A Fast Discriminative Training Algorithm for Minimum Classification Error

B. Silva, H. Mendes, C. Lopes, A. Veiga and F. Perdigão

[1] Department of Electrical and Computer Engineering, FCTUC, University of Coimbra
Instituto de Telecomunicações, Polo II, University of Coimbra, Portugal
{markexilva, maizena.mendes}@gmail.com, {calopes, aveiga, fp}@co.it.pt

Abstract

In this paper a new algorithm is proposed for fast discriminative training of hidden Markov models (HMMs) based on minimum classification error (MCE). The algorithm is able to train acoustic models in a few iterations, thus overcoming the slow training speed typical of discriminative training methods based on gradient-descendent. The algorithm tries to cancel the gradient of the objective function in every iteration. Re-estimation expressions of the HMM parameters are derived. Experiments with triphone and word models show that the proposed algorithm always achieves much better results in a single iteration than MCE, MMI or MPE do over several iterations.

Index Terms: Speech recognition, discriminative training, hidden Markov models.

1. Introduction

The conventional HMM training method is based on Maximum Likelihood (ML). However, it is well known that discriminative training methods outperform ML. Different discriminative training criteria have been successfully tested, namely maximum mutual information (MMI) [1], minimum phone error (MPE) [2], and minimum classification error (MCE) [3,4]. The MCE criterion is especially attractive because it minimizes a function that is directly related to the performance of the recognizer, the classification error rate, using N-best hypotheses. Recently a method using the extended Baum-Welch (EBW) algorithm was proposed [5] but this works only for the 1-best hypothesis. Discriminative training approaches use iterative optimization algorithms to estimate model parameters and convergence speed therefore plays an important role in training. The conventional optimization method used in MCE is based on a gradient descent (GD) technique called Generalized Probabilistic Descent [4]. This method is easy to implement and presents effective results, but the training speed is slow and it is difficult to estimate learning rates. These limitations have led to a need for new training algorithms that perform faster than GD-based algorithms.

Another problem encountered in the conventional objective function used in MCE is that the sigmoidal loss function saturates when gross errors occur and gradient methods cannot subsequently improve on these errors. In this paper we present a new discriminative training objective function which solves this problem.

A new fast discriminative training algorithm for HMMs based on MCE is also introduced. This algorithm is an extension of continuous speech recognition using HMMs from the method proposed in [6] for multiple-category classification problems. We have called it fast minimum error training (FMET). In order to compare the performance of the proposed training algorithm we have implemented MCE using the new objective function and one of the fastest GD based algorithms, resilient backpropagation (Rprop) [7]. We also compare the performance with maximum mutual information (MMI) and minimum phone error (MPE) algorithms.

The rest of the paper is organized as follows: in Section 2 the new objective function is defined and FMET is derived. Experimental results comparing FMET, Rprop, MMI and MPE are presented in Section 3. Finally, Section 4 presents some conclusions and guidelines for future work.

2. Fast Discriminative Training for HMMS

In this section a new objective function is introduced for discriminative training with MCE and the HMM parameter re-estimation formulas are derived for the proposed training algorithm, FMET.

2.1. The MCE objective function

The objective function used in MCE is defined as

$$J = \sum_{n=1}^{N_u} l\left(d^{(n)}\right) \tag{1}$$

where N_u is the number of training utterances and $l(d^{(n)})$ is a smooth loss function that emulates the zero-one recognition error count. Typically a sigmoid is used:

$$l(d^{(n)}) = \frac{1}{1+e^{\left(-\lambda d^{(n)}\right)}}\,. \tag{2}$$

In these expressions $d^{(n)}$ is the misclassification measure between the score of the labelled HMM sequence $W_{lab}^{(n)} = \left\{w_{n,1}^{lab},...,w_{n,T^{(n)}}^{lab}\right\}$ and a generalized mean (softmax) over the scores of the competing N_{Best} HMM sequences, $W_k^{(n)}$, for the n[th] utterance:

$$d^{(n)} = L \cdot \log\left(\frac{1}{N_{Best}} \sum_{\substack{k=1 \\ k\neq lab}}^{N_{Best}} e^{\alpha g(W_k^{(n)})}\right)^{\frac{1}{\alpha}} - g(W_{lab}^{(n)}), \tag{3}$$

$g(W)$ is a discriminant function computed as the log likelihood of the sequence of acoustic observations,

$$X^{(n)} = X_{1:T^{(n)}}^{(n)} = \{\underline{x}_1^{(n)},...,\underline{x}_{T^{(n)}}^{(n)}\}\,, \tag{4}$$

given the best state alignment of the HMM sequence W, which is computed with the Viterbi algorithm.

In (3) we introduce a scalar L, multiplied by the softmax term which controls the relative importance of true sequence $W_{lab}^{(n)}$ and competing sequences $W_k^{(n)}$, k=1..N_{Best}. This scalar is the key point of this algorithm: setting L=0 implies ML

training and L=1 corresponds to the classical MCE. It can be even greater than 1, if the stochastic restrictions of the HMM parameters remain verified, as explained below.

2.2. The method

The proposed method aims to cancel the gradient of the objective function, J, in every iteration. This is a necessary condition in order to minimize J. Setting the gradient $\nabla J = 0$ leads to a set of re-estimation expressions for the HMM parameters. However, there are restrictions that these parameters must obey: the transition probabilities and mixture weighs must be positive and add up to 1 and the covariance matrices must be non-negative definite. These conditions can easily be set by an appropriate choice of the scalar L. It turns out that this scalar has the same role as the learning rate in the GD methods. GD methods benefit from a different learning rate per parameter. Almost the same applies here: instead of a single global L, we found it advantageous to have an L-scalar for each HMM state and for each transition from that state. This turns into a slight modification of the decoding process of the competing sequences $W_k^{(n)}$, with the Viterbi algorithm, that becomes

$$g(W_k^{(n)}) = \sum_{t=1}^{T^{(n)}} \left(L_{q_t}^{(w_{n,t}^k)} \log\left(b_{q_t}^{(w_{n,t}^k)}(\underline{x}_t^{(n)}) \right) + L_{q_{t-1}}^{(w_{n,t}^k)} \log\left(a_{q_{t-1},q_t}^{(w_{n,t}^k)} \right) \right). \quad (5)$$

In this equation, $Q = \{q_1, q_2, ..., q_T\}$ is the best state sequence given by Viterbi alignment over all models in the utterance n; $a_{ij}^{(h)}$ is a transition probability and $b_j^{(h)}(x)$ is the pdf of state j belonging to HMM $h = w_{n,t}^k$ found at each frame t for each utterance n and competing sequence k. The two types of L-scalars are $L_j^{(h)}$ for each state j and $L_i^{(h)}$ for the transitions from each state i, $a_{ij}^{(h)}$. These weightings will help to ensure the statistical constraints of the HMM parameters. As indicated in [6], we cannot guarantee convergence at each iteration; however, it has been shown experimentally that this method produces much better results than the GD algorithm.

Although the sigmoid loss function is suitable for error counting, its gradient approaches zero for an utterance with a large value of $d^{(n)}$, meaning that the utterance is misclassified. This is not suitable for approaches which optimize (1) through differential methods, because these utterances will make an insignificant contribution to the gradient. In order to solve this limitation we propose the following function

$$l(d^{(n)}) = \log\left(1 + e^{\lambda d^{(n)}}\right). \quad (6)$$

This function approaches zero when $d^{(n)}$ is negative (utterance n is correctly recognized) and approaches $\lambda d^{(n)}$ when $d^{(n)}$ is positive (utterance n is incorrectly recognized). This overcomes the sigmoid limitation, especially at the first steps of the algorithm where a large $d^{(n)}$ does not mean necessarily an outlier, due to a mislabelled utterance for example.

2.3. Estimation formulas

In this section the re-estimation equations are derived for the HMM parameters as well as the limits for the L-scalars. It is assumed that the state pdf is a Gaussian mixture,

$$b_j^{(h)}(\underline{x}) = \sum_{m=1}^{M_j} c_{jm} \cdot b_{jm}(\underline{x}), \quad (7)$$

where M_j is the number of mixture components of state j within HMM h, c_{jm} is the weight of the mth mixture component and $b_{jm}(\underline{x})$ is a Gaussian pdf

$$b_{jm}(\underline{x}) = \frac{1}{\sqrt{|2\pi \boldsymbol{\Sigma}_{jm}|}} e^{\left(-\frac{1}{2}(\underline{x}-\underline{\mu}_{jm})^T \boldsymbol{\Sigma}_{jm}^{-1}(\underline{x}-\underline{\mu}_{jm})\right)}, \quad (8)$$

where $\underline{\mu}_{jm}$ and $\boldsymbol{\Sigma}_{jm}$ are respectively the mean vector and the covariance matrix of the mth mixture component of state j. It is also assumed that the covariance matrices are diagonal. The gradient of the objective function is

$$\nabla J = \sum_{n=1}^{N_u} \psi^{(n)} \cdot \nabla d^{(n)}, \quad (9)$$

where

$$\psi^{(n)} = \frac{\lambda e^{\lambda d^{(n)}}}{1 + e^{\lambda d^{(n)}}}. \quad (10)$$

Differentiating $d^{(n)}$ in order to a parameter θ of an HMM and assuming, without loss of generality, that $k = 0$ corresponds to a labelled (lab) utterance, results in

$$\frac{\partial d^{(n)}}{\partial \theta} = \sum_{k=0}^{N_{Best}} \xi_k^{(n)} \cdot \frac{\partial g(W_k^{(n)})}{\partial \theta} \quad (11)$$

where

$$\xi_k^{(n)} = \begin{cases} -1 & , \quad k = 0 \\ \zeta_k^{(n)} & , \quad k = 1 \cdots N_{Best} \end{cases} \quad (12)$$

and

$$\zeta_k^{(n)} = \frac{e^{\alpha g(W_k^{(n)})}}{\sum_{r=1}^{N_{Best}} e^{\alpha g(W_r^{(n)})}}. \quad (13)$$

These last parameters weight the importance of the competing sequence k in the solution and add up to 1. In order to simplify the analysis, the following parameters are introduced:

$$\overline{\Omega}_{htjm}^{(n)} = \psi^{(n)} \cdot \sum_{k=1}^{N_{Best}} \delta(w_{n,t}^{(k)}, h) \cdot \delta(q_t, j) \cdot \xi_k^{(n)} \cdot \beta_{jm}(\underline{x}_t^{(n)}) \quad (14)$$

$$\Omega_{htjm}^{(n)} = \psi^{(n)} \cdot \delta(w_{n,t}^{(lab)}, h) \cdot \delta(q_t, j) \cdot \beta_{jm}(\underline{x}_t^{(n)}) \quad (15)$$

$$\overline{\Theta}_{htij}^{(n)} = \psi^{(n)} \cdot \sum_{k=1}^{N_{Best}} \delta(w_{n,t}^{(k)}, h) \cdot \delta(q_{t-1}, i) \cdot \delta(q_t, j) \cdot \xi_k^{(n)} \quad (16)$$

and

$$\Theta_{htij}^{(n)} = \psi^{(n)} \cdot \delta(w_{n,t}^{(lab)}, h) \cdot \delta(q_{t-1}, i) \cdot \delta(q_t, j) \quad (17)$$

where

$$\beta_{jm}(\underline{x}) = \frac{c_{jm} \cdot b_{jm}(\underline{x})}{b_j(\underline{x})}. \quad (18)$$

$\delta(m,n)$ is the Kronecker delta function. Expression (18) can be interpreted as the weight of the mth component in the overall mixture.

Resuming the differentiation that began in (9), in order to obtain all parameters of the HMMs and make the gradient vanish, we obtain the following estimation expressions for each vector component l:

$$\mu_{jml}^{(h)} = \frac{\sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \Omega_{htjm}^{(n)} \cdot x_{t,l}^{(n)} - L_j^{(h)} \sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \overline{\Omega}_{htjm}^{(n)} \cdot x_{t,l}^{(n)}}{\sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \Omega_{htjm}^{(n)} - L_j^{(h)} \sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \overline{\Omega}_{htjm}^{(n)}}; \quad (19)$$

$$\sigma_{jml}^{(h)^2} = \frac{\sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \Omega_{htjm}^{(n)} \left(x_{t,l}^{(n)} - \mu_{jml}^{(h)}\right)^2 - L_j^{(h)} \sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \overline{\Omega}_{htjm}^{(n)} \left(x_{t,l}^{(n)} - \mu_{jml}^{(h)}\right)^2}{\sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \Omega_{htjm}^{(n)} - L_j^{(h)} \sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \overline{\Omega}_{htjm}^{(n)}} \quad (20)$$

To obtain estimation expressions of probability transitions or Gaussian mixture weights, we need to ensure that the following stochastic restrictions are verified:

$$\sum_{j=1}^{N_s^{(h)}} a_{ij}^{(h)} = 1 \; ; \; \sum_{m=1}^{M_j^{(h)}} c_{jm}^{(h)} = 1 \quad (21)$$

where $N_s^{(h)}$ is the number of states of HMM h. Using Lagrangian multipliers the solutions are

$$a_{ij}^{(h)} = \frac{p_{ij}^{(h)}}{\sum_{j=1}^{N_s^{(h)}} p_{ij}^{(h)}} \; ; \; c_{jm}^{(h)} = \frac{d_{jm}^{(h)}}{\sum_{m=1}^{M_j^{(h)}} d_{jm}^{(h)}} \quad (22)$$

where

$$p_{ij}^{(h)} = \sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \Theta_{htij}^{(n)} - L_i^{(h)} \cdot \sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \overline{\Theta}_{htij}^{(n)} \quad (23)$$

and

$$d_{jm}^{(h)} = \sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \Omega_{htjm}^{(n)} - L_j^{(h)} \cdot \sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \overline{\Omega}_{htjm}^{(n)} \; . \quad (24)$$

It should be noted that some other constraints need to be verified:

$$\sigma_{jml}^{(h)^2} > 0 \; ; \; a_{ij}^{(h)} \geq 0 \; ; \; c_{jm}^{(h)} \geq 0, \; \forall \; h, \; j, \; i, \; m, \; l \quad (25)$$

Therefore $L_j^{(h)}$ and $L_i^{(h)}$ need to fulfil the following restrictions

$$L_i^{(h)} = \eta \cdot \min_j \left(\frac{\sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \Theta_{htij}^{(n)}}{\sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \overline{\Theta}_{htij}^{(n)}} \right) \quad (26)$$

and

$$L_j^{(h)} = \eta \cdot \min_m \left\{ \min_l \left\{ \frac{\sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \Omega_{htjm}^{(n)} \left(x_{t,l}^{(n)} - \mu_{jml}^{(h)}\right)^2}{\sum_{n=1}^{N_u}\sum_{t=1}^{T^{(n)}} \overline{\Omega}_{htjm}^{(n)} \left(x_{t,l}^{(n)} - \mu_{jml}^{(h)}\right)^2} \right\}, \frac{\Omega_{hjm}}{\overline{\Omega}_{hjm}} \right\} \quad (27)$$

with the constraint that $0 \leq \eta < 1$. Note that when $\eta = 0$ FMET leads to ML training.

2.4. Training Procedure

It should be noted that this algorithm is intended for batch mode operation. The main steps to FMET implementation are the following:

1. Initialize all HMMs with ML (or take η=0).
2. Accumulate (14), (15), (16) and (17) for all training utterances.
3. Determine $L_i^{(h)}$ and $L_j^{(h)}$ for all HMMs and states, according to (26) and (27), respectively.
4. Update all HMMs parameters computing (19), (20) and (22), using (23) and (24).
5. Save the new HMMs and evaluate the performance using the updated HMMs.
6. Return to step 2) until the required number of iterations is reached.

If the performance does not improve in one iteration, it will normally improve in the subsequent ones. Also, the FMET first result, in all experiments, was better than the best GD result over several iterations.

3. Experiments and Results

The experiments were carried out using a Portuguese speech command database [8]. The training set consisted of 103001 utterances and the test set consisted of 27382 different utterances of 254 commands.

Acoustic models were built for monophones, triphones and words using HTK3.4 [9]. The input features were 12 MFCCs plus energy, and their first and second order time derivatives computed at a rate of 10ms and within a window of 25ms. Evaluation was done by means of *accuracy* rate.

To describe the Portuguese language 38 monophones were used plus a silence model. Each class was modelled by a three-state left-to-right HMM, except the silence one where transitions to previous states were allowed. Each state was modelled using a mixture of 16 Gaussians. The set of triphones is composed by 955 HMMs (found in the 254 commands) also with 16 Gaussians. The 255 whole-word models correspond to the 254 commands plus the silence. The number of emitting states ranged from 3 to 39 modeled with 10 Gaussians. The test was carried out using a task grammar with all the 254 word-commands in parallel, preceded by and ending with the silence model.

One of the initial conditions of the method is the choice of the λ in (10) and in α (13) parameters. Using typical Viterbi decoding scores we found $\lambda = 0.05$ and $\alpha = 0.001$ a good trade-off between these two parameters. In FMET a value of η between 0.6 and 0.8 was used. In MCE using Rprop the update value of each HMM parameter was set by multiplying 0.01 by each parameter value, and the increasing and decreasing factors were η^+=1.2 and η^-=0.5, respectively. For MMI and MPE the i-smoothing and learning rate factors were set as suggested in [9].

Table 1 presents the results obtained after training the HMMs with only one iteration of FMET. The results with Rprop were obtained with 10 iterations for monophones and 2 iterations for triphones and words. The results with MMI and MPE are the best obtained after 4 iterations. In fact, the results with MPE for triphones decreased at the 2nd iteration and then

increased again at further iterations. As can be seen the single iteration of FMET method outperform Rprop, MMI and MPE using triphone and whole-word models.

Table 1 – *Comparison of FMET, Rprop, MMI and MPE performances.*

Method	Monophones	Triphones	Word
ML (before)	90.87%	97.48%	96.82%
MCE/Rprop	91.53%	97.55%	96.92%
MMI	91.95%	97.53%	96.92%
MPE	91.97%	97.55%	-
FMET (1st it.)	91.86%	97.66%	97.28%

With monophone models, MMI and MPE methods outperform FMET's first iteration result but only at the 4th iteration. This is shown in Figure 1 where the evaluation performances with monofone models over the first 4 iterations are presented.

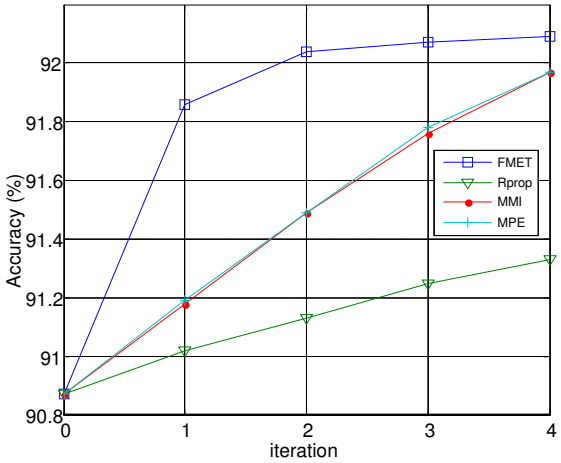

Figure 1: Performance comparison with monofone models.

4. Conclusions

In this paper a fast training algorithm based on MCE was introduced. This algorithm attempts to minimize the objective function in a single step. A new objective function was also proposed. Although the convergence of this method cannot be guaranteed, it has been shown experimentally that this method produces much better results than the Rprop, MMI or MPE approach. Moreover, it does not only achieve better results faster, but also archive results that other approach cannot achieve with several iterations. The presented results, although preliminary, allow us to extend the conclusions derived in [6] for the MCE-based HMM parameter estimation. As future work we intend also to apply the method to other well-known speech databases and larger tasks.

5. REFERENCES

[1] Y. Normandin et al., "High performance connected digit recognition using maximum mutual information estimation," IEEE Trans. Speech Audio Processing, vol. 2, pp. 229–311, April 1994.

[2] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," ICASSP-02, Orlando, FL, May 2002, pp. 105–108.

[3] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," IEEE Transactions on Speech and Audio Processing, vol. 5, pp. 257–265, May 1997.

[4] W. Chou, "Discriminant-function-based minimum recognition error rate pattern-recognition approach to speech recognition", Proc. IEEE, vol. 88, no. 8, pp. 1201–1222, Aug. 2000.

[5] H. Xiaodong, L. Deng, W. Chou, "A Novel Learning Method for Hidden Markov Models in Speech and Audio Processing", IEEE 8th Workshop on Multimedia Signal Processing, Oct. 2006.

[6] Q. Li and B-H. Juang, "Study of a Fast Discriminative Training Algorithm for Pattern Recognition", IEEE Trans. Neural Networks, Vol. 17, No. 5, pp. 1212-1221, Sep. 2006.

[7] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm", Proc. ICNN, San Francisco, CA, 1993, pp. 586–591.

[8] J. Lopes, C. Neves, A. Veiga, A. Maciel, C. Lopes, F. Perdigão, L. Sá, "Development of a Speech Recognizer with the Tecnovoz Database", Propor 2008, International Conference on Computational Processing of Portuguese, Sept. 2008.

[9] Young, S. et al, "The HTK book. Revised for HTK version 3.4", Cambridge University Engineering Department, Cambridge, December 2006.