[5] in Fig. 5, is designed also with conventional HMM's as another benchmark using identical MFCC and delta-MFCC features, but being trained with the MCE algorithm (five epochs). The best classification results obtained are 65.66% and 80.51% for HMM with five-mixtures CI and CD models, respectively. As can be seen from Fig. 5, the performance in terms of classification accuracy was significantly improved by the MCE training over the ML counterpart.

The third and the fourth classifiers, denoted by THMM-1 and THMM-2, respectively, in Fig. 5, are designed with the two versions of the THMM described in Section II and trained using the MCE algorithm presented in detail in Section III. Our goal is to test the effectiveness of incorporating the optimal state-dependent transforms on raw features in the classification performance. Since good initialization of transformation matrices is important to avoid local optimum that would necessarily occur due to the use of gradient descent,

[5] All remaining three classifiers use MCE training rather than ML training, so the label MCE will not be attached in Fig. 5.

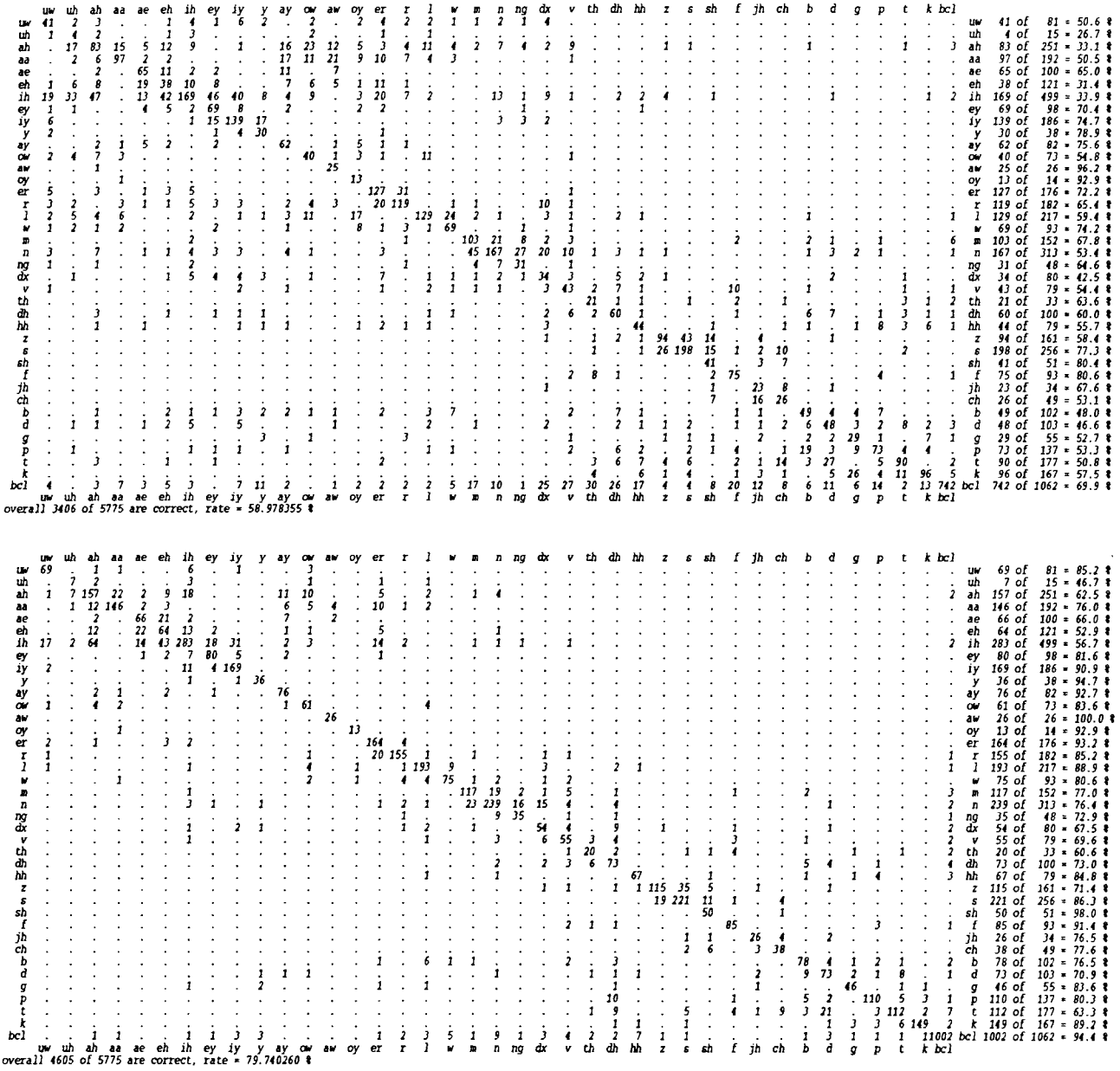

Fig. 6. Confusion matrix for the standard TIMIT 39-phone classification task with use of conventional HMM's (MFCC features not optimized) trained by the ML criterion. The top matrix shows results with context-independent models and the bottom matrix shows results with context-dependent models.The models have five Gaussian mixtures in each state.

we use the first benchmark classifier (HMM (ML)), modified by introducing a state-specific DCT matrix, to initialize the THMM's for the MCE training. In the discriminative training, the initial DCT matrices are progressively improved according to (14) in a state-dependent manner. For the THMM-1 based classifier, the best classification results are 68.6% (CI) and 82.2% (CD) with five mixtures (see the results labeled with THMM-1 in Fig. 5). The CD result of 82.19% is translated to reduction in error rate of 9% compared with the benchmark HMM (MCE) classifier's performance. For all other cases, the THMM-1 outperforms the conventional HMM by at least 7% in error rate reduction. We also from the results in Fig. 5 that the unimodal Gaussian (1-mixture) THMM-1 performs better than the five-mixtures MCE-based HMM for both CI and CD cases. In either case, the unimodal Gaussian THMM-1 and five-mixtures conventional HMM have comparable numbers of classifier parameters: The number of classifier parameters for one-mixture THMM-1 is $21 \times 12 + 26 + 26 = 304$, similar to that of the five-mixtures HMM, which is $5 \times (26 + 26 + 1) = 265$. Comparison of the performance improvement with other systems, unfortunately, is somewhat difficult because of the completely different model structure and entirely different training criterion. For example, a closest comparison can be made with the study reported in [27] where error rate reduction of 3.8% was obtained using HMM's with state-specific linear discriminant analysis on the input features (which is state independent, different from our state-dependent transformation).

From the final classifier based on THMM-2, which incorporates generalized dynamic weighting functions, the best

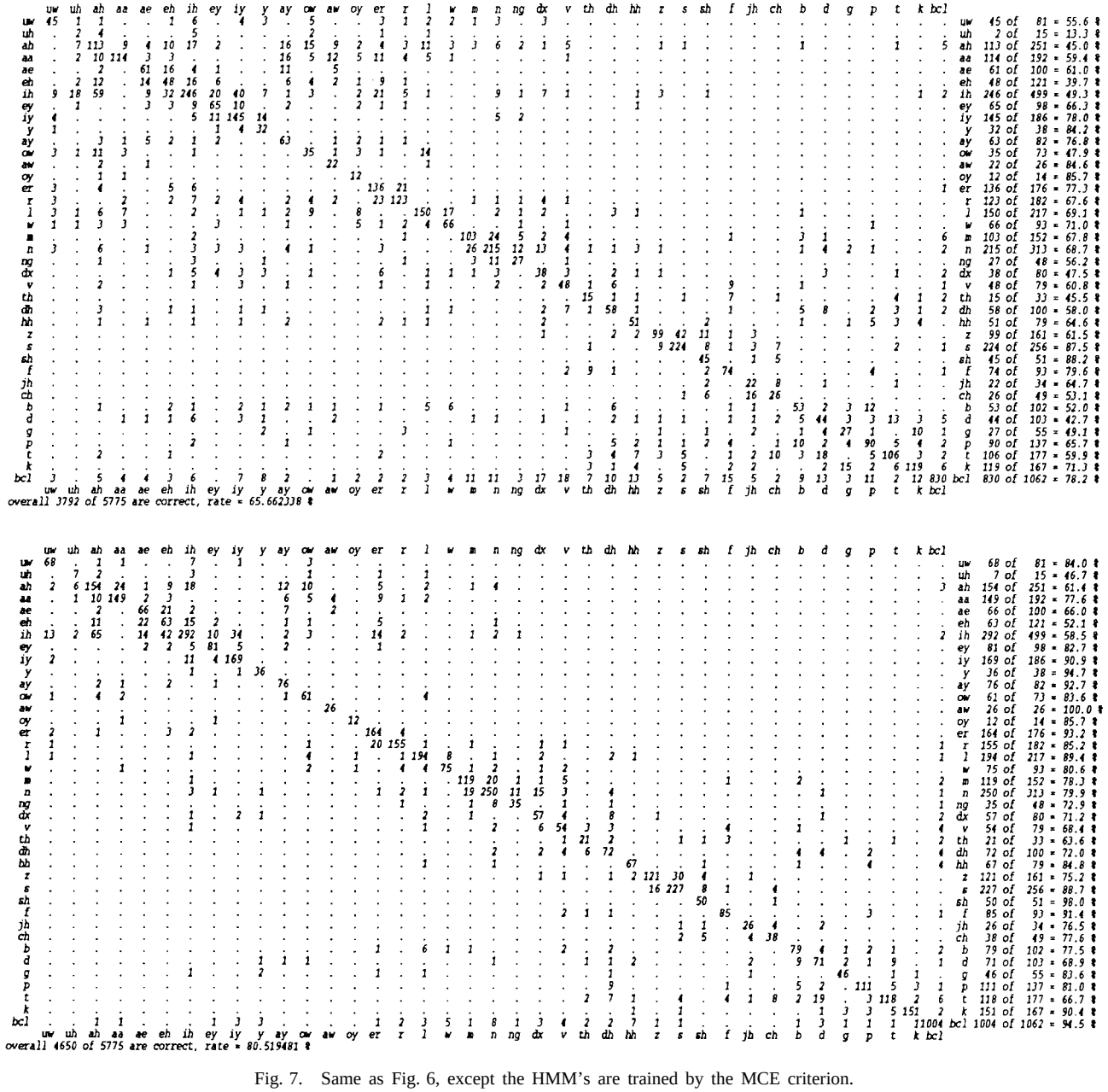

Fig. 7. Same as Fig. 6, except the HMM's are trained by the MCE criterion.

classification results have been obtained, shown as the THMM-2 label in Fig. 5. In training the THMM-2 classifier, (15) and (16) are used to simultaneously adjust the static linear transformation matrices and dynamic weighting parameters. This joint training of both static-feature- and dynamic-feature-related classifier parameters produces performance improvement over the THMM-1 (where no optimization on dynamic-feature related parameters) more significantly for CD (right plot of Fig. 5) than for CI models (left plot of Fig. 5). The classification rates with the THMM-2 classifier improve from 80.51% (benchmark MCE-trained HMM) 83.4% with use of five-mixture CD models. This translates to 15% error rate reduction. For single-mixture CD models, the corresponding error rate reduction from benchmark to THMM-2 is 17%, with classification rate improvement from 79.41% to 82.84%. On the other hand, we observe that while the MCE training substantially improves the classification performance, the improvement relative to the corresponding ML case decreases as the model complexity increases. For example, an error rate reduction of 25% for the single mixture case is reduced to an error rate reduction of 18% (five-mixture CD models) when compared with respect to the ML-trained HMM benchmark. This suggests that if a larger amount of training data were available, the performance improvement achievable using the MCE-trained THMM-2 would be even more significant than reported here.

In analyzing the comparative performances of the various classifiers shwon in Fig. 5, we have noted the issue of fairness of comparison with respect to the varying numbers of model parameters. Admittedly, with use of the same number of mixtures, the THMM's have one additional trainable full matrix each state compared with the conventional HMM.

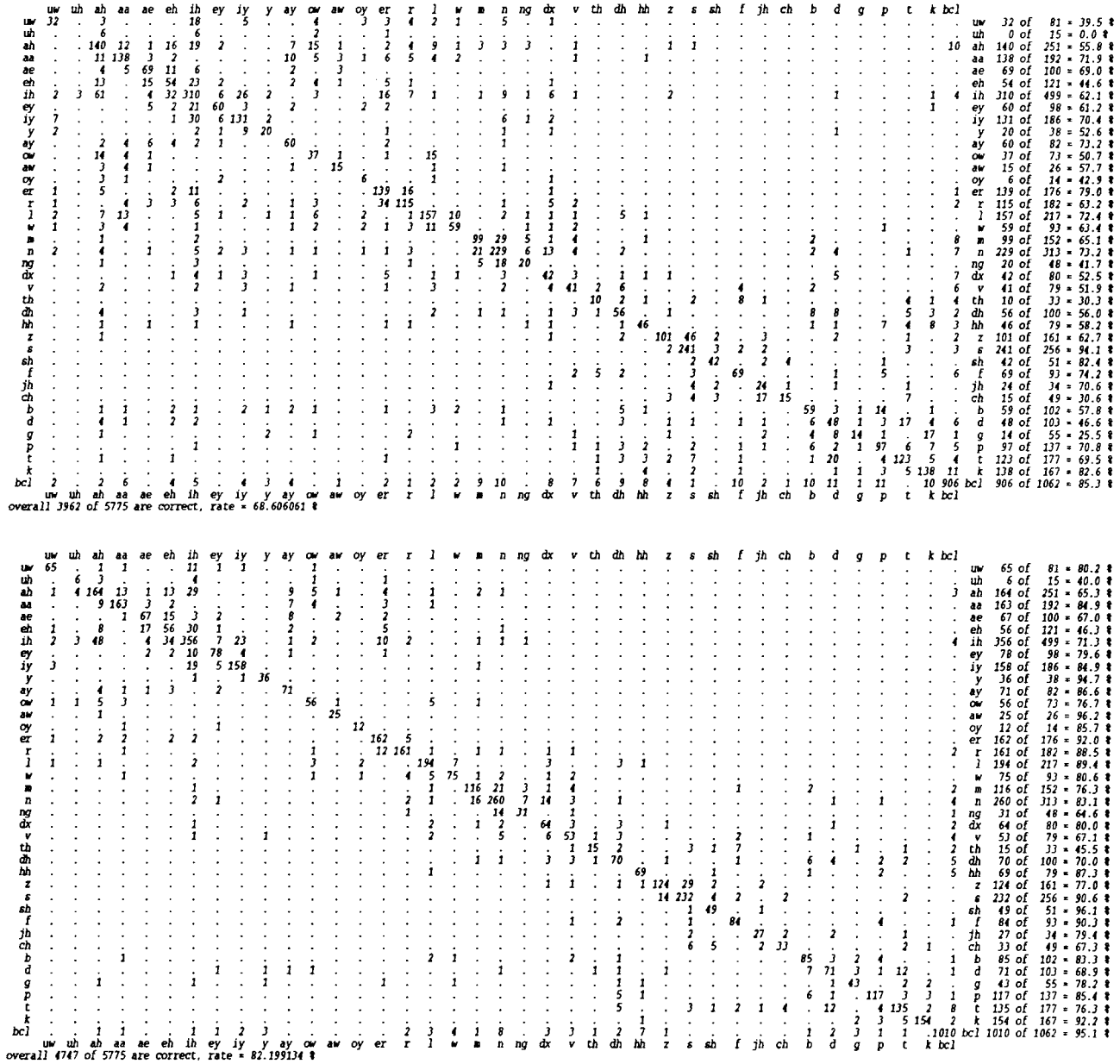

Fig. 8. Same as Fig. 6, except the classifier is based on the THMM-1 trained with the MCE criterion.

However, in comparing the THMM's with only one mixture with the conventional HMM with five mixtures, we still observe consistently better performance of the THMM's in both the CI and CD cases (Fig. 5). This suggests that the performance improvement of THMM's is not simply due to the greater number of model parameters. Rather, the state-dependent feature transformation incorporated into the structure of the THMM's accounts for much of the performance gain.

To provide further details of the phonetic classification results, we include in the Appendix several confusion matrices associated with various summarized results shown in Fig. 5.

V. Summary and Discussions

In this study, we have developed a novel technique for discriminative feature extraction from the mel-warped log channel energies computed directly from short-time DFT's of the speech waveform; this feature-extraction technique has been integrated with the design of an HMM-based speech recognizer. The preprocessing (or feature extraction) component and the modeling (or classification) component of the recognizer are jointly trained with a discrimination-motivated MCE algorithm. A significant aspect of our research is that it represents the effort toward the automatic definition of speech-dependent acoustic parameters, which are subject to statistical optimization rather than relying on heuristic construction. Along this line, we note an earlier work as a representative of the nonparametric (speech-frame based) approach to this problem [3]. Our own earlier parametric (HMM-state based) approach [7], [24] has been extended in this study from the previous level of MFCC to the present level of log-channel energy computed from DFT's, a step closer

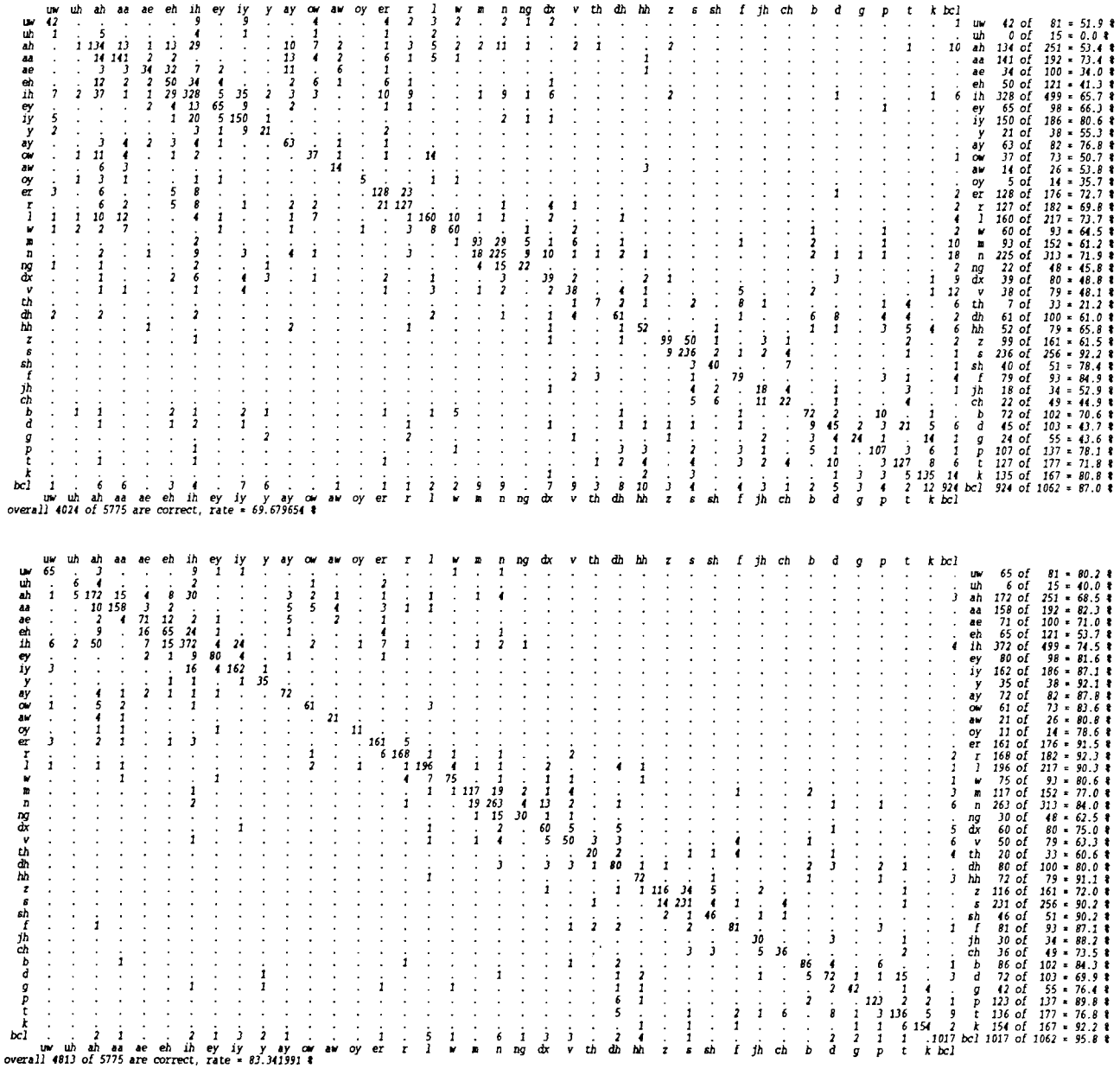

Fig. 9. Same as Fig. 6, except the classifier is based on the THMM-2 trained with the MCE criterion.

toward the most primitive form of the data as speech waveform.[6]

At the writing of this article, a few studies appeared in the literature that are also aimed at the same goal of integrating the design of the preprocessing and modeling components in speech recognizers. In particular, the work of [9] uses a similar type of MCE algorithm for a global linear transformation on linear predictive coefficient-based (LPC-based) cepstral coefficients. This is a special case of the method we have presented in this paper in that our transformation is made dependent on each speech class and on each HMM state. Moreover, our method reported in this paper goes further toward the raw information contained in the speech waveform by applying the transformation on acoustic measurements of speech at the filterbank output level rather than at the cepstral level. Another set of studies [20]–[22] intended to use more general, neural-net-like nonlinear transforms on the primitive acoustic measurements of speech. However, due to the implementation complexity, only the linear version of the transform has been reported.

In this study, we have developed and evaluated two versions of the optimum-transformed HMM: THMM-1 and THMM-2. THMM-1 performs state-dependent linear transformation on the mel-warped log channel energies (static) in a way that is independent of the successive frames (i.e., no optimization of dynamic feature parameters). THMM-2 generalizes THMM-1

[6] An earlier attempt to design a statistical speech recognizer using raw speech waveforms directly as the input features [26] encountered two main difficulties: i) prohibitively high computation burden for implementing a large system, and ii) less accurate modeling assumptions made in the statistical model (hidden filter model) characterizing the statistical properties of the speech waveform, in comparison with the models which characterize the statistical properties of the relatively slowly changing frame-based spectral features.

and includes a jointly optimal transformation so as to arrive at the static and dynamic parameters together. We found in our experiments that the empirical average loss computed from the training samples decreases faster for both versions of the THMM than for conventional HMM. This leads us to believe that the objective of minimizing the misclassification error intented with the MCE criterion is achieved more effectively for the new THMM's than for the conventional HMM.

We have conducted a series of phone classification experiments using TIMIT to evaluate the performance of the THMM's. To make the experimental results interpretable, all of our experiments have used a fixed number of features. An attempt has been made to use a best static feature dimension of 12, which has been determined by an early fast experiment. Experimental results show that use of the state-dependent transformation on mel-warped log-channel energies is superior in performance to the conventional use of the MFCC's, which are not subject to optimization together with the model parameters in training. Overall, an error-rate reduction of 15% is achieved on a standard 39-class phone classification task in comparison with the conventional MCE-trained HMM using MFCC's.

For THMM-1, the best classification rate of 82.2% is obtained using five-mixtures context-dependent models, compared with 80.5% with the conventional MCE-trained HMM. Further improvement of about 8% error-rate reduction is achieved moving from THMM-1 to THMM-2 by incorporating the state-dependent generalized dynamic feature parameters. Compared across four classifiers (two versions of THMM's and two benchmark HMM's), THMM-2 consistently produces the lowest error rate due to its new, efficient way of organizing and utilizing the input data in the form of mel-warped log-channel energies.

We believe that the results reported in this paper are the first to demonstrate that the mel-warped DFT features, subject to appropriate transformation in a state-dependent manner, are more effective than the MFCC's that have dominated current speech recognition technology. To the best of our knowledge, our performance results based on joint optimization of DFT-derived features and of HMM parameters are the best reported in the literature on speaker-independent TIMIT phonetic classification task using comparably sized training data. Although the experiments reported in this paper are limited to only the phonetic classification task, the model is well suited for use in continuous speech recognition tasks. Demonstration of the effectiveness of the THMM proposed in this paper in large scale continuous speech recognition tasks will be our future effort.

Eight sample confusion matrices that supplement the phonetic classification results shown in Fig. 5 are shown in Figs. 6–9.

ACKNOWLEDGMENT

The authors would like to thank Dr. C. Lee for valuable discussions on the MCE approach, and they would like to thank the reviewers and the associate editor who provided valuable suggestions improving the quality of the paper.

REFERENCES

[1] A. Biem and S. Katagiri, "Filter bank design based on discriminative feature extraction," in *IEEE Proc. ICASSP*, 1994, pp. 485–488.
[2] E. Bocchieri and J. Wilpon. "Discriminative feature selection for speech recognition," *Comput. Speech Language*, no. 7, pp. 229–246, 1993.
[3] E. Bocchieri and G. Doddington, "Frame specific statistical features for speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 755–764, Aug. 1986.
[4] W. Chou, C. H. Lee, B. H. Juang, and F. K. Soong, "A minimum error rate pattern recognition approach to speech recognition," *Int. J. Pattern Recog. Artif. Intell.*, vol. 8, no. 1, pp. 5–31.
[5] W. Chou, B. H. Juang, and C. H. Lee, "Segmental GPD training of HMM based speech recognizer," in *IEEE Proc. ICASSP*, 1992, pp. 473–476.
[6] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, pp. 357–366, 1980.
[7] L. Deng, "Integrated optimization of dynamic feature parameters for hidden Markov modeling of speech," *IEEE Signal Processing Lett.*, vol. 1, no. 4, pp. 66–69.
[8] L. Deng *et al.*, "Phonemic hidden Markov models with continuous mixture output densities for large vocabulary word recognition," *IEEE Trans. Acoust, Speech, Signal Processing*, vol. 39, pp. 1677–1681, July 1991.
[9] S. Euler, "Integrated optimization of feature transformation for speech recognition," in *Proc. EUROSPEECH*, Sept. 1995, vol. 1, pp. 109–112.
[10] K. S. Fu, *Sequential Methods in Pattern Recognition and Machine Learning*. New York: Academic, 1968.
[11] S. Furui, "On the role of spectral transition for speech perception," *J. Acoust. Soc. Amer.*, vol. 80, no. 4, pp. 1016–1025.
[12] B. Hanson and T. Applebaum, "Robust speaker-independent word recognition using static, dynamic and acceleration features: Experiments with lombard and noisy speech," in *Proc. ICASSP*, 1991, pp. 857–860.
[13] C. R. Jankowski, H. Vo, and R. P. Lippman, "A comparison of signal processing front ends for automatic word recognition," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 286–293, July 1995.
[14] B. H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Processing*, vol. 40, pp. 3043–3054, Dec. 1992.
[15] S. Katagiri, B. H. Juang, and A. Biem, "Discriminative feature extraction," in *Artificial Neural Networks for Speech and Vision*. London, U.K.: Chapman and Hall, 1993, pp. 278–293.
[16] H. Leung, B. Chigier, and J. Glass, "A comparative study of signal representations and classification techniques for speech recognition," in *Proc. ICASSP*, 1993, pp. 680–683.
[17] L. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Inform. Theory*, vol. 28, pp. 729–734, 1982.
[18] C.-S. Liu, C.-H. Lee, B.-H. Juang, and A. Rosenberg, "Speaker recognition based on minimum error discriminative training," in *Proc. ICASSP*, 1994, pp. 325–328.
[19] E. McDermott and S. Katagiri, "Prototype-based minimum classification error/generalized probabilistic descent training for various speech units," *Comput. Speech Language*, vol. 8, pp. 351–368, 1994.
[20] K. K. Paliwal, M. Bacchiani, and Y. Sagisaka, "Minimum classification error training algorithm for feature extractor and pattern classifier in speech recognition," in *Proc. Europ. Conf. Speech Commununication Technology*, 1995, vol. 1, pp. 541–544.
[21] M. Rahim, C. H. Lee, and B. H. Juang, "An integrated ANN-HMM speech recognition system based on minimum classification error training," in *Proc. 1995 IEEE Workshop on Automatic Speech Recognition*, pp. 143–144.
[22] M. Rahim and C. H. Lee, "Simultaneous feature and HMM design using string-based minimum classification error training criterion," in *Proc. ICSLP*, Oct. 1996, pp. 1824–1827.
[23] R. Chengalvarayan and L. Deng, "Use of generalized dynamic feature parameters for speech recognition," *IEEE Trans. Speech Audio Processing*, this issue, pp. 232–242.
[24] ———, "Use of generalized dynamic feature parameters for speech recognition: Maximum likelihood and minimum classification error approaches," in *Proc. ICASSP*, 1995, pp. 373–376.
[25] S. Sandhu and O. Ghitza, "A comparative study of mel-cepstra and EIH for phone classification under adverse conditions," in *Proc. ICASSP*, 1995, pp. 409–412.
[26] H. Sheikhzadeh and L. Deng, "Waveform-based speech recognition using hidden filter models: Parameter selection and sensitivity to power

normalization," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 80–91, Jan. 1994.

[27] P. Woodland and D. Cole, "Optimizing hidden markov models using discriminative output distributions," in *Proc. ICASSP*, 1991, pp. 545–548.

[28] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, *Springer Series in Information Sciences* New York: Springer-Verlag, 1990, pp. 133–155.

Rathinavelu Chengalvarayan (S'92–M'96), for photograph and biography, see this issue, p. 242.

Li Deng (S'83–M'86–SM'91), for photograph and biography, see this issue, p. 242.