A NEW ALGORITHM FOR FAST DISCRIMINATIVE TRAINING

Qi Li and Biing-Hwang Juang†

Multimedia Communications Research
Bell Labs, Lucent Technologies
Murray Hill, NJ 07974
Email: qli@bell-labs.com

†AVAYA Labs Research
Basking Ridge, NJ 07920
Email: bjuang@avaya.com

ABSTRACT

Currently, almost all discriminative training algorithms for nonlinear classifier design are based on gradient-descent methods, such as the backpropagation and the generalized probabilistic descent algorithm. These algorithms are easy to derive and effective in applications. However, a drawback for the gradient-descent approaches is the slow training speed, which limits their applications in large training problems, such as large vocabulary speech recognition and other applications. For hidden Markov models, some training algorithms, such as the reestimation (or expectation-maximization) algorithm for maximum likelihood estimation (MLE), are fast, but they are not readily extendible to discriminative training for recognition performance improvements. To address the problem, we proposed a fast discriminative training algorithm in this paper. It is a batch-mode algorithm derived for the objective function of minimal error rate. The significant advantage is its closed-form solution for parameter estimation during iterations, instead of incremental search in the direction of gradient, as conventionally done. We experimentally show that the algorithm requires only a few iterations to achieve the optimization objective and that the estimated results lead to better recognition performance than a traditional MLE.

1. INTRODUCTION

There are two kinds of pattern classifier: linear and nonlinear. A linear classifier, such as a single perceptron, usually constructs a hyper-plan to partition the data space. A nonlinear classifier, such as multi-layer perceptron (MLP) networks, Gaussian mixture models (GMM), hidden Markov models (HMM), etc., usually consists of nonlinear kernels to model the data distribution. There ate two kinds of training algorithms: discriminative and distribution estimation. The objective of discriminative training is to reduce the classification error as much as possible; algorithms in this class include backpropagation [1], and generalized probabilistic descent (GPD) [2]. Distribution estimation based algorithms are based on the Bayes classifier formulation with suggests estimation of the data distribution as the first and imperative step in the design of a classifier. The most commonly used criterion for distribution estimation is maximum likelihood (ML) [3]. The Expectation-maximization (EM) algorithm for ML estimation is in general very efficient because while it is a hill-climbing algorithm, it guarantees net gain in the optimization objective at every step of the iteration, leading to uniform, rather than stochastic, convergence. Discriminative training algorithm for nonlinear classifiers are generally based on gradient descent and thus converge slowly. The slow convergence makes applications of these algorithms in large vocabulary speech recognition (LVSR) unattractive.

In this paper, we propose a fast discriminative training algorithm for a nonlinear classifier, namely the Gaussian Mixture Model (GMM), which is the basic component of hidden Markov model (HMM). We use *minimum error rate* (MER) as the objective for discriminative training and thus name the algorithm *fast MER optimization*. It is a batch-mode approach with a closed-from solution for local optimization during each iteration. The model parameters are first initialized by an ML estimation, then in a few iterations, the model is trained to optimize the objective function of MER. We demonstrate through experiment that the algorithm produces optimization results as fast as the reestimation algorithm in terms of efficient and better than the MLE in terms of the classification performance.

2. DERIVATION OF THE FAST MER ESTIMATION

In an M-class classification problem, we are 1) given an observation $\mathbf{x}$, a member of class C_j, and 2) asked to make a decision, to classify $\mathbf{x}$ into, say, class C_i. We denote this as an action α_i. The decision is correct if $i = j$, otherwise incorrect. It is natural to seek a decision rule that minimize the probability of error, or error rate. A popular function of interest for this kind of problem is called zero-one loss function:

$$\mathcal{L}(\alpha_i|C_j) = \begin{cases} 0 & i = j \\ 1 & i \neq j. \end{cases} \qquad i, j = 1, ..., M \tag{1}$$

It assigns no loss to a correct decision and assign a unit loss to any error. The risk corresponding to this loss function is

$$R(\alpha_i|\mathbf{x}) = \sum_{j=1}^{M} \mathcal{L}(\alpha_i|C_j)P(C_j|\mathbf{x}) = 1 - P(C_i|\mathbf{x}) \quad (2)$$

where $P(C_i|\mathbf{x})$ is the conditional probability that the action α_i is correct. To minimize the average probability of error, we should maximize the posterior probability $P(C_i|\mathbf{x})$. This is the basis of maximum a posteriori (MAP) decision theory and is also empirically referred to as minimum error rate (MER) or minimum classification error (MCE) [2] criterion. We therefore aim at maximizing the overall posterior probability as:

$$\max J = \sum_{m=1}^{M}\sum_{n=1}^{N_m} P(\lambda_m|\mathbf{x}_{m,n}) = \sum_{m=1}^{M}\sum_{n=1}^{N_m} \frac{p(\mathbf{x}_{m,n}|\lambda_m)P_m}{p(\mathbf{x}_{m,n})} \quad (3)$$

where for class m, $\mathbf{x}_{m,n}$ is the nth training token, N_m is the total number of tokens, and P_m and λ_m are the prior probability and the mixture model, respectively.

Note that the MAP objective can be rewritten as:

$$\max J = \sum_{m=1}^{M}\sum_{n=1}^{N_m} \ell(d_{m,n}) = \sum_{m=1}^{M}\sum_{n=1}^{N_m} \ell_{m,n} \quad (4)$$

where $\ell = \frac{1}{1+e^{-d_{m,n}}}$ is the sigmoid function and

$$d_{m,n} = \log p(\mathbf{x}_{m,n}|\lambda_m)P_m - L\log\sum_{j\neq m} p(\mathbf{x}_{m,n}|\lambda_j)P_j \quad (5)$$

where $0 < L \leq 1$ is a weighting scalar. When $L = 1$, Eqn. (4) is equivalent to Eqn. (3) and to the empirical cost defined in [2] as the so-called MCE discriminative training objective. Specifically, let the model be a Gaussian mixture (GMM):

$$p(\mathbf{x}_{m,n}|\lambda_m) = \sum_{i=1}^{I} c_{m,i}p(\mathbf{x}_{m,n}|\lambda_{m,i}) \quad (6)$$

where $\sum_i^I c_{m,i} = 1$, and I is the total number of mixtures.

Let $\nabla_{\theta_{m,i}} J$ be the gradient of J with respect to $\theta_{m,i} \subset \lambda_{m,i}$. Vanishing the gradient to maximize J, we have:

$$\begin{aligned}\nabla_{\theta_{m,i}} J = &\sum_{n=1}^{N_m} \omega_m(\mathbf{x}_{m,n})\nabla_{\theta_{m,i}}\log p(\mathbf{x}_{m,n}|\lambda_{m,i}) \\ &-L\sum_{j\neq m}\sum_{\bar{n}=1}^{N_j} \bar{\omega}_j(\mathbf{x}_{j,\bar{n}})\nabla_{\theta_{m,i}}\log p(\mathbf{x}_{m,\bar{n}}|\lambda_{m,i}) \\ &= 0\end{aligned} \quad (7)$$

where,

$$\omega_m(\mathbf{x}_{m,n}) = \ell_{m,n}(1-\ell_{m,n})\frac{c_{m,i}p(\mathbf{x}_{m,n}|\lambda_{m,i})P_m}{p(\mathbf{x}_{m,n}|\lambda_m)P_m} \quad (8)$$

$$\bar{\omega}_j(\mathbf{x}_{j,\bar{n}}) = \ell_{j,\bar{n}}(1-\ell_{j,\bar{n}})\frac{c_{m,i}p(\mathbf{x}_{j,\bar{n}}|\lambda_{m,i})P_m}{\sum_{k\neq j} p(\mathbf{x}_{j,\bar{n}}|\lambda_k)P_k}. \quad (9)$$

To be able to get a closed-form solution, we assume that the model parameters in $p(\mathbf{x}_{m,n}|\lambda_{m,i})$ of Eqn. (7) is independent to the parameters in ω_m and $\bar{\omega}_j$ within each iteration.

From Eqn. (6), we have

$$\begin{aligned}\log p(\mathbf{x}_{m,n}|\lambda_{m,i}) = &-\log[(2\pi)^{d/2}|\Sigma_{m,i}|^{1/2}] \\ &-\frac{1}{2}(\mathbf{x}_{m,n}-\mu_{m,i})^T\Sigma_{m,i}^{-1}(\mathbf{x}_{m,n}-\mu_{m,i}).\end{aligned}$$

For the covariance matrix estimation, taking the derivative with respect to matrix $\Sigma_{m,i}$, we have

$$\begin{aligned}\nabla_{\Sigma_{m,i}}\log p(\mathbf{x}_{m,n}|\lambda_{m,i}) = &-\frac{1}{2}\Sigma_{m,i}^{-1} \\ &+\frac{1}{2}\Sigma_{m,i}^{-1}(\mathbf{x}_{m,n}-\mu_{m,i})(\mathbf{x}_{m,n}-\mu_{m,i})^T\Sigma_{m,i}^{-1}.\end{aligned} \quad (10)$$

Bringing Eqn. (10) into Eqn. (7) and rearranging the terms, we obtain the formula to estimate the covariance matrix:

$$\begin{aligned}\Sigma_{m,i} = &\frac{1}{D}\left[\sum_{n=1}^{N_m}\omega_m(\mathbf{x}_{m,n})(\mathbf{x}_{m,n}-\mu_{m,i})(\mathbf{x}_{m,n}-\mu_{m,i})^T\right. \\ &\left.-L\sum_{j\neq m}\sum_{\bar{n}=1}^{N_j}\bar{\omega}_j(\mathbf{x}_{j,\bar{n}})(\mathbf{x}_{j,\bar{n}}-\mu_{m,i})(\mathbf{x}_{j,\bar{n}}-\mu_{m,i})^T\right] \\ = &\frac{A-LB}{D}\end{aligned} \quad (11)$$

where

$$D = \sum_{n=1}^{N_m}\omega_m(\mathbf{x}_{m,n}) - L\sum_{j\neq m}\sum_{\bar{n}=1}^{N_j}\bar{\omega}_j(\mathbf{x}_{j,\bar{n}}). \quad (12)$$

To ensure that $\Sigma_{m,i}$ be positive definite, we need to adjust the value of L. Let $A - LB = U^T(\tilde{A} - L\tilde{B})U$, where U is the orthogonal eigenvalue matrix so both $\tilde{A}$ and $\tilde{B}$ are diagonal. Then, $A - LB$ is positive definite if

$$L < \min\left\{\frac{\tilde{a}_i}{\tilde{b}_i}\right\}_{i=1}^{d}, \quad (13)$$

where $\tilde{a}_i > 0$ and $\tilde{b}_i > 0$ are the diagonal entries of $\tilde{A}$ and $\tilde{B}$, respectively. L also needs to satisfy $D(L) > 0$ and $0 < L \leq 1$.

For the mean vector estimation, take the derivative of Eqn. (10) with respect to vector $\mu_{m,i}$. We have

$$\nabla_{\mu_{m,i}}\log p(\mathbf{x}_{m,n}|\lambda_{m,i}) = \Sigma_{m,i}^{-1}(\mathbf{x}_{m,n}-\mu_{m,i}). \quad (14)$$

Bringing Eqn. (14) into Eqn. (7) and rearranging the terms, we obtain the formula for mean vector estimation:

$$\mu_{m,i} = \frac{1}{D}\left[\sum_{n=1}^{N_m}\omega_m(\mathbf{x}_{m,n})\mathbf{x}_{m,n} - L\sum_{j\neq m}\sum_{\bar{n}=1}^{N_j}\bar{\omega}_j(\mathbf{x}_{j,\bar{n}})\mathbf{x}_{j,\bar{n}}\right] \quad (15)$$

where D is defined in Eqn. (12) and, and the scalar L has been determined when estimating $\mathbf{\Sigma}_{m,i}$.

The last step is to estimate the mixture parameters $c_{m,i}$. We now need to maximize J subject to $\sum_i^I c_{m,i} = 1$. Introducing Lagrangian multiplier γ, we have

$$\tilde{J} = J + \gamma(\sum_{i=1}^{I} c_{m,i} - 1) + \cdots. \tag{16}$$

Taking the first derivate and vanish it for maximum, we have

$$\frac{\partial \tilde{J}}{\partial c_{m,i}} = \frac{1}{c_{m,i}} D + \gamma = 0. \tag{17}$$

Rearranging the terms, we have

$$c_{m,i} = -\frac{1}{\gamma} D. \tag{18}$$

Summer over $c_{m,i}$, for $i = 1 \ldots I$, we can solve γ as

$$\gamma = -\left[\sum_{n=1}^{N_m} \sum_{i=1}^{I} \omega_m(c_i, \mathbf{x}_{m,n}) - L \sum_{j \neq m} \sum_{\bar{n}=1}^{N_j} \sum_{i=1}^{I} \bar{\omega}_j(c_i, \mathbf{x}_{j,\bar{n}}) \right]. \tag{19}$$

In summary, the procedure of the proposed fast MER estimation is:

1. Initialize all models parameters by MLE;
2. For every mixture component i in model m, compute $\omega_{m,i}$ and $\bar{\omega}_{m,i}$ using Eqn. (8) and Eqn. (9);
3. Determine the weighting parameter L by Eqn. (13) and other constrains;
4. For every mixture component i, estimate $\mathbf{\Sigma}_{m,i}$ $\mu_{m,i}$, and $c_{m,i}$ using Eqns. (11), (15), and (18);
5. Evaluate the performance using Eqn (3) for model m;
6. If the performance is improved, keep the new model and goto Step 2, otherwise break;
7. Repeat the above procedure for all models.

3. EXPERIMENTAL RESULTS

To evaluate the proposed algorithm, we conduct two experiments.

3.1. Three-Class Classification

In this example, we artificially generated three classes of 2-dimensional data. The distributions were of Gaussian mixture types with three components in each class. Each token is of two dimensions. For each class, 1,500 tokens were drawn from each of the three components; therefore, there are 4,500 tokens in total. The contours of the training data for class 1, 2 and 3 are shown in Fig. 1 where the means are represented as $+$, $*$, and $\square$, respectively.

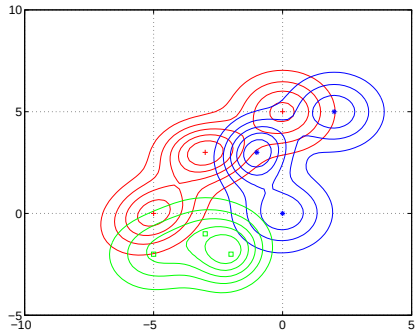

Figure 1: Contours of 3 classes of training data

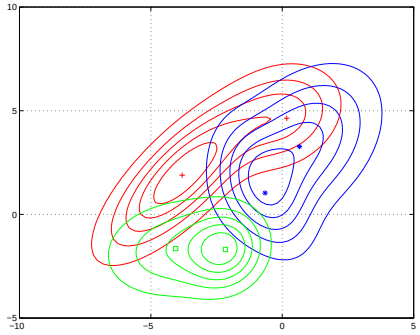

Figure 2: Contours of 3 models after MLE.

To simulate real applications, we assume that the number of mixture components are unknown. Therefore, we assume the GMMs, which need to be trained, have two mixture components for each class. In the first step, MLE was applied to initialize the GMMs. The contours that represent the pdf's of each of the GMMs after MLE are plotted in Fig. 2. We then use the proposed fast MER estimation to train new GMMs. The contours to represent the pdf's of each of the GMMs after MER estimation are plotted in Fig. 3. All above contours are plotted on the same scale.

By comparing Fig. 2 and Fig. 3, we can observe that MER training reduces the overlaps amount three classes significantly. The MLE took 3 or 4 iterations for each of the models while the MER estimation conducted two iterations for the first class and 0 iterations for others since MER training found further iterations on them unnecessary. The testing data with 4,500 tokens for each class were drawn in the same methods as the training data. The MLE provides an accuracy of 76.46% and 76.47% for training and testing datasets respectively, while the proposed MER estimation improved the accuracy to 76.96% and 76.71%. If we use the same model of generating the training data, the ideal

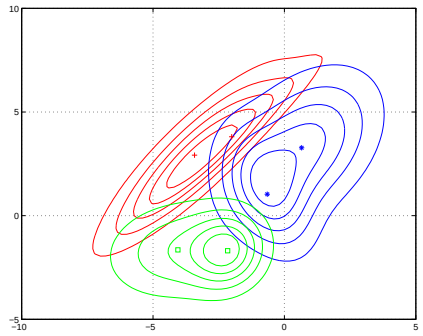

Figure 3: Contours of 3 models after the proposed MER training with two iterations on the classes with "+" means.

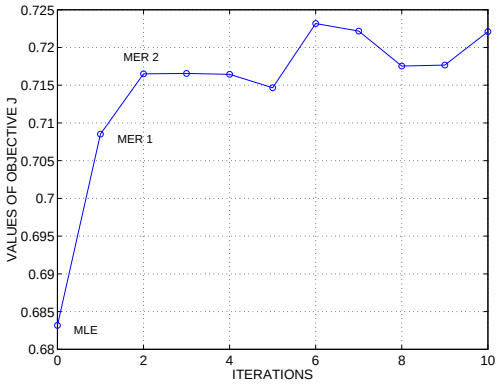

Figure 4: MER iterations on the first class.

performances are 77.39% and 77.11% for training and testing, which are the ceilings of this example. To evaluate the behaviors of the MER training, we plot the values of the objective as a function of 10 iterations in Fig. 4. Although there is no guarantee on convergence on each iteration, the proposed algorithm seems effective. Actually, training was stooped at the second iteration for the above performances.

3.2. Vowel Classification

The second experiment used the Peterson-Barney vowel formant data [4]. It consisted of 10 vowels from 76 speakers (33 men, 28 women and 15 children). Each vowel was represented in terms of 4 features: the fundamental frequency and the first three formant frequencies. Fundamental and the formant frequencies were measured by Peterson and Barney from the central steady-state portions of the utterances. There were two tokens for each vowel of each speaker. We used the first token as training dataset and the second token as testing dataset. There were 10 classes in total and each vowel represents one class. The classifiers were first trained by MLE in two iterations, then MER estimation. The performance was saturated after the first MER iteration. The

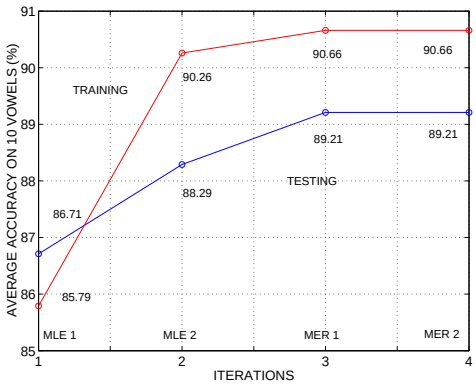

Figure 5: MLE and MER training and testing performances.

experimental results on both training and testing datasets are plotted in Fig. 5. The proposed MER algorithm improved the classification performance in one iteration.

4. CONCLUSIONS

We proposed a fast discriminative training algorithm for MER estimation. Based on an assumption, the algorithm provides a closed-form solution for parameter re-estimation in each iteration. Therefore, like MLE, the algorithm can train classifiers in a few iterations which is faster than gradient-descent-based methods. We note that this paper only reports some preliminary results. Further research is necessary to investigate the properties of the proposed MER algorithm. It will be straight forward to extend the algorithm to train hidden Markov models for automatic speech recognition if we can further prove that this algorithm is effective and efficient in more classification applications.

5. ACKNOWLEDGMENT

The authors wish to thank Jingdong Chen for providing the second experimental datasets.

6. REFERENCES

[1] P. J. Werbos, *The roots of backpropagation : from ordered derivatives to neural networks and political forecasting*. New York: J. Wiley & Sons, 1994.

[2] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Transactions on Signal Processing*, vol. 40, pp. 3043–3054, December 1992.

[3] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, Second Edition*. New York: John & Wiley, 2001.

[4] R. Watrous, "Current status of Paterson-Barney vowel formant data," *J. Accoust. Soc. Am.*, vol. 89, pp. 2459–2460, May 1991.