INCREMENTAL MAP ESTIMATION OF HMMS FOR EFFICIENT TRAINING AND IMPROVED PERFORMANCE

Yoshihoko Gotoh [1] *Michael M. Hochberg* [2] *Daniel J. Mashao* [1] *Harvey F. Silverman* [1]

[1] LEMS, Division of Engineering, Brown University, Providence, RI 02912, USA e-mail: {yg,djm,hfs}@lems.brown.edu
[2] Cambridge University Engineering Department, Cambridge CB2 1PZ, UK e-mail: mmh@eng.cam.ac.uk

ABSTRACT

Continuous density observation hidden Markov models (CD-HMMs) have been shown to perform better than their discrete counterparts. However, because the observation distribution is usually represented with a mixture of multivariate normal densities, the training time for a CD-HMM can be prohibitively long. This paper presents a new approach to speed-up the convergence of CD-HMM training using a *stochastic, incremental* variant of the EM algorithm. The algorithm randomly selects a subset of data from the training set, updates the model using *maximum a posteriori* estimation, and then iterates until convergence. Experimental results show that the convergence of this approach is nearly an order of magnitude faster than the standard batch training algorithm. In addition, incremental learning of the model parameters improved recognition performance compared with the batch version.

1. INTRODUCTION

Continuous density observation hidden Markov models (CD-HMMs) have been shown to perform better than their discrete counterparts. However, because the observation distribution is usually represented with a mixture of multivariate normal densities, the training time for a CD-HMM can be prohibitively long. For example, convergence of standard forward-backward training for a tied-mixture observation HMM developed at LEMS takes several days using multiple Sparc10s [1]. The amount of computation can be reduced by using Viterbi training instead of the full forward-backward approach. However, this approach may cause some degradation in recognition performance. This paper reports on a recent effort to speed-up the training of a CD-HMM without any loss of recognition performance.

Neal and Hinton have investigated variants of the expectation-maximization (EM) algorithm [2]. They reported a substantial speed-up in convergence for a mixture estimation problem using an incremental EM algorithm Fast convergence of an incremental generalized EM algorithm was also noted by Jordan and Jacobs in their work on hierarchical mixtures of experts [3]. The use of incremental training is common in gradient-based learning methods (e.g., back-propagation training of connectionist systems [4]) and has recently been applied to gradient-based training of HMMs [5]. It was hoped that speed improvements could be obtained by applying a similar technique to the training of CD-HMMs.

[1] Partially funded by NSF grant MIP-9120843.
[2] Partially funded by Wernicke ESPRIT Project (BRA 6487).

This paper presents a *stochastic, incremental* variant of the EM algorithm to estimate the parameters of a CD-HMM. The algorithm randomly selects a subset of data from the training set, updates the model parameters based on the subset, and then iterates the process until convergence of the parameters. The random subset selection is done with replacement. This approach is stochastic because the training data is randomly selected. It is considered incremental because the HMM parameters are adjusted before all the training data has been considered. The training strategy contrasts sharply to standard *batch training* where the model is updated only after all the data in the training set are processed. Experimental results show that the convergence of the incremental training algorithm is nearly an order of magnitude faster than batch training. An additional feature — and rather unexpected to be honest — is that the incremental training also improves the recognition performance over the batch version.

2. INCREMENTAL MAP TRAINING

The learning technique presented here is a variation on the *recursive Bayes* approach [6] for performing sequential estimation of model parameters given incremental data. Let $x_1, \ldots, x_T$ be i.i.d. observations and θ be a random variable such that $f(x_t|\theta)$ is a likelihood on θ given by x_t. The posterior distribution of θ is

$$f(\theta|x_1, \ldots, x_t) \sim f(x_t|\theta) f(\theta|x_1, \ldots, x_{t-1}) \qquad (1)$$

where $f(\theta|x_1) \sim f(x_1|\theta)f(\theta)$ and $f(\theta)$ is the prior distribution on the parameters. The recursive Bayes approach results in a sequence of *maximum a posteriori* (MAP) estimates of θ,

$$\hat{\theta}_t = \operatorname*{argmax}_{\theta} f(\theta|x_1, \ldots, x_t). \qquad (2)$$

There is a corresponding sequence of posteriors $f(\theta|x_1, \ldots, x_t)$ which act as the memory for previously observed data. Note that if $f(\theta)$ is a *non-informative prior*, then (2) gives the maximum likelihood (ML) estimate of θ. If the likelihood $f(x_t|\theta)$ is from the exponential family (i.e., a sufficient statistic of fixed dimension exists) and $f(\theta)$ is the conjugate prior, then the posterior $f(\theta|x_1, \ldots, x_t)$ is a member of the same distribution as the prior regardless of sample size t [7]. This implies that the representation of the posterior remains fixed as additional data is observed.

In the case of missing-data problems (e.g., hidden Markov models), the EM algorithm can be used to provide an iterative solution for estimation of the MAP parameters [8]. The iterative EM MAP estimation process can be combined

0-7803-2431-5/95 $4.00

with the recursive Bayes approach. In addition, incorporating (1) and (2) with the incremental EM approach described in [2] (i.e., randomly selecting data from the training set and immediately applying the updated model) has led to the following HMM training algorithm:

1. Initialization: Initialize the update counter $t = 1$. Choose a prior $f(\theta)$ on the HMM parameters and initialize the HMM parameters by $\theta_0 = \text{argmax}_\theta f(\theta)$.
2. Process data: Randomly choose a subset of utterances x_t from the training set. Given θ_{t-1}, run the forward-backward algorithm over x_t and compute the forward and backward recursion terms. Update the posterior distribution (equivalent to the expectation step of accumulating forward-backward terms) by

$$f(\theta|x_1,\ldots,x_t) \sim f(x_t|\theta_{t-1}) f(\theta|x_1,\ldots,x_{t-1}). \quad (3)$$

   Determine the HMM parameters from the updated posterior (the maximization step) by

$$\theta_t = \underset{\theta}{\text{argmax}}\, f(\theta|x_1,\ldots,x_t). \quad (4)$$

3. If no convergence, set $t \leftarrow t+1$ and go to 2.

There are a number of points to note about the above algorithm. First, the sequence of parameters θ_t represents the HMM parameters after the t^{th} update and x_t is the observed acoustic data from a random subset of the training utterances. In their work on MAP estimation of HMM parameters, Gauvain and Lee have presented the expressions for computing the posterior distributions and MAP estimates of CD-HMM parameters [9]. Because the posterior is from the same family as the prior, (3) and (4) are equivalent to the update expressions in [9] and are not repeated here. As a final note, it should be pointed out that there is no proof of convergence for this algorithm. However, the following sections will show empirical results indicating that it does converge to a useful solution.

3. EXPERIMENTS

3.1. Setup and Prior Parameter Generation

The experiments presented here were carried out on a talker-independent, connected-alphadigit recognition task [1]. The vocabulary consists of the American English alphabet ($A \sim Z$) and the digits ($0 \sim 9$). No language model was used. The training (testing) data set contains 3484 (595) utterances from 80 (20) talkers. The typical utterance includes about 15 vocabulary items and has a duration of five seconds.

Standard signal processing was used for the frontend, and three sets of feature vectors were generated from LPC-based mel-cepstral coefficients and energy. The initial parameters of the tied-mixture CD-HMM were derived from a discrete observation hidden semi-Markov model (HSMM) which used a Poisson distribution to model state duration. This model was then converted to a tied-mixture HSMM by simply replacing each discrete symbol with a multivariate normal distribution. Normal means and full covariances were estimated from the training data. Because of the initialization, the system achieved 87.0% correct before further training.

The initial prior distributions were also derived from the training data set. The employed prior distributions were the normal-Wishart distribution for the parameters of the normal distribution and the Dirichlet distribution for the rest of model parameters. The parameters describing the priors were set such that the mode of the distribution corresponded to the initial CD-HMM [7]. The *strength* of the prior (i.e., the amount of observed data required for the posterior to significantly differ from the prior) was determined empirically. A subjective measure of prior strength was used where a very weak prior was (almost) equivalent to a non-informative prior and a very strong prior (almost) corresponded to impulses at the initial parameter values. In the following experiments, trainings were started from a relatively weak prior, unless otherwise noted.

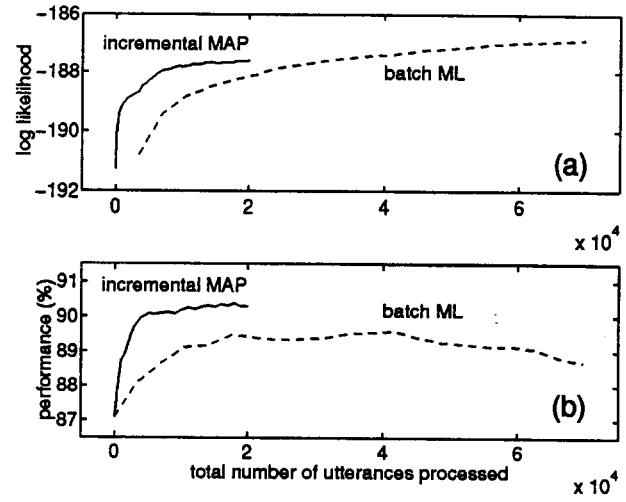

Figure 1. Speed of convergence observed by (a) the log-likelihood and (b) the recognition performance. For both cases, the batch ML training and the incremental MAP training with a subset size of 20 utterances are compared.

3.2. Speed of Convergence

In this experiment, speed of convergence for the incremental MAP training was compared with batch training using the standard ML criterion. The incremental MAP training iterated the model estimation process from the subset size of 20 randomly-selected utterances. Figure 1 shows (a) the log-likelihood and (b) the recognition performance as a function of the total number of utterances processed. Unlike the conventional EM algorithm that guarantees monotonic likelihood improvement, the incremental MAP does not achieve this nice property at each update. However, it is still possible to observe the global trend of the likelihood by computing the running average of the past values. Here, the past 175 likelihoods (approximately equivalent to the number of utterances processed by the batch training for one iteration) were averaged.

According to the log-likelihood in Figure 1(a), convergence of the batch training (dashed line) is not very apparent. On the other hand, the incremental MAP training (solid line) converged when approximately 10,000 utterances were processed. The recognition performance in Figure 1(b) is more interesting. Performance of the batch training reached 89.5% after 6 iterations (approximately 20,000 utterances). It remained at this level and then gradually declined; probably due to overfitting to the training data. The incremental algorithm stabilized after only 4,000 utterances — a factor of five faster than the batch algorithm — to an even higher level of performance. Because the overhead of the incremental MAP processing is negligi-

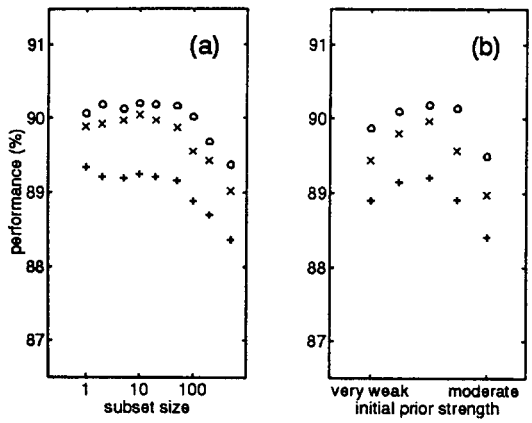

Figure 2. Recognition performance for the incremental MAP training as a function of (a) the subset size and (b) the initial prior strength. "+", "x", and "o" denote performances achieved after 2000, 4000, and 10,000 utterances, respectively.

ble when compared with the batch training, the reduction in the number of utterances required is directly reflected in the training time. The fact that the algorithm converges after 4,000 utterances (200 updates) is not surprising because prior/posterior terms in the MAP estimator become stronger and stronger with each update. As training progresses, each subsequent update has less of an effect.

3.3. Improvement of Performance

In the second set of experiments, the recognition performance of the incremental MAP training was tested as a function of (a) the subset size and (b) the initial prior strength. Figure 2(a) shows performance versus different numbers of randomly-selected utterances (between 1 and 500) in the subsets. For each subset size, the performances achieved after 2000, 4000, and 10,000 utterances are plotted. Although not significant after 10,000 utterances, further improvement ($0.2 \sim 0.4\%$) is still possible. The best performance (90.4%) was obtained when the subset size was 20 and 50. This represents more than an 8% reduction in the error rate from the batch ML training method. Another point to note is that the performance is relatively independent of the subset size if it is small enough (i.e., less than 50). However, the performance gets evidently worse as the subset size grows beyond 100 utterances.

In Figure 2(b), performance versus the different initial prior strengths, from very weak to moderate, were examined for the fixed subset size of 20. As can be seen from the figure, the choice of prior parameter strength does seem to have an effect on the performance. Based on this, investigation into automatic procedures to determine prior strength is under consideration. As a final note on prior strength, this effect may not be significant as more utterances are processed.

3.4. Processing Time

Table 1 shows the actual CPU time for the incremental MAP trainings for the different subset sizes. A very slight increase in processing time is observed as fewer utterances were processed per subset. This is simply because the parameter estimation computation is insignificant compared with the computation of normal mixtures for each utterance. Thus, processing time is nearly proportional to the total number of utterances and independent of the subset size (and, needless to say, independent of the prior strength).

(a)	1	2	5	10	20	50	100	200
(b)	7.8	7.4	7.2	7.1	7.1	7.0	7.0	7.0

Table 1. Processing time for the incremental MAP training on Sparc10-51 workstation: (a) subset size and (b) processing time per 1000 utterances (unit: hours).

subset size	incremental MAP (%)	incremental ML (%)
1	89.9	63.3
10	90.0	78.3
100	89.5	87.1
500	89.0	88.7

Table 2. Recognition performance achieved after 4000 utterances: incremental MAP vs. incremental ML training.

4. VARIATIONS

In this section, several variations to the incremental MAP training are examined.

4.1. Incremental MAP vs. Incremental ML

First, incremental MAP training is compared to incremental ML training[1]. The latter is identical to the former except the ML criterion is used to estimate the parameters. It should be noted that the conventional ML method cannot simply be applied when the subset size is very small; expectations of parameters will not be estimated well from an insufficient amount of data.

Table 2 summerizes the recognition performances for the incremental MAP and the incremental ML trainings after 4000 utterances. As described in Section 3 for the incremental MAP approach, performance reached $89 \sim 90\%$ and higher performance is obtained when the subset size is smaller than 50. For the incremental ML method, performance gets worse with the smaller subset size. Furthermore, performance even decreased from the initial level (87.0%) when the subset size was less than 100.

4.2. Sequential vs. Randomized Data Sampling

So far the incremental MAP training approach has been tested from the perspective that it works as a frequent updating algorithm with a small amount of data in each subset. Here, the effect of randomization in choosing the subset data is examined. Figure 3 compares randomized and sequential sampling of utterances for the incremental MAP training. For both cases, the subset size was fixed to 20. It should be noted that training utterances were arranged in such a way that those from female talkers (1328 utterances) were followed by those from male talkers (2156 utterances). This arrangement may be the cause of the exaggerated defect for the sequential sampling case shown in Figure 3. The reason for the performance going far below the randomized sampling case in the early part of the training was probably due to the strong bias to female talkers. Eventually, the sequential sampling almost caught up to the randomized sampling after the whole training set (3484 utterances) was supplied.

If training utterances were arranged randomly enough, the sequential sampling should have worked as well as the

[1] Incremental ML training was described in [2] and [3]. In our experiment, however, incremental MAP training with the non-informative prior was used instead of incremental ML method, since, as noted in Section 2, they are equivalent.

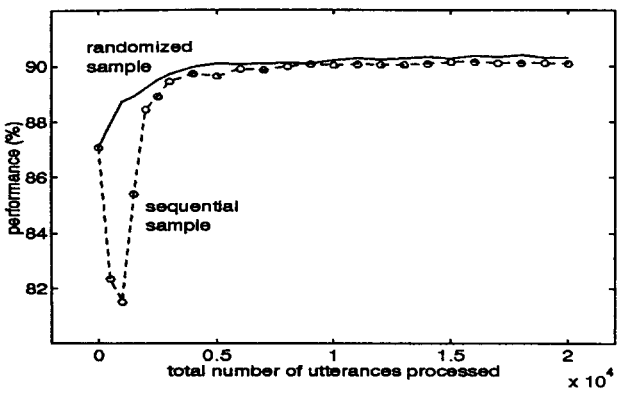

Figure 3. Speed of convergence using the recognition performance criterion. Utterances were sampled randomly and sequentially: for both cases, the incremental MAP training with subset size of 20 utterances was used.

randomized sampling. However, there is no clear picture how the satisfactory randomness can be achieved. The randomized sampling strategy has a good chance of avoiding this problem no matter how the training database has been prepared.

4.3. Refreshing the Prior during Training

This experiment is motivated by the observation that the incremental MAP method is stabilized as the prior terms become stronger after thousands of utterances are processed. It is suspected that further improvement is prevented because the update from the data has less of an effect. It is reported in [2] and [3] that, for some cases, "forgetting" older values is quite useful for attaining better results. Although very heuristic, the idea behind this is that the algorithm improves further by reducing the the effect of the early accumulation which is possibly less accurate than the later ones. The experiment is now under way and the results will be reported at the next opportunity.

5. DISCUSSION

In this paper, a stochastic algorithm was studied using an incremental MAP approach for training a CD-HMM. Convergence of the training algorithm was found to be nearly an order of magnitude faster than the standard batch training. Also, the incremental training had the additional benefit of improving recognition performance.

The incremental algorithm is more efficient because more update steps are taken by frequently estimating the parameters. Consider a simple case; suppose a gradient-based learning method is taken on a parameter space. If the gradient is nearly flat in some area of the parameter space (e.g., a plateau), it will take many steps to traverse the space. Although the estimate of the gradient might be less accurate than the batch training algorithm, the incremental algorithm gets off the plateau faster using more steps per data sample.

There are a number of hypothesized reasons for the improved performance. A possible reason may be due to accumulation of the posteriors for infrequently observed events. This may provide a smoothing of the parameter estimates which reduces the effect of overfitting. Furthermore, the accumulation of likelihoods may be susceptible to precision errors. This becomes more evident as the amount of training data increases. It is difficult to be avoided for the HMM parameter estimation problem where operations such as rounding and/or adding a small number to a large value are quite common, and many of computations need to be done in the log-domain for speed.

It was found that existence of the prior (and, in consequence, use of the MAP estimator) is an important factor when the HMM parameters are estimated frequently from a very small amount of data. Even a very weak prior works far better than the zero prior case, or the incremental ML method. Generally for a gradient method, the first few steps are much larger than the later ones. Intuitively, the "weak prior" lets the steps go in the right direction without enforcing too much restriction. If it is too weak, the first few steps might be very erroneous; if it is stronger than appropriate, it would prevent the steps from being sufficiently large. This idea empirically justifies the use of the MAP estimator over the ML for the task described in this paper.

Finally, it was also found that the improvement of the algorithm is highly dependent on the prior strength. The choice of the prior is very difficult, to say the least, and is mostly a task-dependent issue. However, it is expected the efficiency (and probably improved performance as well) of the incremental MAP approach will remain for a broad spectrum of training scenarios.

REFERENCES

[1] Harvey F. Silverman and Yoshihoko Gotoh. On the implementation and computation of training an HMM recognizer having explicit state durations and multiple-feature-set tied-mixture observation probabilities. Technical Report LEMS Monograph Series: 1-1, Division of Engineering, Brown University, 1994.

[2] Radford M. Neal and Geoffrey E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. *submitted to Biometrika.*

[3] Michael I. Jordan and Robert A. Jacobs. Hierarchical mixture of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, March 1994.

[4] Hervé A. Bourlard and Nelson Morgan. *Connectionist Speech Recognition: A Hybrid Approach.* Kluwer Academic Publishers, Boston, MA, 1994.

[5] Pierre Baldi and Yves Chauvin. Smooth on-line learning algorithms for hidden Markov models. *Neural Computation*, 6(2):307–318, March 1994.

[6] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

[7] Morris H. DeGroot. *Optimal Statistical Decisions.* McGraw-Hill, New York, 1970.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, series B*, 39(1):1–38, 1977.

[9] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, April 1994.