RC22616 (W0210-208) October 31, 2002
Computer Science

IBM Research Report

Discriminative Training of Tied Mixture Density HMMs for Online Handwritten Digit Recognition

Roongroj Nopsuwanchai
Computer Laboratory,
University of Cambridge,
Cambridge, CB3 0FD, UK

Alain Biem
IBM T. J. Watson Research Center,
P.O. Box 218, Yorktown Heights,
New York, 10598, USA

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

DISCRIMINATIVE TRAINING OF TIED MIXTURE DENSITY HMMS FOR ONLINE HANDWRITTEN DIGIT RECOGNITION

Roongroj Nopsuwanchai

Computer Laboratory,
University of Cambridge,
Cambridge, CB3 0FD, UK
Email: *rn225@cam.ac.uk*

Alain Biem

IBM T. J. Watson Research Center,
P.O. Box 218, Yorktown Heights,
New York, 10598, USA
Email: *biem@us.ibm.com*

ABSTRACT

This paper describes and evaluates the Maximum Mutual Information criterion (MMI) for online unconstrained-style handwritten digit recognition based on Hidden Markov Models (HMMs). The study focuses on determining the best MMI optimization scheme and the HMM parameters that exhibit the most discriminative capabilities in the context of Tied Mixture Density Hidden Markov Models (TDHMMS), where all HMM states share a pool of Gaussians. The experimental results show that the second-order optimization scheme is the most efficient and that although means and covariance matrix are shared by all models, they contribute the most to discrimination.

1. INTRODUCTION

Online handwriting recognition has recently received a renewed interest as exemplified by its widespread use in handheld computers and Personal Digital Assistants (PDAs). Although tremendous progress has been made in handwriting recognition, the recognition of unconstrained-style handwriting where the writer freely writes with his/her own style remains a challenging problem, requiring complex models and powerful hardware. The problem is compounded when the recognizer is deployed on a platform with limited power and memory: the recognizer should be light and at the same time should exhibit high performance on freely-written handwriting to be useful. In a HMM-based framework, a lightweight recognizer can be obtained by parameters tying, where parameters are fully or partially shared by all models, and, as demonstrated in various studies, discriminative training can significantly improve a system with a smaller number of parameters [1].

Tied Mixture Density HMMs, also known as Semicontinuous HMMs, use a single set of Gaussian kernels that is shared by all states [2]. Given a feature-vector $\mathbf{x}$, the probability density function (pdf) $b_{ij}(\mathbf{x})$ of state j and character C_i is given by

$$b_{ij}(\mathbf{x}) = \sum_{k=1}^{K} w_{ijk} \aleph(\mathbf{x}, \mu_k, \mathbf{\Sigma}_k), \tag{1}$$

where $\aleph(\mathbf{x}, \boldsymbol{\mu}_k, \mathbf{\Sigma}_k)$ is k-th Normal density with mean μ_k and covariance matrix $\mathbf{\Sigma}_k$ taken from a codebook of K Normal densities. The means and the covariance matrices are shared by all states, only the mixing weights $w_{ijk} > 0$ are state-dependent with $\sum_{k=1}^{K} w_{ijk} = 1$. It has been demonstrated that tying pdfs as described in Eq. (1) significantly reduces the number of parameters in the system while achieving similar modeling capabilities as Continuous Densities HMMs (CDHMMS) where pdfs are untied [3].

HMM-based discriminative training has been successfully applied to speech and handwriting recognition, showing improvement from the Maximum Likelihood Estimation (MLE)-based framework on a variety of tasks ranging from small vocabulary to large-vocabulary tasks [4]. Most success of discriminative training, however, has been realized in the context of CDHMMs, where the output densities attached to each state cover a localized region of the feature space. Such an architecture, where models are well-separated from each other is quite appropriate for a discriminative training scheme. It remains unclear whether discriminative training is effective in a highly-tied system such as a system based on TDHMMs.

This paper thus concentrates on evaluating MMI training of a TDHMM-based system targeting unconstrained-style handwriting recognition digits. This multi-writers digit recognition task provides the ideal framework to study the effectiveness of the MMI training in this context. The goal of this exhaustive study is twofold: finding the best optimization algorithm to be used for MMI training in the context of TDHMMs and determining the best set of optimized parameters that provide the maximum performance.

2. MAXIMUM MUTUAL INFORMATION TRAINING OF TDHMMS

MMI training consists of maximizing the mutual information between a training token X and its label/category C. We are given a body of training data $D = \{X_1, ..., X_n, ..., X_N\}$ of N training tokens, with a sequence of labels $\{\mathcal{L}_1, ..., \mathcal{L}_n, ..., \mathcal{L}_N\}$, where $\mathcal{L}_n$ is label of X_n and $\mathcal{L}_n$ belongs to the set of categories $\{C_1, C_2, ..., C_M\}$. The MMI objective function $\mathcal{M}(\lambda)$ to be optimized is

$$\mathcal{M}(\lambda) = \sum_{n=1}^{N} \left(\log P(X_n|\mathcal{L}_n; \lambda) - \log \sum_{i=1}^{M} P(X_n|C_i; \lambda) P(C_i) \right) \tag{2}$$

where λ represents the set of all parameters in the system. In contrast to MLE which is aimed at estimating probabilities, the optimization of the mutual information objective function in Eq. (2) realizes a discrimination between the likelihood of the correct model and the probability of the data. In the context of this paper, the prior probability of class $P(C_i)$ is assumed to be uniform.

2.1. MMI Optimization algorithms

One reason behind the popularity of MLE training in HMM is the powerful Baum-Welch optimization algorithm which has guaranteed convergence to a local minimum. Empirical evidence shows that good parameter estimates are usually obtained after a small number of iterations. A similar algorithm, called the extended Baum-Welch algorithm (EBW), adapted to the MMI situation has been proposed [5] and extensively used.

We implemented a first- and second-order optimization gradient descent technique, in the context of a TDHMM-based system, and compared it to the EBW technique. The details of the optimization algorithms are described below. In this paper, all optimizations are done within the Viterbi training framework.

2.1.1. Extended Baum-Welch Algorithm

Although the EBW version described in this section uses the Viterbi path to gather parameter sufficient statistics, it is still referred as EBW. We let Φ_i^n represents the sequence of states within the model of character C_i as found by the Viterbi algorithm, given the n-th training token $X_n = \{\mathbf{x}_{n1}, ..., \mathbf{x}_{nt}, ...\mathbf{x}_{nT_n}\}$ of length T_n. Given a parameter λ_{ijk} belonging to the HMM of character C_i, state j, and mixture k, the EBW re-estimates the parameters as

$$\hat{\lambda}_{ijk} = \frac{\lambda_{ijk}\left(\frac{\partial \mathcal{M}(\lambda)}{\partial \lambda_{ijk}} + D\right)}{\sum_{l=1} \lambda_{ijl}\left(\frac{\partial \mathcal{M}(\lambda)}{\partial \lambda_{ijl}} + D\right)} \tag{3}$$

with a sufficiently large constant D. The algorithm can be applied straightforwardly for all variables with property that $\sum_k \lambda_{ijk} = 1$. This includes the mixing weight w_{ijk}. However, there is no closed-form solution for this algorithm when applied to means and covariance matrix.

Fortunately, it has been shown that by considering a continuous Gaussian distribution as a limit of a discrete probability function, a re-estimation formula, can be computed [6]. Let m_k, Σ_k and x_{nt} be a component of the means μ_k, covariance matrix $\mathbf{\Sigma}_k$, and feature-vector $\mathbf{x}_{nt}$, respectively. The re-estimation formulae, adapted to the TDHMM context and using the Viterbi path statistics are as follows:

$$\hat{m}_k = \frac{(\Theta_k - \overline{\Theta}_k) + Dm_k}{(\gamma_k - \overline{\gamma}_k) + D} \tag{4}$$

$$\hat{\Sigma}_k = \frac{(\Psi_k - \overline{\Psi}_k) + D(\Sigma_k + m_k^2)}{(\gamma_k - \overline{\gamma}_k) + D} - m_k^2 \tag{5}$$

where

$$\Theta_k = \sum_{n=1}^{N}\sum_{i=1}^{M}\sum_{j\in\Phi_i^n, t=1}^{T_n} x_{nt}\gamma_{ijk}^n(t)\delta(i,n) \tag{6}$$

$$\overline{\Theta}_k = \sum_{n=1}^{N}\sum_{i=1}^{M}\sum_{j\in\Phi_i^n, t=1}^{T_n} x_{nt}\overline{\gamma}_{ijk}^n(t) \tag{7}$$

and

$$\Psi_{ijk} = \sum_{n=1}^{N}\sum_{i=1}^{M}\sum_{j\in\Phi_i^n, t=1}^{T_n} x_{nt}^2\gamma_{ijk}^n(t)\delta(i,n) \tag{8}$$

$$\overline{\Psi}_{ijk} = \sum_{n=1}^{N}\sum_{i=1}^{M}\sum_{t=1}^{T_n} x_{nt}^2\overline{\gamma}_{ijk}^n(t) \tag{9}$$

with $\delta(i,n) = 1$ if $\mathcal{L}_n = C_i$ and zero otherwise. The occupancy γ_k and the anti-occupancy $\overline{\gamma}_k$ of the Gaussian k are defined respectively as:

$$\gamma_k = \sum_{n=1}^{N}\sum_{i=1}^{M}\sum_{j\in\Phi_i^n, t=1}^{T_n} \gamma_{ijk}^n(t)\delta(i,n) \tag{10}$$

$$\overline{\gamma}_k = \sum_{n=1}^{N}\sum_{i=1}^{M}\sum_{j\in\Phi_i^n, t=1}^{T_n} \overline{\gamma}_{ijk}^n(t) \tag{11}$$

where

$$\gamma_{ijk}^n(t) = \frac{w_{ijk}\aleph(\mathbf{x}_{nt}, \mu_k, \mathbf{\Sigma}_k)}{\sum_{l=1}^{K} w_{ijl}\aleph(\mathbf{x}_{nt}, \mu_l, \mathbf{\Sigma}_l)} \tag{12}$$

and

$$\overline{\gamma}_{ijk}^n(t) = \frac{P(X_n|C_i)}{\sum_{l=1}^{M} P(X_n|C_l)}\gamma_{ijk}^n(t). \tag{13}$$

One of the critical issues in EBW is the choice of the parameter D. In this paper, we followed the approach in [6], where the choice is made so that the weights estimation remains positive.

2.1.2. Gradient Descent

The gradient descent approach is an iterative process that updates the parameter λ_{ijk} according to:

$$\hat{\lambda}_{ijk} = \lambda_{ijk} + \eta\frac{\partial \mathcal{M}(\lambda)}{\partial \lambda_{ijk}} \tag{14}$$

where a small $\eta > 0$ is the learning rate. When applied to mixing weights, a softmax transformation is used, that is, $w_{ijk} = \frac{\exp(\alpha_{ijk})}{\sum_{l=1}^{K}\exp(\alpha_{ijl})}$, ensuring the mixing weight remains positive during optimization and its summation equals one.

2.1.3. Second-Order Gradient Descent

The second order gradient descent refers to the quickprop algorithm [7] that has received widespread use in Neural-Network-based systems. The quickprop algorithm combines a gradient descent technique and the Newton algorithm, and uses an approximation of the Hessian matrix. The parameter set is updated as follows:

$$\hat{\lambda_{ijk}} = \lambda_{ijk} - [\frac{\partial^2 \mathcal{M}(\lambda)}{\partial^2 \lambda_{ijk}} + \mu I]^{-1}\frac{\partial \mathcal{M}(\lambda)}{\partial \lambda_{ijk}} \tag{15}$$

where μ is a learning rate and I is the Identity matrix. The Hessian matrix is assumed to be diagonal and is approximated by

$$\frac{\partial^2 \mathcal{M}(\lambda)}{\partial^2 \lambda_{ijk}} \approx \frac{\frac{\partial \mathcal{M}(\lambda)}{\partial \lambda_{ijk}}(\tau) - \frac{\partial \mathcal{M}(\lambda)}{\partial \lambda_{ijk}}(\tau-1)}{\lambda_{ijk}(\tau) - \lambda_{ijk}(\tau-1)}. \tag{16}$$

The quickprop algorithm replaces the approximated Hessian by zero whenever its sign does not change at learning time τ and $\tau - 1$ which indicates that the Hessian is not sufficiently positive definite. Compared to the first-order gradient descent and the EBW, the quickprop algorithm appears to be less sensitive to the choice of learning parameters issue as it makes use of the Hessian.

3. EXPERIMENTAL RESULTS

3.1. Comparison of Optimization Algorithms

We performed a number of experiments to evaluate the performance of the three algorithms. The task chosen for the evaluation was the classification of online unconstrained-style handwritten digit recognition using TDHMMs. The comparison focuses on optimizing the mixing weights parameters since they are not shared across all models, and thus are conjectured to provide the most discriminative capability in TDHMM-based systems.

The training set for this experiment is composed of 1000 isolated digits written by various writers. We use a codebook size of 100 with one state per HMM. The codebook was initially generated by a K-means clustering procedure and re-estimated when training all models by MLE. MMI training was carried out from the MLE-trained HMM models for about 40 iterations (an iteration is one pass over the whole training data).

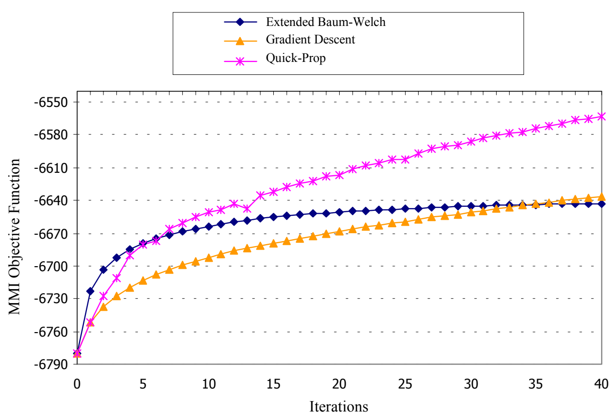

Fig. 1. Comparison of the EBW, gradient descent and quickprop for mixing weight optimization. The curve shows the MMI objective function versus the number of iterations.

Fig. 1 shows the comparison of the three optimization algorithms in terms of the objective function versus the number of iterations. The three optimization methods are all clearly efficient. The EBW algorithm exhibits the highest growth rate during the first iterations but then it tails off quickly. However, the quickprop algorithm displays the highest growth as the number of iteration increases, leading us to conclude that the quickprop optimization is the most efficient in this context.

3.2. Parameter Discriminability

Having concluded that quickprop is the most efficient algorithm, a series of experiments was carried out to determine which parameters contribute the most to performance. The parameters of the TDHMMs considered were the mixing weights (MIX), the means of Gaussians (MNS) and covariance matrices of Gaussian distributions (COV), assumed to be diagonal.

We used 5000 tokens for training and 3500 for testing. The TDHMMs share a codebook of 160 Gaussians with one state per HMM, generated as described in the section (3.1). Parameter updates in MMI training was carried out for 100 iterations using quickprop.

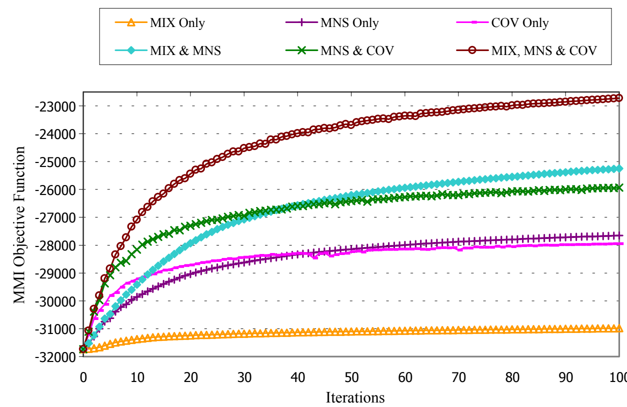

Fig. 2. Evolution of the MMI objective function versus the number iterations for various parameters.

Fig. 2 shows the evolution of the MMI objective function (on the training set) versus the number of MMI iterations for various parameters and Fig. 3 shows the performance of the training set. As seen in these figures, means and covariance optimization is the most effective in the growth of the MMI objective function and exhibits the highest increase in recognition rate on the training set. The mixing weight optimization shows little improvement to the objective function and actually results in a small decrease in recognition rate. This is rather counter-intuitive in the context of TDHMMs as mixing weights are the only parameters which are not shared across models and are therefore expected to carry more

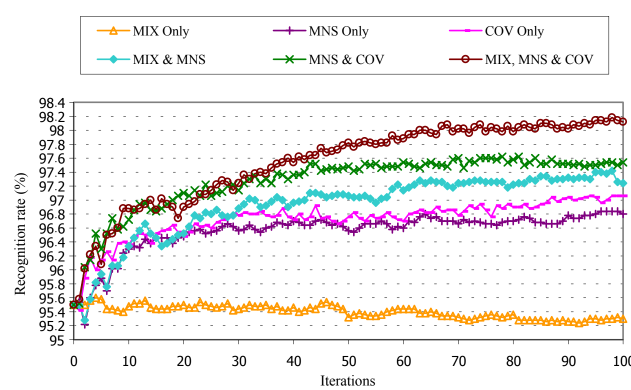

Fig. 3. Recognition rate on training set versus the number of iterations for various parameters.

discriminative capability than the means and covariance matrix. The best result is achieved, as expected, when mixing weights, means, and the covariance matrices are updated at the same time.

Table 1. Recognition results of MMI-trained THMMs on the testing set.

ML result	95.40 %
Parameters updated in MMI training	Recognition rate
MIX only	95.49 %
MNS only	95.86 %
COV only	96.14 %
MIX & MNS	95.60 %
MNS & COV	96.20 %
MIX, MNS & COV	96.29 %

Performance on the testing set is shown in Table 1. From the baseline MLE-based system, MMI training improves the recognition rate on the testing set from 95.40% to 96.29% when all HMM parameters are updated. This is equivalent to 19% relative error rate reduction, realized from an already high performing baseline. The parameter that contributes the most is the covariance matrix; the means and the mixing weights have slightly lesser contribution to discrimination.

3.3. The Effect of Parameter Tying in MMI Training

As previously argued, tied systems may inhibit the discriminative power of the MMI algorithm. In order to test this hypothesis, the same set of experiments were carried on the CDHMM-based system, using a similar number of Gaussians as the TDHMM-based system; each state pdf has its own set of 16 Gaussian distributions.

Table 2. Recognition results of MMI-trained CDHMMs on testing set.

ML result	93.71 %
Parameters updated in MMI training	Recognition rate
MIX only	94.17 %
MNS only	94.66 %
COV only	96.00 %
MIX & MNS	94.83 %
MNS & COV	96.20 %
MIX, MNS & COV	95.94 %

The results in Table 2 show the best performance on the testing set. It is clear that the mixing weights have less discriminating power than the means and covariances. The most discriminative parameters in this case are the covariance matrix of Gaussian distributions: the covariance matrix optimization improves the result of the baseline system from 93.71% to 96.00% and is far better than means and mixture weights optimization. This result is similar to the TDHMMs case but the relative improvement from the MLE-based system is much larger: The best result shows an improvement on recognition rate from 93.71% on MLE to 96.20% on MMI, which is equivalent to 40% error rate reduction. Clearly, MMI training is more effective when applied to an untied system.

4. CONCLUSIONS

We have described a study of the Maximum Mutual Information approach aimed at online unconstrained handwritten digit recognition using Tied Mixture Density HMMs. The study evaluated three optimization algorithms, namely Extended Baum-Welch, gradient descent and the quickprop algorithm. It was shown that although Extended Baum-Welch has the highest initial growth rate, the quickprop algorithm is more efficient in the long run. Optimization of various HMM parameters shows that mixing weight optimization has less effect in increasing the performance and covariance matrix optimization contributes the most. The same experiment, carried out on a Continuous Density HMMs system, confirms that covariance matrix optimization is also the most effective in an untied system.

5. ACKNOWLEDGEMENTS

Mr. Nopsuwanchai is partially funded for his PhD study by Cambridge Thai Foundation and Cambridge Overseas Trust. The authors would like to thank their colleagues Jane Snowdon for her critical reviewing of the manuscript and Michael Perrone for his interest and valuable comments on this research.

6. REFERENCES

[1] A. Biem, "Minimum Classification Error Training of Hidden Markov Model for Handwriting Recognition," *Proc. ICASSP'01*, vol. 3, pp. 1529–1532, 2001.

[2] J. R. Bellagarda and D. Nahamoo, "Parameter Modeling for Speech Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 38, no. 12, pp. 2033–2045, Dec. 1990.

[3] X. D. Huang, "Phoneme Classification Using Semicontinuous Hidden Markov Models," *IEEE Transactions on Signal Processing*, vol. 40, no. 5, 1992.

[4] V. Valtchev, J.J. Odell, P.C. Woodland, and S.J. Young, "Lattice-Based Discriminative Training for Large Vocabulary Speech Recognition," in *Proc. ICASSP'96*, Atlanta, USA, 1996, vol. 2, pp. 605–608.

[5] P. S. Gopalakrishnan, S. P. Kanevsky, D. Nadas, and D. Nahamoo, "A Generalisation of the Baum algorithm to rational objective function," in *Proc. ICASSP'89*, 1988, vol. I, pp. 20–23.

[6] Y. Normandin, R. Cardin, and R.D. Mori, "High-performance connected digit recognition using maximum mutual information estimation," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 2, no. 2, pp. 299–311, 1994.

[7] S. E. Fahlman, "An Empirical Study of Learning Speed in Back-Propagation Networks," Technical Report CMU-CS-88-162, Carnergie Mellon University, 1988.