ON-LINE ADAPTATION OF HIDDEN MARKOV MODELS USING INCREMENTAL ESTIMATION ALGORITHMS

V. Digalakis
Dept. of Electronics & Computer Engineering
Technical University of Crete
73100 Chania, Crete, GREECE
vas@telecom.tuc.gr

ABSTRACT

The mismatch that frequently occurs between the training and testing conditions of an automatic speech recognizer can be efficiently reduced by adapting the parameters of the recognizer to the testing conditions. The maximum likelihood adaptation algorithms for continuous-density hidden-Markov-model (HMM) based speech recognizers are fast, in the sense that a small amount of data is required for adaptation. They are, however, based on reestimating the model parameters using the batch version of the expectation-maximization (EM) algorithm. The multiple iterations required for the EM algorithm to converge make these adaptation schemes computationally expensive and not suitable for on-line applications, since multiple passes through the adaptation data are required. In this paper we show how incremental versions of the EM and the segmental k-means algorithm can be used to improve the convergence of these adaptation methods so that they can be used in on-line applications.

1. INTRODUCTION

In statistical approaches to continuous speech recognition (CSR), the model parameters are estimated using several hours of speech data with conditions that match as closely as possible the expected conditions during testing or field deployment. When a mismatch exists, because of speaker, accent, channel or any other type of variability, the recognizer parameters must be adapted to the new conditions. A family of fast adaptation algorithms [1, 5] for continuous mixture density hidden Markov models (HMMs) is based on constrained reestimation of the mixture Gaussians. The observation densities of the speaker-independent (SI) mixture density HMMs have the form:

$$P_{SI}(x_t|s_t) = \sum_{i=1}^{N_\omega} p(\omega_i|s_t) N(x_t; m_{ig}, S_{ig}), \qquad (1)$$

where $N(x_t; m_{ig}, S_{ig})$ denotes the multivariate normal density with mean vector m_{ig} and covariance matrix S_{ig}, and g is the index of the Gaussian codebook used by state s_t. Then, the speaker-adapted (SA) observation densities can be obtained from the SI ones by applying the same affine transformation to the means and covariances of all Gaussians in a particular mixture [1]

$$P_{SA}(x_t|s_t) = \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|s_t) N(x_t; A_g m_{ig} + b_g, A_g S_{ig} A_g^T), \qquad (2)$$

where A^T denotes the transpose of a matrix. In [5], the linear constraint is only applied to the means of the adapted observation densities, which become

$$P_{SA}(x_t|s_t) = \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|s_t) N(x_t; A_g m_{ig} + b_g, S_{ig}). \qquad (3)$$

The algorithms that have previously appeared for the estimation of the transformation parameters are all *batch* adaptation algorithms: all the adaptation data need to be stored and the parameter estimation step, which is usually based on the expectation-maximization (EM) algorithm, requires multiple iterations, that is, multiple passes through the data. The batch nature of these algorithms limits their use in enrollment-type applications: the user must record a number of adaptation sentences, and the system parameters will be adapted off-line before the user can see the benefits from adaptation.

In contrast, most interesting applications require *on-line*, or incremental, adaptation algorithms: the recognizer parameters should be continuously updated after each sentence, without having to store previous utterances. Using on-line adaptation, the recognition performance can improve progressively, as the system is being used. Moreover, since the adaptation process is continuous, the recognizer has the capability to adapt to changing conditions. An algorithm is suitable for on-line adaptation when it involves a single pass through the data, it can be integrated with a speech recognizer easily, and can be used in unsupervised mode, that is, without knowledge of what is being said by the user. An incremental version of the EM algorithm has recently appeared in [6]. The incremental algorithm converges significantly faster than its batch counterpart, which means that it may be possible to use a single pass through the adaptation sentences. This makes the incremental EM algorithm suitable for on-line applications.

To facilitate the integration with an HMM-based speech recognizer, it is preferable to use the segmental k-means training procedure [7], instead of the EM algorithm. This is because Viterbi decoding is the more widespread search technique in HMM-based speech recognizers, and k-means training is based on the most-likely state sequence (MLSS), which is the output of the Viterbi decoder. To implement on-line adaptation in a speech recognizer using the k-means training algorithm, we have developed an incremental variant of this algorithm, analogous to that of the EM algorithm.

2. INCREMENTAL ML ADAPTATION

The EM algorithm can be used to reestimate the transformation parameters in (2), (3) [1, 5]. The number of re-

quired iterations can be determined by measuring recognition performance on independent cross-validation set, and is typically between 5 and 10 iterations. Each iteration consists of a forward-backward pass through the adaptation sentences, followed by a reestimation step. Hence, this approach does not conform to the on-line adaptation requirements.

The incremental version of the EM algorithm has been shown to speed up convergence of ML training of HMMs [3], although the recognition performance did not improve. For on-line adaptation, however, faster convergence is critical, and the incremental EM can be used to achieve it. Both the batch and incremental versions of the EM algorithm can be derived by viewing the EM algorithm as jointly maximizing the quantity [6]

$$F(\tilde{\mathrm{P}}, \theta) = \mathrm{E}_{\tilde{\mathrm{P}}}[\log \mathrm{P}(X, S|\theta)] + \mathrm{H}(\tilde{\mathrm{P}}) \qquad (4)$$

over all distributions $\tilde{\mathrm{P}}(S)$ and parameter sets θ, where $X = (X_1, X_2, \ldots, X_N)$ represents the multiple observations, $S = (S_1, S_2, \ldots, S_N)$ are the corresponding hidden variables, $\mathrm{E}_{\tilde{\mathrm{P}}}$ denotes that the expectation is computed using the distribution $\tilde{\mathrm{P}}$, and $\mathrm{H}(\tilde{\mathrm{P}})$ is the entropy of the distribution $\tilde{\mathrm{P}}$. Assuming independence of observations, the function F can be written as

$$\begin{aligned} F(\tilde{\mathrm{P}}, \theta) &= \sum_{i=1}^{N} F_i(\tilde{\mathrm{P}}_i, \theta), \\ F_i(\tilde{\mathrm{P}}_i, \theta) &= \mathrm{E}_{\tilde{\mathrm{P}}_i}[\log \mathrm{P}(X_i, S_i|\theta)] + \mathrm{H}(\tilde{\mathrm{P}}_i), \quad i = 1, \ldots, N. \end{aligned}$$

The batch and incremental versions of the EM algorithm can be derived by alternating between the maximizations over the distribution $\tilde{\mathrm{P}}(S)$ and the parameter set θ. In the incremental algorithm, we perform each E-step over either a single observation, or a group of observations at a time, and multiple reestimations (M-steps) are performed before a single pass through all observations is completed. In its application to speech, the n-th observation X_n, and its associated hidden variable S_n, consist of the set of observations and their associated state and mixture-mode sequences over a block of utterances (a subset of the total training utterances). When the joint distribution of the observed and hidden variables is a member of the exponential family, the E-step of the EM algorithm reduces to computing the sufficient statistics and the M-step to computing the ML estimates of the parameters θ given these sufficient statistics. If we use $t(X_n, S_n)$ to denote the sufficient statistics of the n-th observation X_n and its associated hidden variable S_n, then the k-th iteration of the incremental EM algorithm for members of the exponential family can be written [6]:

Incremental EM algorithm for the exponential family, k-th iteration:

E-step Select the n-th observation and compute the expected value of the sufficient statistics

$$\begin{aligned} \bar{t}_n^{(k)} &= \mathrm{E}_{\tilde{\mathrm{P}}_n}[t(X_n, S_n)], \\ &\quad \text{where } \tilde{\mathrm{P}}_n(S_n) = \mathrm{P}(S_n|X_n, \theta^{(k-1)}), \\ \bar{t}_j^{(k)} &= \bar{t}_j^{(k-1)}, \quad j \neq n, \\ \bar{t}^{(k)} &= \sum_{i=1}^{N} \bar{t}_i^{(k)} = \bar{t}^{(k-1)} + \bar{t}_n^{(k)} - \bar{t}_n^{(k-1)}. \end{aligned}$$

M-step Set $\theta^{(k)}$ to that value of θ that maximizes the likelihood of the data given $\bar{t}^{(k)}$.

During a single pass through the data a number of iterations of the incremental EM are performed. From the description of the algorithm, we can see that we need to keep a separate copy of the current value of the sufficient statistics for each block of observations. The last requirement makes the algorithm impractical, unless the adaptation sentences are subdivided in a small number of blocks, or a single pass through the data is performed.

In on-line adaptation, we perform a single pass through the data and the need for separate copies of the sufficient statistics is eliminated if the initial values for the statistics of each block are set to zero, $\bar{t}_n^{(0)} = 0$. The incremental EM simply reduces to accumulating the sufficient statistics as we would do normally in the batch version of the algorithm, with the difference being that reestimation (the M-step) is performed multiple times during a single pass through the data, once after visiting each block of observations. Each block of observations can be either a single utterance or multiple utterances.

To use the incremental EM algorithm for on-line adaptation with the constrained estimation schemes described in equations (2) and (3), we must specify the sufficient statistics. For the first method, where the affine constraint is applied to both the means and covariances of the mixture Gaussians, the statistics for each block of observations consist of the following first- and second-order statistics for all multivariate normal densities i of all the Gaussian codebooks g,

$$\begin{aligned} \bar{\mu}_{ig}^{(k)} &= \frac{1}{n_{ig}^{(k)}} \sum_{t, s_t \in \gamma^{-1}(g)} \rho(s_t)\phi_i(s_t)x_t \\ \bar{\Sigma}_{ig}^{(k)} &= \frac{1}{n_{ig}^{(k)}} \sum_{t, s_t \in \gamma^{-1}(g)} \rho(s_t)\phi_i(s_t)(x_t - \bar{\mu}_{ig}^{(k)})(x_t - \bar{\mu}_{ig}^{(k)})^T \\ n_{ig}^{(k)} &= \sum_{t, s_t \in \gamma^{-1}(g)} \rho(s_t)\phi_i(s_t), \end{aligned}$$

where the summation is performed over all the frames t in the block of utterances X_n visited at the current iteration of the algorithm. For simplicity, we have dropped the dependence on the block index n in the equations of the sufficient statistics. The quantity $\rho(s_t) = \mathrm{p}(s_t|X_n, \theta^{(k)})$ is the probability of being at state s_t at time t given X_n and the current parameter estimates $\theta^{(k)}$, and is computed using the forward-backward algorithm. The posterior probability $\phi_i(s_t) = \mathrm{p}(\omega_i|A_g^{(k)}, b_g^{(k)}, x_t, s_t)$ can be computed using Bayes's rule. Given these sufficient statistics, the transformation parameters can be reestimated by solving the following system of equations [1]:

$$\sum_{i=1}^{N_\omega} n_{ig}^{(k)} \Bigg\{ A_g^{(k+1)} - S_{ig}^{-1} \Big[(A_g^{(k+1)})^{-1} (\bar{\mu}_{ig}^{(k)} - b_g^{(k+1)}) - m_{ig} \Big] \cdot (\bar{\mu}_{ig}^{(k)} - b_g^{(k+1)})^T - S_{ig}^{-1} (A_g^{(k+1)})^{-1} \bar{\Sigma}_{ig}^{(k)} \Bigg\} = 0$$

$$b_g^{(k+1)} = \Big[\sum_{i=1}^{N_\omega} n_{ig}^{(k)} (A_g^{(k+1)})^{-T} S_{ig}^{-1} (A_g^{(k+1)})^{-1} \Big]^{-1} \Big[\sum_{i=1}^{N_\omega} n_{ig}^{(k)} \cdot (A_g^{(k+1)})^{-T} S_{ig}^{-1} (A_g^{(k+1)})^{-1} (\bar{\mu}_{ig}^{(k)} - A_g^{(k+1)} m_{ig}) \Big].$$

This is a system of quadratic equations that are decoupled and easy to solve under the assumption of diagonal covariance matrices.

For the second method (3), where the affine constraint is applied only to the means of the Gaussians, then only the first-order sufficient statistics must be computed. This should be expected, since this method reestimates only the means, and not the covariances of the Gaussians. The reestimation equations (M-step) for the transformation parameters can be written in this case (adapted from [5]):

$$\sum_{i=1}^{N_\omega} n_{ig}^{(k)} S_{ig}^{-1} \bar{\mu}_{ig}^{(k)} v_{ig}^T = \sum_{i=1}^{N_\omega} n_{ig}^{(k)} S_{ig}^{-1} W_g^{(k+1)} v_{ig} v_{ig}^T,$$

where $W_g^{(k+1)} = [b_g^{(k+1)} | A_g^{(k+1)}]$, represents the extended matrix of the affine transformation, and $v_{ig} = [1\ m_{ig}^T]^T$ is the extended mean vector.

3. INCREMENTAL K-MEANS ADAPTATION

The segmental k-means training algorithm for HMMs [7] maximizes the joint likelihood of the observations and the associated hidden state sequences $\mathrm{P}(X, S|\theta)$, instead of the maximum likelihood criterion $\mathrm{P}(X|\theta)$ used in EM training. The segmental k-means algorithm performs this optimization iteratively, by alternating between finding the MLSS given the current parameter estimates, and obtaining new estimates for θ by optimizing the joint likelihood of the observations and the MLSS found in the previous step.

The relation between k-means and EM training can be seen by writing the auxiliary function of the EM algorithm, which is maximized at each iteration to give new parameter estimates $\theta^{(k)}$, as

$$\mathrm{E}_{\tilde{\mathrm{P}}}[\log \mathrm{P}(X, S|\theta^{(k)})] = \sum_S \tilde{\mathrm{P}}(S) \log \mathrm{P}(X, S|\theta^{(k)}) \quad (5)$$

where $\tilde{\mathrm{P}}(S) = \mathrm{P}(S|X, \theta^{(k-1)})$. In contrast, the related auxiliary function for the segmental k-means algorithm is

$$\log \mathrm{P}(X, \hat{S}|\theta^{(k)})] = \sum_S \delta(S - \hat{S}) \log \mathrm{P}(X, S|\theta^{(k)}) \quad (6)$$

where $\delta(S - \hat{S}) = 1$ if $S = \hat{S}$, and 0 otherwise. The unobserved state sequence $\hat{S}$ is the MLSS, $\hat{S} = S(\theta^{(k-1)}) = \mathrm{argmax}_S \mathrm{P}(X, S|\theta^{(k-1)})$.

The form of the two auxiliary functions (5) and (6) leads us, in analogy to the modified view of the EM algorithm presented in [6] and summarized in Section 2., to introduce the k-means algorithm as maximizing the joint function

$$F_{SKM}(\tilde{\mathrm{P}}_{SKM}, \theta) = \mathrm{E}_{\tilde{\mathrm{P}}_{SKM}}[\log \mathrm{P}(X, S|\theta)] + \mathrm{H}(\tilde{\mathrm{P}}_{SKM}), \quad (7)$$

where $\tilde{\mathrm{P}}_{SKM}$ is restricted to be a delta function, $\tilde{\mathrm{P}}_{SKM}(S) = \delta(S - \hat{S})$. In this case, $\mathrm{H}(\tilde{\mathrm{P}}_{SKM}) = 0$, and the k-means algorithm maximizes

$$\begin{aligned} F_{SKM}(\tilde{\mathrm{P}}_{SKM}, \theta) &= \sum_{i=1}^{N} F_{i,SKM}(\tilde{\mathrm{P}}_{i,SKM}, \theta), \\ F_{i,SKM}(\tilde{\mathrm{P}}_{i,SKM}, \theta) &= \mathrm{E}_{\tilde{\mathrm{P}}_{i,SKM}}[\log \mathrm{P}(X_i, S_i|\theta)] \\ &= \log \mathrm{P}(X_i, \hat{S}_i|\theta), \quad i = 1, \ldots, N, \end{aligned}$$

since the expectation of the function $\log \mathrm{P}(X_i, S_i|\theta)$ is calculated using the distribution $\delta(S_i - \hat{S}_i)$.

In analogy to the incremental EM algorithm, we can create an incremental version of the segmental k-means algorithm by inserting one reestimation step after the computation of the MLSS $\hat{S}_n^{(k)}$ of each observation (block of utterances) X_n. For the exponential family, the segmentation step of the k-means algorithm reduces to accumulating the sufficient statistics over the states of the MLSS. As in the incremental EM algorithm for the exponential family, the statistics are maintained incrementally, as shown below:

Incremental k-means algorithm for the exponential family, k-th iteration:

Segmentation Select the n-th observation, find the corresponding MLSS

$$\hat{S}_n^{(k)} = \underset{S_n}{\mathrm{argmax}}\, \mathrm{P}(X_n, S_n|\theta^{(k-1)}$$

and update the sufficient statistics:

$$\begin{aligned} \bar{t}_n^{(k)} &= t(X_n, \hat{S}_n^{(k)}), \\ \bar{t}_j^{(k)} &= \bar{t}_j^{(k-1)}, \quad j \neq n, \\ \bar{t}^{(k)} &= \bar{t}^{(k-1)} + \bar{t}_n^{(k)} - \bar{t}_n^{(k-1)}. \end{aligned}$$

Reestimation Set $\theta^{(k)}$ to that value of θ that maximizes the likelihood of the data given $\bar{t}^{(k)}$.

When the incremental k-means algorithm is used for on-line adaptation with the constrained estimation schemes (2) and (3), the sufficient statistics are collected over only the states that belong to the MLSS of the n-th utterance,

$$\bar{\mu}_{ig}^{(k)} = \frac{1}{n_{ig}^{(k)}} \sum_{s_t \in \gamma^{-1}(g) \cap \hat{S}_n^{(k)}} \phi_i(s_t) x_t$$

$$\bar{\Sigma}_{ig}^{(k)} = \frac{1}{n_{ig}^{(k)}} \sum_{s_t \in \gamma^{-1}(g) \cap \hat{S}_n^{(k)}} \phi_i(s_t)(x_t - \bar{\mu}_{ig}^{(k)})(x_t - \bar{\mu}_{ig}^{(k)})^T$$

$$n_{ig}^{(k)} = \sum_{s_t \in \gamma^{-1}(g) \cap \hat{S}_n^{(k)}} \phi_i(s_t).$$

These statistics replace the ones used by the EM-based on-line adaptation adaptation scheme. Given the new sufficient statistics, the transformation parameters for methods (2) and (3) can be reestimated using the same reestimation equations used by the incremental EM algorithm that are given in Section 2..

4. EXPERIMENTAL RESULTS

We evaluated batch and on-line adaptation on the "spoke 3" task of the large-vocabulary Wall Street Journal (WSJ) corpus [4]. The goal of this task is to improve recognition performance for nonnative speakers of American English. Experiments were carried out using SRI's DECIPHER^{TM} speech recognition system configured with a six-feature front end that outputs 12 cepstral coefficients, cepstral energy, and their first- and second-order differences computed from a fast Fourier transform (FFT) filterbank. Cepstral-mean normalization on a sentence basis was performed. We used genonic hidden Markov models with an arbitrary degree of Gaussian sharing across different HMM states as described in [2]. The speaker-independent continuous HMM systems that we used as seed models

for adaptation were gender-dependent. The system for each gender had 12,000 context-dependent phonetic models sharing 500 Gaussian codebooks with 32 Gaussians per codebook. We used the baseline, 5,000-word closed-vocabulary bigram language model provided by the MIT Lincoln Laboratory, and the 1994 development set that consists of 11 speakers. Instead of the 40 standard phonetically rich adaptation sentences, we used 20 WSJ adaptation sentences that are mostly covered by the 5,000-word language model. We did this because we wanted to compare supervised with unsupervised adaptation, and the adaptation sentences should be the same in both cases. The test set consisted of 20 sentences per speaker.

4.1. Batch and Incremental Supervised EM Adaptation

We first compare the batch and incremental versions of the EM algorithm. The speaker-independent, baseline word-error rate for this set was 27.4%. The batch EM adaptation reduces the word-error rate to 17.6% using 5 iterations over the 20 adaptation sentences in supervised mode, that is, with knowledge of the sentence transcriptions. However, the word-error rate for the batch EM increased from 17.6% to 18.8% when the number of iterations decreased from 5 to 1. The faster convergence of the incremental EM regains the performance loss that occurs when a single pass through the adaptation sentences is used, and the best performance for a single iteration of the incremental EM is 17.4% word-error and is achieved by adapting the parameters (i.e. performing the M-step) every 4 sentences. The results comparing the performance of the batch and incremental EM adaptation algorithms are summarized in Table 1.

Adaptation Algorithm	WER (%)
Speaker Independent	27.4
Batch EM, 5 iterations	17.6
Batch EM, 1 iteration	18.8
Incremental EM, 1 iteration Updating Interval (sentences)	
1	17.7
2	17.5
4	17.4
8	18.0
10	18.3

Table 1. Comparison of the batch and incremental EM adaptation algorithms in supervised mode.

4.2. Unsupervised On-line Adaptation

In CSR systems using Viterbi decoding, the recognizer actually produces the most likely state sequence (MLSS) and it is easier to implement on-line adaptation by integrating k-means adaptation rather than Baum-Welch (EM) adaptation. Supervised and unsupervised adaptation can be performed by running the recognizer in different modes: in forced-alignment mode, where the recognizer is restricted to produce the MLSS for a given word string, supervised adaptation can be performed; in recognition mode, where the word string is unknown, the MLSS of the hypothesized string can be used to perform unsupervised adaptation.

In standard batch unsupervised adaptation, the adaptation sentences are recognized using the speaker-independent models, and the hypothesized strings are used by the batch adaptation algorithm. The recognizer models are reestimated at the end of the pass through all the adaptation sentences. The recognition performance is then evaluated on a separate test set. However, when we use incremental k-means adaptation, the models are reestimated more than once during a single pass through the data. This enables a system to use its more recently adapted models to extract the MLSSs of the subsequent adaptation sentences, and improve the quality of these MLSSs. The word error rates of the speaker-independent models, of supervised adaptation (a lower bound on the unsupervised adaptation), the standard batch unsupervised adaptation, and the new incremental k-means unsupervised scheme are summarized in Table 2. The SI word error rate of 27.4% drops to 17.6% when supervised batch adaptation is used over 20 sentences. However, when the transcriptions of the adaptation sentences are not available, the unsupervised adaptation gain is smaller, resulting in a word error rate of 21.6%. The gain of the incremental k-means adaptation algorithm is significantly higher, giving a word error of 19.6% by adapting the models after each sentence.

Adaptation Condition	WER (%)
Speaker-Independent	27.4
Batch EM, Supervised	17.6
Batch EM, Unsupervised	21.6
Incremental k-means, Unsupervised	19.6

Table 2. Word error rates (%) for various adaptation conditions.

Acknowledgments

This work was funded by the U.S. Government under the TRP program and by internal sources. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the Government funding agencies.

REFERENCES

[1] V. Digalakis, D. Rtischev and L. Neumeyer, "Speaker Adaptation Using Constrained Reestimation of Gaussian Mixtures," *IEEE Trans. on Speech and Audio Proc.,* pp. 357–366, September 1995.

[2] V. Digalakis, P. Monaco and H. Murveit, "Genones: Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers," *IEEE Trans. on Speech and Audio Proc.,* pp. 281–289, July 1996.

[3] Y. Gotoh and H. F. Silverman, "Incremental ML Estimation of HMM Parameters for Efficient Training," *Proc. Int'l. Conf. on Acoust., Speech and Signal Processing,* 1996, pp. 585–588.

[4] F. Kubala *et al.,* "The Hub and Spoke Paradigm for CSR Evaluation," *Proc. ARPA Workshop on Human Language Technology,* March 1994.

[5] C. J. Leggetter and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models," *Computer Speech and Language,* pp. 171–185, 1995.

[6] R. M. Neal and G. E. Hinton, "A New View of the EM Algorithm that Justifies Incremental and Other Variants," submitted to *Biometrica*, February 1993.

[7] L. R. Rabiner, J. G. Wilpon and B.-H. Juang, "A Segmental K-means Training Procedure for Connected Word Recognition," *AT&T Tech. J.,* pp. 21–40, May–June 1986.