THE TRENDED HMM WITH DISCRIMINATIVE TRAINING FOR PHONETIC CLASSIFICATION

C. Rathinavelu and Li Deng

Department of Electrical and Computer Engineering, University of Waterloo, Ontario, Canada

ABSTRACT

In this paper, we extend the Maximum Likelihood (ML) training algorithm to the Minimum Classification Error (MCE) training algorithm for optimal estimation of the state-dependent polynomial coefficients in the trended HMM [2]. The problem of automatic speech recognition is viewed as a discriminative dynamic data-fitting problem, where *relative* (not absolute) closeness in fitting an array of dynamic speech models to the unknown speech data sequence provides the recognition decision. In this view, the properties of the MCE formulation for training the trended HMM are analyzed by fitting raw speech data using MCE-trained trended HMMs, contrasting the poor discriminative fitting using the ML-trained models. Comparisons between the phonetic classification as well as data-fitting results obtained with ML and with MCE training algorithms demonstrate the effectiveness of the discriminatively trained trended HMMs.

1. INTRODUCTION

The formulation of the trended HMM (also called the parametric nonstationary-state HMM or trajectory model) has been successfully used in automatic speech recognition applications for the past few years [1, 2]. The model parameters of the trended HMM, including 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. The method of ML, however, need not be optimal in terms of minimizing classification error rate in recognition tasks in which the observation is assumed to be produced by one of the many source classes. Only the in-class information is available to train each model in the ML approach, which leads to a poor discriminative ability. Discrimination can be improved if out-of-class information is also used in training the models. An alternative model estimation criterion to the ML, called discriminative training [3] has been proposed to improve the discrimination ability of the models. This training approach takes into account other competing models and aims at minimizing the recognition error rate of the training data.

In this study, the minimum classification error (MCE) training algorithm, which minimizes the misclassification error based on a given training samples using a gradient descent method, is applied for estimating the state-dependent polynomial coefficients in the trended HMM. The properties of the MCE formulation for training the trended HMM is analyzed by fitting raw speech data using MCE trained models, contrasting the poor discriminative fitting using the ML trained models, and positive experimental results on phonetic classification using the TIMIT database are reported.

2. THE TRENDED HMM

The nonstationary state or trended HMM is of a data-generative type and can be written as

$$\begin{aligned} O_t &= F_t(i) + R_t(\Sigma_i), \qquad (1) \\ &= \sum_{p=0}^{P} B_i(p)(t-\tau_i)^p + R_t(\Sigma_i) \end{aligned}$$

where O_t, $t = 1, 2, \cdots, T$ are the modelled data sequences of length T within the HMM state indexed by i, $B_i(p)$ is the state-dependent polynomial regression coefficients of order P, the term R_t is the stationary residual (after the data-fitting by the first term F_t) assumed to be IID and zero-mean white Gaussian source characterized by covariance matrix Σ_i.

In the conventional HMM [4], the first term is only a function of state i, not a function of time t. Note that the polynomial for each state depends not only on the coefficients $B_i(p)$, but also on the time-shift parameter τ_i. The term $t - \tau_i$ represents the sojourn time in state i at time t, τ_i registers the time when state i in the HMM is just entered before regression on time takes palce. Polynomial coefficients $B_i(p)$ are considered as true model parameters and τ_i is merely an auxiliary parameter for the purpose of obtaining maximal accuracy in estimating $B_i(p)$. 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.

3. DISCRIMINATIVE TRAINING

In this section, the discriminative training process is briefly summarized. One major contribution of this study is to develop and implement the already well established discriminative training for achieving optimal accuracy in estimating the state-dependent polynomial coefficients. Let Φ_j, $j = 1, 2, \cdots, \mathcal{K}$, denote the HMM for the j-th class, where $\mathcal{K}$ is the total number of classes. The classifier based on these $\mathcal{K}$ class-models is defined by $\Phi = \{\Phi_1, \Phi_2, \cdots, \Phi_{\mathcal{K}}\}$. The purpose of discriminative training is then to find the parameter set Φ such that the probability of misclassifying all the training tokens is minimized.

Let $g_j(\mathcal{O}, \Phi)$ denotes the log-likelihood associated with the optimal state sequence Θ for the input token $\mathcal{O}$, obtained by using Viterbi algorithm based on the HMM Φ_j for the j-th class. Then, for an utterance $\mathcal{O}$ from class c the misclassification measure $d_c(\mathcal{O}, \Phi)$ is defined as

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

χ denoting the incorrect model with the highest log-likelihood. In this definition, a negative value of $d_c(\mathcal{O}, \Phi)$ corresponds to a correct classification. The definition in eqn. (2) focuses on the comparison between the true model and the best wrong model. A more general form of the misclassification measure using the log-likelihoods from all models can be found in [5]. A loss function with respect to the input token is finally defined in terms of the misclassification measure to be given as

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

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 recognition error rate and is first-order differentiable with respect to all the HMM parameters described by Φ_j, $j = 1, 2, \cdots, \mathcal{K}$.

3.1. Gradient Descent Algorithm

Let ϕ be any parameter of the model Φ. Provided $\Upsilon(\mathcal{O}, \Phi)$ is differentiable with respect to ϕ, the parameter can be adjusted according to

$$\begin{aligned} \hat{\phi} &= \phi - \epsilon \frac{\partial \Upsilon(\mathcal{O}, \Phi)}{\partial \phi} \\ \hat{\phi} &= \phi - \epsilon \underbrace{\Upsilon(\mathcal{O}, \Phi)(\Upsilon(\mathcal{O}, \Phi) - 1)}_{\psi} \frac{\partial d_c(\mathcal{O}, \Phi)}{\partial \phi}. \end{aligned} \tag{4}$$

Here $\hat{\phi}$ is the new estimate of the parameter and ϵ is a small positive constant which monotonically decreases as the iteration number increases. In case of a error-free recognition $\Upsilon(\mathcal{O}, \Phi) \sim 0$ or a complete loss $\Upsilon(\mathcal{O}, \Phi) \sim 1$ the magnitude of ψ is minimum and therefore the change of ϕ becomes small. On the other hand, the magnitude of ψ is maximum when $\Upsilon(\mathcal{O}, \Phi) = 0.5$, indicating the likelihoods for the correct and the best wrong model are equal. Therefore, the training procedure focuses on input tokens which are likely to be misclassified but can be classified correctly after proper adjustment of the model parameters. Each model state is characterized by a multivariate Gaussian density function with diagonal covariance matrices in the form

$$\begin{aligned} b_i(\mathcal{O}_t|\tau_i) \quad = \quad & \frac{(2\pi)^{\frac{-n}{2}}}{|\Sigma_i|^{\frac{1}{2}}} \exp\left(\frac{-1}{2} \left[\mathcal{O}_t - \sum_{p=0}^{P} B_i(p)(t-\tau_i)^p \right]^{Tr} \right. \\ & \left. \Sigma_i^{-1} \left[\mathcal{O}_t - \sum_{p=0}^{P} B_i(p)(t-\tau_i)^p \right] \right) \end{aligned} \tag{5}$$

where $B_i(p)$, Σ_i denote the polynomial means and variances of the i-th state of the model, $(t - \tau_i)$ is the sojourn time in state i at time t and n is the dimensionality. Superscripts Tr,-1 and the symbol || denote the matrix transposition, inversion and determinant respectively. Based on the model j, the optimum state sequence $\Theta^j = \theta_1^j, \theta_2^j, \cdots, \theta_T^j$ for an input token $\mathcal{O} = \mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_T$ with T frames is obtained by means of Viterbi-algorithm. Then, the log-likelihood is given by

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

The simplified gradient descent algorithm is iteratively applied to all training tokens to minimize the loss function during the training process.

3.2. Gradient Computation

By substituting eqns. (2), (5) and (6) in eqn. (4), the gradient calculation of i-th state parameter $B_{i,j}(r)$, $r = 0, 1, \cdots, P$, for the j-th model becomes

$$\begin{aligned} \frac{\partial \Upsilon(\mathcal{O}, \Phi)}{\partial B_{i,j}(r)} &= \psi \frac{\partial d_c(\mathcal{O}, \Phi)}{\partial B_{i,j}(r)} \\ &= \psi \frac{\partial}{\partial B_{i,j}(r)} (-g_c(\mathcal{O}, \Phi) + g_\chi(\mathcal{O}, \Phi)) \\ &= \psi \frac{\partial}{\partial B_{i,j}(r)} \left(- \sum_{t=1}^{T} \log b_{\theta_t^c}(\mathcal{O}_t|\tau_{\theta_t^c}) \right. \\ &= \left. + \sum_{t=1}^{T} \log b_{\theta_t^\chi}(\mathcal{O}_t|\tau_{\theta_t^\chi}) \right) \\ &= \psi_j \sum_{t \in \mathcal{T}_i(j)} \Sigma_{i,j}^{-1} \left[\mathcal{O}_t - \sum_{p=0}^{P} B_{i,j}(p)(t-\tau_i)^p \right] (t-\tau_i)^r \end{aligned}$$

where the adaptive step size is defined as

$$\psi_j \quad = \quad \begin{cases} \psi & if\ j = c \ \ (correct - class) \\ -\psi & if\ j = \chi \ \ (wrong - class) \\ 0 & otherwise \end{cases}$$

and the set $\mathcal{T}_i(j)$ includes all the time indices such that the state index of the state sequence at time t of belongs to state ith in the N-state Markov chain

$$T_i(j) = \{t|\theta_t^j = i\}, \quad 1 \leq i \leq N, \quad 1 \leq t \leq T.$$

The gradient formula for the variances is similar to that of the standard HMM [5] and is presented without derivation

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

where I indicates the $n \times n$ unit matrix and $\tilde{\Sigma}_{i,j}$ is the log-transformed diagonal covariance matrices for easier implementation to make sure that the variances are always remain positive during learning.

4. DATA FITTING EXPERIMENTS

The problem of automatic speech recognition 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 recognition decision. To study this issue, experimental results on fitting the ML and MCE trained trended HMMs to real speech data are described in this section. Once the structure of the trended HMM is chosen, the MCE algorithm discussed in previous section is used to reestimate the ML trained model parameters from the given subset of training data. After the parameters are estimated, the adequacy of the fitted model is checked through diagnostic analysis of the residuals measuring closeness of model fitting to data. $R_t(\Sigma_i)$ is computed according to $R_t(\Sigma_i) = O_t - F_t(i)$. The overall model data-fitting error is then computed by the linear summation of the residual square over the states and over the state-bound time frames.

The two test data sequences for the phones ae and aa are selected from a female speaker of dialect region 1 of TIMIT speech corpus. The raw speech data is in the form of a digitally sampled signal at 16kHz. The mel-frequency cepstral coefficients are computed as in [5] with a frame rate of 10ms. Context independent, ML and MCE trained trended HMMs with three-state left-to-right models are selected for data-fitting analysis. The data-fitting results for the second-order cepstral coefficients C_2 from phones ae are shown here for illustration. C_2 contains information about summation of log energies of low and high-frequency channels subtracting those of mid-frequency channels. Similar results can be obtained for other cepstral coefficients.

Figures 1 and 2 show the results of fitting the same utterance of ae using the "correct" *ae*-model and using the "wrong" *aa*-model, respectively. The top two plots in each figure show the data-fitting results for ML trained standard HMM (left) and linearly trended HMM (right). The bottom two subplots in each figure show the data-fitting results using the MCE trained HMMs with order 0 and 1. In these plots, the solid lines are the actual speech data, O_t, expressed as the C_2 sequence, from one test token. The vertical axis represents the magnitude of C_2 and the horizontal axis is the frame number.

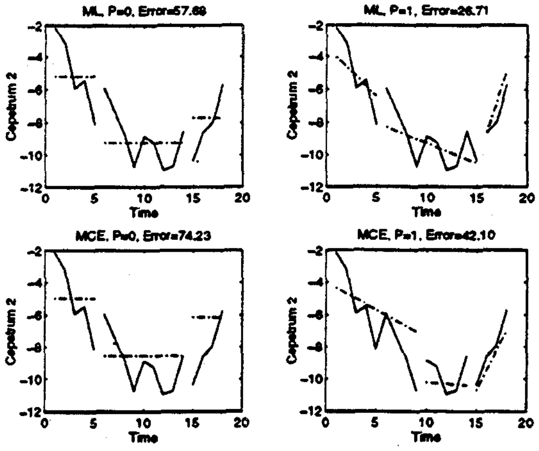

Figure 1: Fitting 3-state *ae*-models to *ae* data sequence

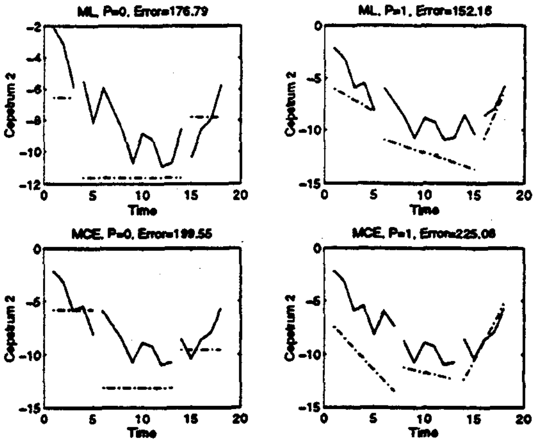

Figure 2: Fitting 3-state *aa*-models to *ae* data sequence

Note that in each plot the two break-points in the otherwise continuous solid lines correspond to the frames where the optimal state transitions occur from state 1 to state 2 and from state 2 to state 3. Superimposed on the four plots as dash-dot lines are the four different fitting functions F_t varying in the order of trend function and training procedure as shown at the top of each plot. To appreciate the role of MCE training in model's discriminability, we examine the data fitting errors in the bottom two plots, one with $P = 0$ and the other with $P = 1$. For the correct model (Figure 1), error reduction in data fitting by incorporating the MCE training goes from 26.71 to 42.10. However, for the purpose of rejecting the wrong model (Figure 2), the MCE method plays a much more significant role — increasing the data fitting error (a measure of the model discrimination power) from 152.16 to as large as 225.08.

Type	ML Method		MCE Method	
of Model	CI Rate	CD Rate	CI Rate	CD Rate
P=0	53.07%	76.62%	63.98%	79.08%
P=1	54.11%	77.07%	69.33%	82.89%

Table 1: TIMIT 39-phone context independent (CI) and context dependent (CD) classification rate using ML (left) and MCE (right) training methods

5. PHONETIC CLASSIFICATION EXPERIMENTS

The standard TIMIT database is chosen for the evaluation experiments. The training subset of the TIMIT database (a total of 462 different speakers) is divided into training-set and test set with no overlapping speakers. The training-set consists of 442 speakers resulting in 3536 sentences and the test set consists of 160 sentences spoken from 20 speakers. The experiments described in this paper aim at classifying the 61 quasi-phonemic labels defined in the TIMIT database folded into 39 classes.

The acoustic analysis used a 21 channel filterbank with approximate Mel spaced filters at a rate of 100Hz. Twelve Mel frequency cepstral coefficients and their differences were formed by taking a discrete cosine transform of the log channel energies (the twelve coefficients exclude the zeroth coefficient which is the log energy). Thus each 10ms speech frame is represented by a vector of 25 components including the delta log energy. Each phone is a left-right, with only self and forward transition, 3-state HMM with Gaussian state observation density. The covariance matrices in all the states of all the models are diagonal. All transition probabilities are uniformly set to 0.5 (all transitions from a state are considered equally likely) and are not trained.

For the MCE approach, the initial models are trained using the ML objective function with 5 iterations of modified Viterbi algorithm based on the most probable state sequence [1]. The parameters of the HMM are then modified by employing the discriminative training method based on MCE optimization as described in Section 3. A complete pass through the training data set is called an epoch. A total of 5 epochs are performed and only the best-incorrect-class is used in the misclassification measure. For context-independent (CI) model, a total of 39 models (39 × 3 = 117 states) are constructed, one for each of the 39 classes intended for the classification task. The procedure outlined in [5] has been adopted to create context-dependent CD models, which results in a total of 1209 states.

Several experiments are run to evaluate the improvement achieved by MCE training. The performance of the phonetic recognizer, organized as the classification rate in terms of the polynomial trend function order (P) is listed in Table 1. In all the experiments MCE training is more stable and achieved an average of 25% classification error rate reduction, uniformly across all types of speech models (both context-dependent and context-independent ones) over the conventional ML-based training. For the CI linear trended HMM, the classification rate is increased from 54.11% (when ML training is used) to 69.33% (when MCE training is used) and yielding an 33.2% error rate reduction. It also represents a 14.85% error rate reduction compared with the corresponding MCE trained standard HMM. We observe that the difference in performance between the $P = 0$ and $P = 1$ HMM is more significant when MCE training are used than the ML training. The best result is achieved by using a combination of linear ($P = 1$) trended HMM and the MCE training algorithm. These performance results are consistent with the data-fitting results and have demonstrated the superiority of the MCE-trained linear trended HMMs over the regular linear trended HMMs.

6. CONCLUSIONS

In this study, the MCE training method using the gradient descent algorithm 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 the poor discriminative ability of conventional ML trained models. The phonetic classification evaluation results show consistent superiority of the MCE approach over earlier ML approach. The greatest error reduction (about 33.2%) has been observed when the first-order trended functions are used in the CI models. The data-fitting results also support the superiority of the MCE approach. We observed that the MCE trained, trended HMM does not fit the speech data (belonging to the *correct* class) as closely as the ML trained counterpart, but it fits the data belonging to the *wrong* class much more poorly than the ML trained model. This apparently gives the mechanism with which the MCE trained model is more capable of discriminating *wrong* sequences from the *correct* sequences than the ML trained model in our phonetic classification experiments.

7. REFERENCES

1. 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, 1994, pp. 507-520.

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

3. B. H. Juang and S. Katagiri, "Discriminative learning for minimum error rate training," *IEEE Trans. Signal Processing*, Vol. 40, 1992, pp. 3043–3054.

4. L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of the IEEE*, Vol.77, 1989, pp. 257-285.

5. C. Rathinavelu and L. Deng, "Use of Generalized Dynamic Feature Parameters for speech recognition: maximum likelihood and minimum classification error approaches," *Proc. IEEE ICASSP*, 1995, Vol. 1, pp. 373-376.