Improved Maximum Mutual Information Estimation Training of Continuous Density HMMs

Jing Zheng, John Butzberger, Horacio Franco, Andreas Stolcke

Speech Technology and Research Laboratory, SRI International
333 Ravenswood Ave. Menlo Park, CA 94025 U.S.A
{zj,johnwb,hef,stolcke}@speech.sri.com

Abstract

In maximum mutual information estimation (MMIE) training, the currently widely used update equations derive from the Extended Baum-Welch (EBW) algorithm, which was originally designed for the discrete hidden Markov model (HMM) and was extended to continuous Gaussian density HMMs through approximations. We derive a new set of equations for MMIE based on a quasi-Newton algorithm, without relying on EBW. We find that by adopting a generalized form of the MMIE criterion, the *H*-criterion, convergence speed and recognition performance can be improved. The proposed approach has been applied to a spelled-word recognition task leading to a 21.6% relative letter error rate reduction with respect to the standard Maximum Likelihood Estimation (MLE) training method, and showing advantages over the conventional MMIE approach in terms of both training speed and recognition accuracy.

1. Introduction

Maximum mutual information estimation (MMIE) is a popular discriminative training method for HMM parameter estimation [1]. MMIE has been successfully applied in both small and large vocabulary domains [2] for reducing error rates compared to the traditional maximum likelihood estimation (MLE) method.

MLE training adjusts HMM parameters to increase the likelihood of the word strings corresponding to the acoustic observations. MMIE, by contrast, maximizes the mutual information between word strings and acoustic observations, and thus takes possible competing word hypotheses into account, by trying to decrease the probability of incorrect hypotheses.

The prevalent estimation technique for MMIE training comes from the Extended Baum-Welch Algorithm (EBW) [3], which extends the well-known Baum-Eagon inequality for optimizing rational objective functions. The original EBW algorithm applies only to discrete HMMs. By using various approximations, Y. Normandin [4] extended EBW to HMMs with continuous Gaussian densities, and obtained a set of update formulae that have proved more efficient than the traditional gradient descent algorithm, and have been widely used since.

In this paper, we study the MMIE optimization problem, and find that by using a quasi-Newton approach we obtain update equations for mean and variance estimation that are similar to Normandin's, without relying on EBW. In addition, by using a generalized form of MMIE, we achieve faster convergence speed and higher recognition accuracy.

The paper is organized as follows. First, the MMIE criterion and the EBW algorithm are introduced. We then derive our new update equations for MMIE. Finally, a series of experiments in a spelled-word recognition task are presented, demonstrating the effectiveness of the proposed approach.

2. MMIE criterion and EBW algorithm

MMIE training is based on a criterion that maximizes the mutual information between the training word sequences and observation sequences, instead of the likelihood.

For R training observation sequences $\{O_1, \ldots, O_r, \ldots, O_R\}$, with corresponding transcriptions $\{w_r\}$, the MMIE objective function is given by

$$F(\lambda) = \prod_{r=1}^{R} \frac{p_\lambda(O_r \mid M_{w_r})}{\sum_{\hat{w}} p_\lambda(O_r \mid M_{\hat{w}}) P(\hat{w})} = \frac{N(\lambda)}{D(\lambda)} \tag{1}$$

where M_w is the composite model corresponding to the word sequence w and $P(w)$ is the probability of this sequence as determined by the language model. The summation in the denominator of (1) is taken over all possible word sequences $\hat{w}$ allowed in the task. $N(\lambda)$ and $D(\lambda)$ are the numerator and denominator part of (1) respectively.

The EBW algorithm as applied to Gaussian mixture HMMs gives the following update equations for the mean of a Gaussian g at each particular dimension μ_g, and the corresponding variance σ_g^2 (assuming diagonal covariance matrices):

$$\hat{\mu}_g = \frac{\left\{\theta_g^{\text{num}}(O) - \theta_g^{\text{den}}(O)\right\} + D\mu_g}{\left\{\gamma_g^{\text{num}} - \gamma_g^{\text{den}}\right\} + D} \tag{2}$$

$$\hat{\sigma}_g^2 = \frac{\left\{\theta_g^{\text{num}}(O^2) - \theta_g^{\text{den}}(O^2)\right\} + D(\sigma_g^2 + \mu_g^2)}{\left\{\gamma_g^{\text{num}} - \gamma_g^{\text{den}}\right\} + D} - \hat{\mu}_g^2 \tag{3}$$

In these equations, the $\theta_g(O)$, $\theta_g(O^2)$ and γ_g are defined as

$$\theta_g(O) = \sum\nolimits_t \gamma_g(t) O_t \tag{4}$$

$$\theta_g(O^2) = \sum\nolimits_t \gamma_g(t) O_t^2 \tag{5}$$

$$\gamma_g = \sum\nolimits_t \gamma_g(t) \tag{6}$$

where $\gamma_g(t)$ is the occupancy probability of the Gaussian *g* at frame *t*. The superscripts "num" and "den" refer to the models corresponding to the transcription and the recognition model for all word sequences, respectively.

The value *D* in the update equations (2) and (3) determines training speed as well as stability of the algorithm. If the value is set too large, then training is very slow (though stable); if it is too small, the objective function could fail to increase on each iteration. An often-used method for choosing the lower bound of *D* is to ensure that all variances remain positive. In [2], using per-Gaussian values of *D* is found to be better in terms of convergence speed than using a more global value of *D*. For each Gaussian, *D* was set at the maximum of (i) twice the value necessary to ensure positive variance updates for all dimensions of the Gaussian, or (ii) the denominator occupancy γ_g^{den}.

The mixture weight values can be updated by directly using the original EBW algorithm; several improved approaches are discussed in [4]. Prior research found that for HMMs with Gaussian mixture models, the means and variances play a much more important role than the mixture weights for MMIE training; this has also been verified in our own experiments. Thus, we will focus on Gaussian estimation in the remainder of the paper.

In [5], a generalized form of MMIE, the *H*-criterion, is proposed, which aims at maximizing the following objective function:

$$F_h(\lambda) = \prod_{r=1}^{R} \frac{p_\lambda(O_r \mid M_{w_r})}{\left[\sum_{\hat{w}} p_\lambda(O_r \mid M_{\hat{w}}) P(\hat{w})\right]^h} = \frac{N(\lambda)}{D(\lambda)^h} \qquad (7)$$

Both MLE and MMIE can be understood as special cases of the *H*-criterion, corresponding to *h*=0 and *h*=1, respectively. In this study, we derive the update formulae for the *H*-criterion through a novel approach.

Although MMIE training has proved to be effective in many cases, it is still not well understood how the increase of mutual information is related to the reduction of error rate, the actual objective function for speech recognition. In this paper, we experimentally study the influence of the parameter *h* on the performance of trained models in terms of word error rate (WER). Although [0, 1] is commonly regarded as the normal range of value for *h*, we find the optimal value for *h* in terms of minimal error rate lies in [1, +∞] in our experiments.

3. Update equations from an improved gradient approach

During the few years after MMIE training was first proposed, gradient descent (GD) algorithms were the common approach to estimating HMM parameters. However, the convergence speed of GD algorithm was often found to be very slow, so that once the EBW algorithm was known, the GD algorithm was seldom used. The EBW algorithm derives from an extended version of the well-known Baum-Eagon inequality for optimizing rational objective functions of polynomials, and thus originally it was applicable only to the discrete HMM. Normandin proposed an approximation approach that maps the continuous Gaussian densities to the discrete distributions in the limit condition, and thus heuristically developed the update equation for means and variances of Gaussian densities, which are widely used today [4]. However, the derivation used several approximations that are not desirable from a theoretical point of view.

In this paper we will show that update equations similar to Normandin's can be obtained by a refinement of the traditional GD algorithm, without relying on the inequality that was applicable only to the discrete case.

Optimizing the objective function $F_h(\lambda)$ in (1) is equivalent to optimizing its logarithm $\log(F_h(\lambda))=\log(N(\lambda))-h\log(D(\lambda))$. To make the notation clearer in the following equation, we use λ_0 to denote the original model that generates the counts information. For each particular dimension of the mean and variances for Gaussian *g*, we can easily get the partial derivatives:

$$\begin{aligned}\left.\frac{\partial(\log(F_h(\lambda))}{\partial\mu_g}\right|_{\lambda_0} &= \frac{1}{N(\lambda_0)}\frac{\partial(N(\lambda_0)}{\partial\mu_g} - \frac{h}{D(\lambda_0)}\frac{\partial(D(\lambda_0)}{\partial\mu_g} \\ &= \sigma_g^{-2}\left[\theta_g^{\text{num}}(O) - h\theta_g^{\text{den}}(O) - (\gamma_g^{\text{num}} - h\gamma_g^{\text{den}})\mu_g\right]\end{aligned} \qquad (8)$$

$$\begin{aligned}\left.\frac{\partial(\log(F_h(\lambda))}{\partial\sigma_g^2}\right|_{\lambda_0} &= \frac{1}{N(\lambda_0)}\frac{\partial(N(\lambda_0)}{\partial\sigma_g^2} - \frac{h}{D(\lambda_0)}\frac{\partial(D(\lambda_0)}{\partial\sigma_g^2} \\ &= \frac{\sigma_g^{-4}}{2}\left[\theta_g^{\text{num}}(\sigma^2) - h\theta_g^{\text{den}}(\sigma^2) - (\gamma_g^{\text{num}} - h\gamma_g^{\text{den}})\sigma_g^2\right]\end{aligned} \qquad (9)$$

where

$$\begin{aligned}\theta_g(\sigma^2) &= \sum\nolimits_t \gamma_g(t)(O_t - \mu_g)^2 \\ &= \theta_g(O^2) - 2\theta_g(O)\mu_g + \gamma_g\mu_g^2\end{aligned} \qquad (10)$$

It is known that using the gradient of $-\log F_h(\lambda)$ directly results in slow convergence. However, we found empirically that descending along the following directions leads to much better performance:

$$\hat{\boldsymbol{\mu}}_{\mathbf{g}} = \boldsymbol{\mu}_{\mathbf{g}} - \delta\boldsymbol{\Sigma}_{\mathbf{g}}\nabla\boldsymbol{\mu}_g \qquad (11)$$

$$\hat{\boldsymbol{\sigma}}_{\mathbf{g}}^{\mathbf{2}} = \boldsymbol{\sigma}_{\mathbf{g}}^{\mathbf{2}} - 2\delta(\boldsymbol{\Sigma}_{\mathbf{g}})^2\nabla\boldsymbol{\sigma}_{\mathbf{g}}^{\mathbf{2}} \qquad (12)$$

where $\boldsymbol{\mu}_{\mathbf{g}}$ is the vector of means, and $\boldsymbol{\sigma}_{\mathbf{g}}^{\mathbf{2}}$ is the vector of variances for the Gaussian *g*; $\boldsymbol{\Sigma}_{\mathbf{g}}$ is a diagonal matrix, in which diagonal elements are the variances of the Gaussian *g*; $\nabla\boldsymbol{\mu}_g$ and $\nabla\boldsymbol{\sigma}_{\mathbf{g}}^{\mathbf{2}}$ are the gradients of $-\log F_h(\lambda)$ in the spaces of $\boldsymbol{\mu}_{\mathbf{g}}$ and $\boldsymbol{\sigma}_{\mathbf{g}}^{\mathbf{2}}$ respectively; and δ is the step length.

Faster convergence might be explained by the principle of the quasi-Newton algorithm. First, both $\boldsymbol{\Sigma}_{\mathbf{g}}$ and $2\boldsymbol{\Sigma}_g^2$ are symmetric and positive definite, which makes the resulted descent direction valid according to the well-known theory of Generalized Probabilistic Descent (GPD) [6]. Furthermore, they are approximately proportional to the inverse of the second-order derivatives matrix (the Hessian). Considering the fact that a small change in the model causes only very small changes in alignments, with proper approximation we can get

$$\nabla(\nabla\mu_g) \approx (\gamma_g^{\text{num}} - h\gamma_g^{\text{den}})(\Sigma_g)^{-1} \qquad (13)$$

$$\nabla(\nabla\sigma_g^2) \approx \frac{1}{2}(\gamma_g^{\text{num}} - h\gamma_g^{\text{den}})(\Sigma_g)^{-2} \qquad (14)$$

Thus, the direction of descent given by (8) is close to the Newton direction, which produces faster convergence speed than the gradient direction does.

Using (8) we get the following update formulae for each particular dimension of Gaussian g:

$$\hat{\mu}_g = \frac{\theta_g^{\text{num}}(O) - h\theta_g^{\text{den}}(O) + D\mu_g}{\gamma_g^{\text{num}} - h\gamma_g^{\text{den}} + D} \qquad (15)$$

$$\hat{\sigma}_g^2 = \frac{\theta_g^{\text{num}}(\sigma^2) - h\theta_g^{\text{den}}(\sigma^2) + D\sigma_g^2}{\gamma_g^{\text{num}} - h\gamma_g^{\text{den}} + D} \qquad (16)$$

where $\theta_g^{\text{num}}(\sigma^2)$ and $\theta_g^{\text{den}}(\sigma^2)$ are defined by (10); $D = \delta^{-1} - (\gamma_g^{\text{num}} - h\gamma_g^{\text{den}})$, a function of step length *δ*. It is easy to see that, the larger the value of *D*, the smaller the step length *δ*, the slower (but more stable) the convergence; the smaller the *D*, the larger the step length *δ*, the faster (but more unstable) the convergence. To select a proper *D* (or equally a step length *δ*), we take a strategy similar to the one that was used in [2], that is, the maximum of the two values: (i) $h\gamma_g^{\text{den}}$, and (ii) twice the minimum value that makes the variances of the Gaussian positive in all dimensions.

Comparing (15) and (16) with (2) and (3), we see that they are very similar except for two differences. The first one is the parameter *h*, which gives a weight to the denominator count with respect to the numerator count. We will see later that tuning *h* can result in improvements in MMIE training. The second difference lies in the variance re-estimation. In fact we find that this difference causes little change in performance. Incidentally, with (16) it is straightforward to determine the value of *D*, while (3) requires solving a quadratic equation.

4. Experimental results

To compare the proposed update equations (15 and 16) with the classical ones (2 and 3), we evaluated both in a spelled-word recognition task, which aims at recognizing the letters from the continuously spoken spelled words over a telephone line.

The system configuration is as follows. The front-end analysis is performed on a 32 ms window with a frame rate of 16 ms for computing a 39-dimension MFCC feature vector: C1~C12 plus C0 with their first- and second-order derivatives. To compute the cepstral coefficients, we use 24 filter banks ranging from 100 Hz to 3760 Hz. The feature's mean subtraction and variance normalization are performed at the speaker level. The acoustic models are gender-independent within-word triphone HMMs, with a 3-state left-to-right topology and Gaussian Mixture Models as output densities. The Gaussian codebooks are tied at (context-independent) phone state level, with each codebook having 32 Gaussians. This acoustic model is compactable with potential vocabulary extension. To deal with real-world data, we use separate noise filler models in addition to the pause model.

The training and test data are drawn from two corpora: MACROPHONE and PARTYLINE. Both were collected through telephone lines at SRI International; MACROPHONE is publicly available. The spelled-word data is only a subset of these two corpora. We used all of MACROPHONE and part of PARTYLINE for training; the rest of PARTYLINE data was reserved for testing. Test and training sets are selected so as to avoid speaker overlap. The training set contains 13,131 waveforms from 6,756 calls; the test set contains 1,133 waveforms from 581 calls. An unconstrained self-loop grammar is used in recognition as well as in MMIE training to collect the denominator counts.

Before MMIE training, we carried out multiple iterations of MLE training until saturation, such that additional iterations of MLE did not further reduce the letter error rate (LER). We then took this model as the starting point for MMIE training. Table 1 lists the results of the final model from MLE training.

Table 1: Final Letter Error Rate (LER) with MLE training. SUB, DEL, INS represent substitute, deletion and insertion error rate respectively.

Method	SUB	DEL	INS	LER
MLE	8.18	1.57	1.40	11.14

4.1. First iteration of MMIE training

We started MMIE training with the initial model from MLE. We did two groups of experiments: in one group, we used update equations (15) and (16), hereafter denoted by N_MMIE; in the other group, we used update equations (2) and (3), denoted by C_MMIE. To make the comparison more focused on Gaussian estimation, we fixed mixture weights and HMM transition probabilities at their initial values from MLE during the MMIE training iterations. In research by others [2] as well as our own, mean and variance re-estimation play a dominant role for HMMs with GMM densities.

In the first iteration of MMIE training, we investigated the influence of different values of *h* in (15) and (16), and we plot the LER curve as a function of *h*.

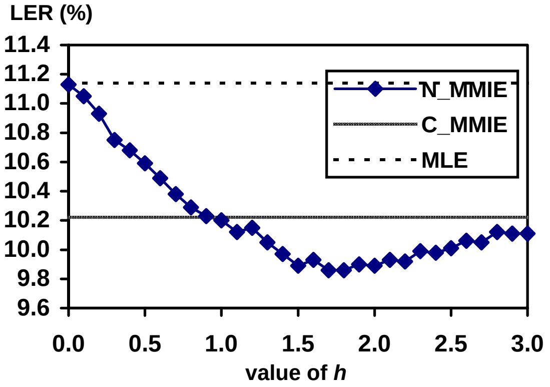

Fig.1 Influence of value of h in the first training iteration: LER versus *h*.

We also investigated larger values of *h*, with the result that when *h* is large enough, the LER becomes constant, since only the denominator part has influence on the model parameters. Table 2 lists the results for some typical values of *h*.

From Figure 1 and Table 2 we can see that, for N_MMIE, which uses our proposed update equations, the letter error rate reaches a minimum at *h*=1.7. This LER is 3.2% (relative) lower than that of C_MMIE, which uses the classical formulas (2) and (3). It is also interesting to look at other typical values of *h*. When *h*=0, it is easy to see that *D*=0, according to our step length selection criterion. Thus, this case is identical to a standard MLE iteration, except that only means and variances are re-estimated. This confirms that there is almost no further improvement from additional MLE iterations. When *h*=1.0, N_MMIE is almost identical to C_MMIE, except for the small difference in variance estimation. The results show that here N_MMIE leads by a margin of 0.02%. When $h \to +\infty$, only the denominator counts play a role in re-estimation, which clearly could not reach optimal results. However, this result is still better than that of MLE.

Table 2: Recognition performance at different values of *h*.

Method	*H*	SUB	DEL	INS	LER
MLE	N/A	8.18	1.57	1.40	11.14
C_MMIE	N/A	7.55	1.57	1.10	10.22
N_MMIE	0	8.22	1.55	1.36	11.13
N_MMIE	1.0	7.55	1.57	1.08	10.20
N_MMIE	1.7	7.33	1.55	0.98	9.86
N_MMIE	$+\infty$	7.81	1.57	1.10	10.48

4.2. More training iterations

We trained the model with more iterations using both N_MMIE and C_MMIE. For N_MMIE, we fixed the parameter *h* at 1.7, which proved to be approximately optimal for later iterations as well. Table 3 compares the results of N_MMIE and C_MMIE in multiple iterations.

Table 3: LER from multiple MMIE iterations.

Iteration No.	N_MMIE (*h*=1.7)	C_MMIE
0 (MLE)	11.14	11.14
1	9.86	10.22
2	9.00	9.65
3	8.80	9.30
4	9.65	9.04
5	-	9.11

Table 3 shows that N_MMIE reaches the best performance at the third iteration, which reduces LER by 21.6% (relative) with respect to MLE. C_MMIE reaches a LER minimum after four iterations, which reduces the LER by 18.9%. It is easy to see that the training speed of N_MMIE is faster than that of C_MMIE. This point can be important for large vocabulary tasks, where MMIE training, especially the denominator counts collection, can be very time consuming. It should be noted that after a few iterations, both N_MMIE and C_MMIE begin to degrade the performance, while still increasing the objective function. This indicates that additional iterations lead to overfitting of the training data, part of which are not from the same corpus that comprises the testing data. As we noticed, in N_MMIE the error rate increase from the third iteration to the fourth iteration is large, which makes the stopping point too sensitive. This could be improved by using gradually decreasing step length, that is, increasing *D*, in the later iterations, as the GPD theory suggests [6].

5. Conclusion

We have derived update formulae for continuous density HMMs for a generalized form of MMIE training based on the *H*-criterion. By using an improved gradient descent algorithm, we obtained update equations similar to the ones derived from EBW. Experiments show that tuning the parameter *h* in the *H*-criterion can lead to better training speed and further recognition performance improvement over the currently widely used MMIE training scheme.

Reference

[1] L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer, "Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition," *Proc. Int'l Conf. Acoust. Speech Signal Processing*, pp. 49-52. Tokyo, 1986.

[2] P.C. Woodland and D. Povey, "Large Scale MMIE Training for Conversational Telephone Speech Recognition," *Proc. Speech Transcription Workshop*, College Park, 2000.

[3] P. S. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo, "An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems," *IEEE Trans. Information Theory*, 37(1):107-113, 1991.

[4] Y. Normandin, *Hidden Markov Models, Maximum Mutual Information Estimation and the Speech Recognition Problem*, Ph.D. Thesis, McGill University, Montreal, 1991.

[5] P. S. Gopalakrishnan, D. Kanevsky, A. Nadas, D. Nahamoo, and M. A. Picheny, "Decoder Selection Based on Cross-Entropies," *Proc. Int'l Conf. Acoust. Speech Signal Processing*, pp. 20-23, 1988.

[6] S. Amari, "A Theory of Adaptive Pattern Classifiers," *IEEE Trans. Electronic Computers*, EC-16(3): 299-307, 1967.