Review on “Efficient ML Training of CDHMM Parameters Based on Prior Evolution, Posterior Intervention and Feedback”

Feng Zheng

Institute for Signal and Information Processing
Mississippi State University
Mississippi State, MS 39762 USA
email: zheng@isip.mstate.edu

ABSTRACT

“Efficient ML Training of CDHMM Parameters Based on Prior Evolution, Posterior Intervention and Feedback” (PEPIF) [1] by Huo *et al.* investigates an efficient maximum likelihood (ML) training procedure for Gaussian mixture continuous density hidden Markov model (CDHMM) parameters. The PEPIF algorithm has been compared against the Baum-Welch algorithm, and the effect of varying various control parameters for the PEPIF algorithm has been investigated. Five experiments for training CDHMMs and evaluations were performed on continuous speech recognition of Mandarin Chinese. The comparison of experiment results nicely verify the improvement achieved by the PEPIF algorithm. From the experiments, the writer demonstrates that the PEPIF algorithm produces a faster increase in likelihood or recognition accuracy than Baum-Welch does, and also offers a 4-fold speed-up over Baum-Welch in the run-time to produce models of given likelihood or accuracy. So this PEPIF algorithm seems to be promising to improve the efficiency of training models, and we will provide a review to verify this.

1. INTRODUCTION

At present, in the estimation methods of Gaussian mixture HMM parameter for speech recognition, the Baum-Welch algorithm remains predominant. However, in order to eventually gain a linear speed of convergence in likelihood scores by Baum-Welch algorithm, speech practitioners need run enough Baum-Welch iterations (say within 10) in HMM training, which has already been a heavy burden with the increasing amount of training data. Furthermore, speech researchers have observed that instead of aiming at a very accurate estimation of the model parameters, obtaining a good rough estimate appears to be more efficient to get a good recognition performance So under this motivation, Huo *et al.* propose to develop an more efficient ML training method which can speedup the convergence and simultaneously increase likelihood score and recognition accuracy.

In this ML training for CDHMM, the algorithm is based on Quasi Bayes (QB) learning framework, which was discussed in detail in [3], and implemented by using the concept of *approximate prior evolution, posterior intervention and feedback* (PEPIF). Three important issues can be expected to achieve the improvement respectively: (1) two different initialization methods which are termed τ-initialization and prior-weight initialization can result in different improvements; (2) instead of formal recursive Bayes learning procedure due to the practical computational difficulties, Huo et al. use an approximate solution: *Quasi-Bayes Learning;* (3) posterior intervention scheme: *a forgetting mechanism* is important to prevent a possible premature convergence of the algorithm.

This paper is well worthy of recommending due to the following nice features: (1) This ML training procedure compares the performance in speed and recognition accuracy against the current popular Baum-Welch training; (2) Some features such as model initialization and posterior intervention are investigated to achieve an improvement in recognition accuracy.

2. METHODOLOGY

Consider a set of parameters of the q-th N-state mixture Gaussian CDHMM’s, $\Lambda = \{\lambda_q\}_{q = 1, \ldots, M}$, where $\lambda_q = (\pi^{(q)}, A^{(q)}, \theta^{(q)})$, where $\pi^{(q)}$ is the

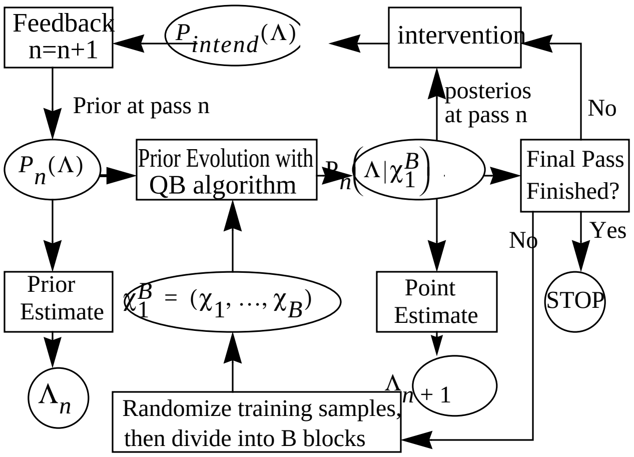

Figure 1: Block of Diagram of PEPIF Procedure

initial state distribution, $A^{(q)} = [a_{ij}^{(q)}]$ is the transition probability matrix, and $\theta^{(q)}$ is the parameter vector composed of mixture parameters $\theta_i^{(q)} = \left\{\omega_{ik}^{(q)}, m_{ik}^{(q)}, r_{ik}^{(q)}\right\}_{k=1,\dots,K}$ for each state i. Furthermore, let χ be a set of training samples. So the problem is then how to efficiently obtain an ML estimate of Λ from χ.

2.1. Quasi-Bayes Learning

Huo et al. claim they use the Bayesian formulation as a tool to derive such an efficient ML training algorithm in [3]. The quasi-bayes procedure is an approximate solution that is motivated by aiming at achieving computational simplicity while still maintaining the flavor of the formal Bayes procedure [4]. One get the approximate MAP estimate $\lambda_{(n)}$ of λ by repeating the following steps.

E-step: Compute

$$R\left(\lambda|\lambda^{\left(n-1+\frac{(m-1)}{M}\right)}\right) = \rho\cdot\log(g(\lambda|\varphi^{(n-1)})) + E\left[\log p(Y_n|\lambda)\middle|\left(X_n, \lambda^{\left(n-1+\frac{(m-1)}{M}\right)}\right)\right]$$

where $0 < \rho \le 1$ is a forgetting factor and $\rho = 1$ means that there is no forgetting.

M-step: Choose

$$\lambda^{(n-1+m/M)} = \arg maxR\left(\lambda|\lambda^{\left(n-1+\frac{(m-1)}{M}\right)}\right)$$

where m = 1, 2,..., M is the iteration index and M is the total iterations performed. Although this is the incremental training procedure to process one utterance at a time. Actually, the Quasi-Bayes learning framework is flexible enough to include the batch or block mode learning as a special case.

2.2. The implementation of PEPIF procedure

Step1: Seed model to initial prior distribution: the first thing starting the CDHMM training process is to seed the models with some initial values. Huo *et el.* investigate two different initialization methods: τ-*initialization* and *Prior-weight initialization*. In a strict Bayesian approach, the hyperparameter $\varphi^{(0)}$ of initial prior pdf $p\{\Lambda|\varphi^{(0)}\}$ is assumed known based on a subjective knowledge about Λ. So in τ-*initialization* method, the hyperparameters $\varphi^{(0)}$ are initialized by a user-defined seed model Λ_{seed} together with a controlling parameter τ to sharper the distribution $p\{\Lambda|\varphi^{(0)}\}$. *Prior-weight initialization* method performs one pass of Baum-Welch training at first, and estimates the hyperparameters according to the statistics from this pass.

Note: here Huo *et al.* cite the reference incorrectly since τ-*initialization* method can not be available in [2].

Step 2: Divide training samples randomly: for every pass of training procedure, the training samples in χ are randomized and divided into B blocks to create a training set $\chi_1^B = \{\chi_1, \chi_2, \cdots, \chi_B\}$, where χ_i's can be viewed as B independent, incrementally obtained set of observation samples.

Step 3: Quasi-Bayes Learning: with $p\{\Lambda|\varphi^{(0)}\}$ and χ_1^B, one can use the *prior evolution* method described in [2] to obtain a posterior pdf $p(\Lambda|\chi_1^B)$, which approximate the posterior distribution by the "closest" tractable distribution $g(\lambda|\varphi^{(n)})$ within the given class, where $\varphi^{(n)}$ denote the updated hyperparameters after observing the sample set χ_n.

Take a point estimate Λ_1 from $p(\Lambda|\chi_1^B)$ and update HMM parameters after this pass of χ.

Step 4: Feedback: by using *intervention scheme,* which apply a forgetting factor to flatten the $p(\Lambda|\chi_1^B)$, one obtain a new pdf $p_{intend}(\Lambda)$, and then *feedback* this pdf to serve as the prior for the next pass of χ. This insight is very important, because according to Huo *et al.* if the posterior pdf is directly feedbacked as a prior for the next pass, the algorithm might converge very quickly to a result which is still far away from the local maximum of the likelihood function. Huo *et al.* point out that the following *exponential refreshing schedule* works quite well:

$$\varepsilon_n = \begin{cases} \varepsilon_0 \times b^{n-1} if(\varepsilon_n < 1) \\ 1 \quad otherwise \end{cases}$$

where ε_n is the refreshing factor for pass n, ε_0 is the initial refreshing factor, and b is the base of the exponential which controls how fast the values of ε_n are increased.

Step 5: N pass for training: take iteration from step 2 to step 4 as a pass of training procedure, and then repeat N passes of this procedure.

3. EXPERIMENT ANALYSIS

The experiments in this paper were performed for continuous speech recognition of Mandarin Chinese on 18224 training utterances (9 females and 9 males) and 1971 testing utterances (1 female and 1 male).

However, the test set only containing 1 female and 1 male is weak, which is hard to convince us that if these experiments were performed on other different speakers the results would be still the same. If the models generated by PEPIF procedure more tend to these two speaker, of course they will achieve better performance. These experiments could have been done by more speakers in the test set.

As can be seen in these experiments, the recognition accuracy is increasing with the likelihood increase, but this is not sure for word error rate (WER). Huo *et al.* could have shown the results in WER as this performance can be more strict to evaluate a speech recognition system.

Huo et al. state in step 2 they need to randomize the sample data and re-divide it to B blocks to obtain a new sample data set for the next pass of training. We doubt the necessity of doing this. He could have some experiment without randomizing sample data for each pass to demonstrate it.

Except for these points, the experiment results nicely match the assumption of Huo *et al.* (1) All the algorithms eventually attain a convergence, especially the first several iterations achieve the most significant likelihood increase. (2) The PEPIF algorithm is more efficient than Baum-Welch algorithm in speed of likelihood increase, and also achieves higher likelihood and recognition accuracy. (3) Prior-weight initialization seems to be a better initialization method than τ-initialization considering the final likelihood and recognition accuracy. (4) The refreshing factor benefits the system to increase final likelihood score.

As Huo *et al.* discuss in the paper, the PEPIF procedure requires the setting of some control parameters which include the initialization parameters and the refreshing factor. These experiments discern the effect of those control parameters on performance, and one can conclude the optimal set of system should be.

Huo *et al.* state that the PEPIF procedure introduces an overhead for each pass: the more the number of data blocks divided, and thus the more updates of parameters are required in one pass. However, as shown in experiments, the biggest overhead is 21% using a block-size of 20 and keeping pruning. Considering PEPIF offers the faster technique for obtaining a set of models of a given likelihood or accuracy. For example, the likelihood of PEPIF at pass 2 is able to be higher than the likelihood of Baum-Welch at pass 10, which indicates a 5-fold speed-up in the number of passes, so by early stopping at pass 2, a 4-fold speed-up can be still achieved by PEPIF procedure.

4. CONCLUSION

This paper presented a review an efficient PEPIF training procedure based on Quasi-Bayes (QB) learning of CDHMM parameters. The paper by Huo *et al.* apply the QB algorithm based on the theory of recursive Bayesian inference. The QB algorithm is designed to update the hyperparameters of the approximate posterior distribution and the CDHMM parameters simultaneously. By further introducing a simple forgetting mechanism to adjust the contribution of previously observed sample utterances to achieve a convergence toward the local maximum of the likelihood function.

According to the review, the optimal implementation of PEPIF algorithm can be concluded briefly: Given all of the training data, one first runs on one Baum-Welch iteration, and thus gets an initial prior pdf estimate; Starting from this initial prior pdf, one can go through the training data again by using Quasi-Bayes (QB) learning framework to update the related parameters. After the pass of the whole training data, one can refresh the posterior pdf and then feedback the refreshed pdf to be the initial prior pdf. The whole process can be repeated until convergence.

As we point out some weakness or suggestion in this paper, especially the testing set needs to be done to make sure that the algorithm can be really verified on this test set. We hope we can see more reports on extensive theoretical analysis of the PEPIF procedure, and more experimental results on the sensitivity of the algorithm to different settings of the control parameters.

REFERENCES

[1] Q. Huo, N. Smith, and B. Ma, “Efficient ML Training of CDHMM Parameters Based on Prior Evolution, Posterior Intervention and Feedback,” *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 1001-1004, Istanbul, Turkey, June 2000.

[2] Y. Gotoh, M. Hochberg, and H. Silerman, “Efficient Training Algorithms for HMM’s Using Incremental Estimation,” *IEEE transactions on Speech and Audio Processing*, Volume 6 Issue: 6, pp. 539-548, November 1998.

[3] Q. Huo and C. Lee, “On-Line Adaptive Learning of the Continuous Density Hidden Markov Model Based on Approximate Recursive Bayes Estimate.” *IEEE Transactions on Speech and Audio Processing*, Volume: 5 Issue: 2, page 161-172, March 1997.

[4] R. Duda, R. Hart, and D. Stork, “Pattern Classification,” NewYork: Wiley, 2000.