MMI TRAINING FOR CONTINUOUS PHONEME RECOGNITION ON THE TIMIT DATABASE

S. Kapadia, V. Valtchev and S.J. Young

Cambridge University Engineering Department, England

ABSTRACT

This paper reports our experiences with a phoneme recognition system for the TIMIT database which uses multiple mixture continuous density monophone HMMs trained using MMI. A comprehensive set of results are presented comparing the ML and MMI training criteria for both diagonal and full covariance models. These results using simple monophone HMMs show clear performance gains achieved by MMI training, and are comparable to the best reported by others including those which use context-dependent models. In addition, the paper discusses a number of performance and implementation issues which are crucial to successful MMI training.

1. INTRODUCTION

Previous work[4] has shown that with infinite training data and a model space which includes the true source, the global Maximum Likelihood (ML) estimate is optimal in the sense that it yields an unbiased estimate with minimum variance. However, when constructing HMM-based speech recognisers, training data is not unlimited and the model space does not include the source. In this case, examples can be constructed where the Maximum Mutual Information (MMI) estimator can provide better discrimination than the corresponding ML estimator [5].

All this, of course, is well-known. However, clear demonstrations of the practical utility of MMI training for continuous speech recognition remain elusive. This is principally because MMI training involves a number of practical difficulties. The Baum-Welch (BW) algorithm is a robust and efficient algorithm for ML estimation, however, it cannot be applied directly to MMI. As a result, early work on MMI training was forced to use slow and somewhat unreliable gradient descent methods. Recent work has shown that the BW algorithm can be extended to the MMI case [1, 6]. However, this extended version is not straightforward to apply since there are parameters to adjust which have to strike a compromise between stability and the rate of convergence.

Thus, there are still few conclusive experimental results in support of MMI for general continuous speech and the properties of the extended BW algorithm needed for MMI training have not been reported in any depth.

2. ML ESTIMATION FOR CONTINUOUS SPEECH

In the ML estimation approach, given an acoustic observation with associated transcription pairs $y(n), t(n)$ $n = 1 \dots N$,the parameter set λ is estimated by maximising

$$L_\lambda = \sum_n \ln p_\lambda(y(n)|t(n)) \tag{1}$$

where $p_\lambda(y(n)|t(n))$ is the probability of the acoustic observations estimated from an HMM with parameters λ built to the transcription $t(n)$. The BW algorithm which is most commonly used for this task applies a transformation on the parameter set λ which is guaranteed to converge on a local maximum of L_λ. For example, the transition probabilities $a_{i,j}$ are re-estimated using BW by

$$a_{i,j}^{t+1} = \frac{a_{i,j}^t \frac{\partial L_\lambda}{\partial a_{i,j}}^t}{\sum_{k=1}^N a_{i,k}^t \frac{\partial L_\lambda}{\partial a_{i,k}}^t} \tag{2}$$

where t is the iteration index.

3. MMI TRAINING FOR CONTINUOUS SPEECH

To apply the MMI training approach to continuous speech, the parameter set is estimated by maximising

$$I_\lambda = \sum_n \ln p_\lambda(y(n)|t(n)) - \ln p_\lambda(y(n)|r) \tag{3}$$

where r represents the recognition-time HMM, that is, the composite system of sub-word models to be used at runtime including any language model.

Since this second term includes all models, it is this term which gives I_λ it's discriminative nature. At the same time however, it implies a significant increase in computational complexity.

4. MMI OPTIMISATION ALGORITHMS

Since equation 3 is a well-defined function, standard optimisation techniques can be used. We examined three such methods steepest descent, conjugate gradients and a crude second order method. We also compared these with the extended BW algorithm.

The second order method uses the following parameter update rule,

$$\lambda^{t+1} = \lambda^{t} - \eta \mathbf{H}^{t} \mathbf{g}^{t} \tag{4}$$

where $\mathbf{g}^t$ is the gradient of the objective function with respect to the parameter vector λ and $\mathbf{H}^t$ is the hessian at iteration t. The full hessian of a system with n parameters will have n^2 elements. In order to reduce the computational load due to the calculation, inversion and storage, most implementations of this method use some approximation to the hessian matrix.

In the work presented here, we use a difference approximation to the diagonal elements of the hessian.

$$\mathbf{H}^{t} = \left[h^{t}_{ii}\right] \tag{5}$$

$$h^{t}_{ii} = \frac{\partial^2 I_\lambda}{\partial \lambda_i^2} \approx \frac{\left(\frac{\partial I_\lambda}{\partial \lambda_i}^{t} - \frac{\partial I_\lambda}{\partial \lambda_i}^{t-1}\right)}{\left(\lambda_i^t - \lambda_i^{t-1}\right)} \tag{6}$$

Using equations 6 and 4 gives

$$\lambda_i^{t+1} = \lambda_i^t - \eta \frac{1}{h_{ii}} \frac{\partial I_\lambda}{\partial \lambda_i}^{t} \tag{7}$$

$$\lambda_i^{t+1} - \lambda_i^t = \eta \frac{\frac{\partial I_\lambda}{\partial \lambda_i}^{t}}{\left(\frac{\partial I_\lambda}{\partial \lambda_i}^{t-1} - \frac{\partial I_\lambda}{\partial \lambda_i}^{t}\right)} \left(\lambda_i^t - \lambda_i^{t-1}\right) \tag{8}$$

If η in the above equation is chosen to be 1.0, the equation becomes identical to the update strategy of QuickProp proposed by Fahlman [11].

The update rule given by equation 4 will converge to the nearest turning point. In order to handle this special case, we adopt the method used by Fahlman in QuickProp. No parameter change is allowed to be greater in magnitude than μ times the previous update for that parameter. If the change computed by the update formula is too large, or in the opposite direction to the current gradient, we instead use μ times the previous change as the current change. After some brief experimentation the value of 1.75 was chosen for μ in our experiments. Too large a value leads to unstable behaviour, too small leads to slow learning. One steepest descent iteration was used to bootstrap the process.

Our implementation of line search had three phases. First a step was taken out onto the line and the function and its directional derivative evaluated. This step was the previous iteration's final step size. Then the maximum was bracketed using a cubic extrapolation scheme. Bounds were placed on the candidate step size multipliers to ensure the step sizes grew at a satisfactory exponential rate. Lastly four iterations of cubic interpolation were carried out to refine the bracketing interval. Again bounds were placed on the step size candidates for all iterations except the last to ensure the bracketing interval shrunk at a satisfactory rate.

This line search was carried out on a subset of the training set chosen to contain 3 examples of each phoneme. This subset was changed for each iteration. The line search always converged with 3-5 function evaluations each requiring approximately 7% of a full function evaluation. It was noted that a good step size was mostly found by the second repetition of cubic interpolation.

This task is a constrained optimisation one. These constraints were eliminated with an x^2 substitution which ensures that all parameters (except means) are positive (positive definite for the case of full covariance matrices). The sum to one constraint on transitions and weights were not enforced. Viterbi decoding remains well defined without them.

As noted above, the BW algorithm has recently been extended to yield a new set of re-estimation equations, whereby convergence is re-established for the MMI case by adding a constant to the numerator and denominator terms.

For example, the new equation for $a_{i,j}$ is

$$a^{t+1}_{i,j} = \frac{a^{t}_{i,j}\left(\frac{\partial I_\lambda}{\partial a_{i,j}}^{t} + C\right)}{\sum_{j=1}^{N} a^{t}_{i,j}\left(\frac{\partial I_\lambda}{\partial a_{i,j}}^{t} + C\right)} \tag{9}$$

where C is a constant.

There are two drawbacks to this however,

1. equation 9, has been proved to increase I_λ only when C is large, but large C results in slow optimisation. In practice C is chosen so that the re-estimated parameters are admissible (i.e. all transition probabilities and Gaussian weights positive, and covariance matrices positive definite).
2. The extension is so far only applicable to means, variances, transitions and Gaussian weights. Hybrid schemes, where these parameters are re-estimated with BW, and the remaining with standard optimisation schemes have been suggested, and in this work, we re-estimated the stream exponents (in our models we have only 1 stream, hence the stream exponents effectively become state exponents), with gradient descent as in [6].

5. COMPARISON OF MMI TRAINING ALGORITHMS

These experiments were carried out on the 304 sentences of the dialect region 1 subset of the TIMIT training data.

Thirty-nine three-state, left to right HMMs were built. Each state had an associated mean, diagonal variance, weight, covariance multiplier and state exponent. The initial models were trained using the ML objective function.

The preprocessor used produced an observation vector every 10ms (from a 16ms window of speech), consisting of 12 mfcc coefficients, 1 log energy coefficient, and the corresponding delta coefficients.

A null-gram grammar was imposed during training.

The results are plotted in figure 1. The 'standard' methods of steepest descent and conjugate-gradients can be seen to be not very effective. Various combinations of using the full training set during line search, and alternative conjugate direction update equations (i.e. Beale-Sorenson, Fletcher-Reeves and Polak-Ribere) were tried without success.

The second order method has intermediate efficiency, but the best algorithm is the extended BW, which provides fast optimisation during the early iterations, even though

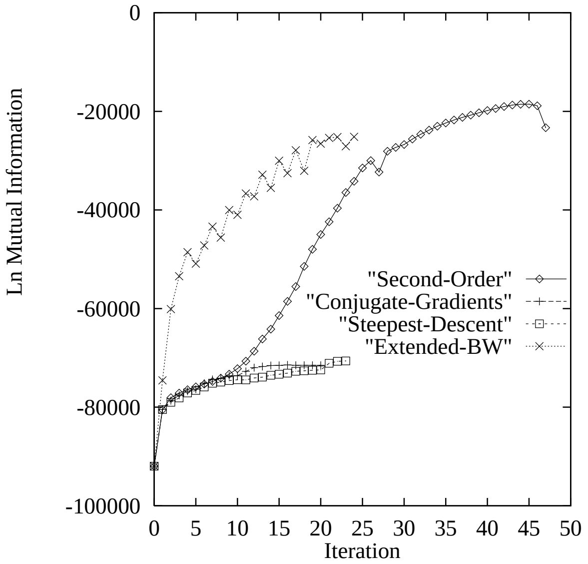

Figure 1: Comparison of MMI training algorithms

the re-estimation mapping is no-longer guaranteed to increase the objective function.

6. COMPUTATIONAL REQUIREMENTS

The quantities needed in equation 8 and equation 9 are calculated using the standard forward-backward (FB) algorithm [7, 3]. The FB algorithm has to be applied just once in the case of ML using the transcription HMM. However, it has to be applied twice for MMI, firstly using the transcription HMM and secondly using the recognition HMM.

After the forward pass the function value is available, and after the backward pass the derivatives can be calculated.

By using beam search pruning the computational complexity for the FB method can be reduced. For the left to right transcription HMM, this reduction can be significant since it is rarely necessary to have more than four models active on the forward pass and two models active in the backward pass. In our experiments, we typically achieve a 10-fold reduction in this case. Applying pruning to the FB method using the recognition model is, however, rather less effective. In our experiments, the recognition HMM is a looped phonetic model[3] where any state can be reached from any other state in a maximum of 3 time units. We found that all of the sub-word models remain active on the forward pass, on average 6 models remain active on the backward pass. In addition there is a large overhead caused by the higher connectivity of the recognition model. Hence, in practice, using the phoneme loop recognition model, the computational complexity of MMI is typically 15 times greater than for (pruned) ML. We expect this factor to decrease on a word based task such as the DARPA RM task (because of its lower connectivity). Equation 1 and equation 3 are therefore unusual in that the derivatives can be calculated with only a small increase in computation in addition to that needed for the function value.

Another method of reducing the computational complexity is to remove all training utterances which are correctly recognised by the recognition HMM. For example, in connected digit recognition, high performance can be achieved allowing a large number of strings to be discarded from the training set and thereby drastically reducing training times[6]. However, this method is of little use for phoneme recognition where very few (if any) training utterances will be correctly recognised.

7. PHONEME RECOGNITION EXPERIMENTS

We have performed a number of experiments using the TIMIT database, in order to investigate the accuracy of MMI on a difficult task with a realistically large training set.

The extended BW algorithm was used for all the MMI experiments.

During training and testing a bigram language model was imposed. The language model probabilities were squared before combining with the acoustic probabilities supplied by the HMMs. This was empirically determined to improve performance. No optimisation of the language model which was estimated from the training set transcriptions was attempted.

The experimental conditions are similar to those established by Lee and Hon in their benchmark experiments[2] on the TIMIT database. The training set consists of all *si* and *sx* sentences and the test set consists of 336 *si* and *sx* sentences chosen at random. The results are tabulated in Tables 1 and 2 below Twelve iterations of BW were used for the ML experiments. As theory predicts, the likelihood always increased from iteration to iteration. The recognition performance on the test set, however, usually peaked after about 4 iterations and then varied randomly usually well within 1 percent. Twelve iterations of extended BW were also used for the MMI experiments. The conditional cross entropy always increased for the first six iterations but after that it would occasionally decrease. For MMI, the recognition performance often peaked after about 8 iterations and then again varied randomly usually well within 1 percent.

	ML		MMI	
#Mix	% Acc	% Corr	% Acc	% Corr
1	52.72	57.18	60.07	65.43
2	56.70	60.85	62.45	67.77
4	60.09	64.38	65.46	70.04
8	63.69	67.44	67.36	71.94
16	66.07	69.68	67.50	73.53

Table 1: Results for Diagonal Covariance Models

	ML		MMI	
#Mix	% Acc	% Corr	% Acc	% Corr
1	60.24	64.14	66.95	71.63
2	64.42	67.82	68.08	72.41
3	66.24	69.71	69.04	73.50
4	67.38	70.80	69.31	74.38

Table 2: Results for Full Covariance Models

Comparison with other systems is difficult because of slightly different test conditions. In these tests although the same final phoneme set as[2] was used, in our test, sequences of identical phonemes were forbidden, which was not done by Lee.

With this proviso, Lee achieved 66.1%/73.8% using 1450 right context-dependent phone models[2]; Young reported 59.9%/73.7% using 807 generalised triphone models[9], and Robinson 75.0%/78.6% using a recurrent error propagation networks[8].

Tests with a slightly different phoneme set, but with collapsing of phoneme sequences have been reported in [8, 10]. Robinson obtained 70.7%/74.3%, and Ljolje obtained 69.4%/74.8% who used 'Quasi-triphonic models' and a trigram grammar.

Note, however, that the system produced by Robinson, has the opportunity to learn the true TIMIT grammar. Its phonetic modelling ability is therefore difficult to gauge.

8. CONCLUSIONS

This paper has described an implementation of an MMI HMM-based phoneme recogniser and it has presented comparative phoneme recognition results for the TIMIT database.

A selection of different training algorithms were compared, and we found that the extended BW algorithm was the most efficient. However, we expect that future work in this area will be fruitful. For instance, by decreasing the step size we hope to improve our finite difference approximation to the diagonal hessian.

As can be seen MMI training does substantially improve recognition performance, but the improvement relative to the corresponding ML case decreases as the models complexity increases. The best results achieved are comparable to the best state-of-the-art systems including those which use large numbers of context-dependent models. An alternative view of the benefit of the MMI training method is that it allows the number of parameters needed to reach a certain level of performance to be reduced by around a factor of 4. This may be of considerable value for implementing real-time systems.

Finally, it should be noted that a large difference between training and test set accuracies for MMI models was observed. This suggests that if much larger training databases were available then the performance improvements obtainable from MMI would be even more significant.

9. ACKNOWLEDGEMENTS

S. Kapadia is supported by a CASE studentship and V. Valtchev is a Benefactors' student at St. John's College, Cambridge. The authors are grateful to section DAT13, British Telecom Research Laboratories for use of their computers. We would especially like to thank Geoffrey Walker for his assistance in porting the software.

10. REFERENCES

[1] PS Gopalakrishnan, D Kanevsky, A Nádas, D Nahamoo. *An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems.* IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol 37, No 1, pp107-113, 1991

[2] K-F Lee, H-W Hon. *Speaker-Independent Phone Recognition Using Hidden Markov Models.* IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol 37, No 11, pp1641-1648, 1989

[3] B Merialdo. *Phonetic Recognition using Hidden Markov Models and Maximum Mutual Information Training.* ICASSP Proceedings, pp111-114, 1988

[4] A Nádas. *Optimal Solution of a Training Problem in Speech Recognition.* IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol 31, No 4, pp814-817, 1983

[5] A Nádas, D Nahamoo, M A. Picheny. *On a Model-Robust Training Method for Speech Recognition.* IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol 36, No 9, pp1432-1435, 1988

[6] Y Normandin. *Hidden Markov Models, Maximum Mutual Information Estimation, and the Speech Recognition Problem.* Phd thesis, Department of Electrical Engineering, McGill University, Montreal, 1991,

[7] LR Rabiner. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, pp257-285, 1989

[8] AJ Robinson. *Several Improvements to a Recurrent Error Propagation Network Phone Recognition System.* Technical report CUED/F-INFENG/TR.82, Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, England, 1991

[9] SJ Young. *The General Use of Tying in Phoneme-based HMM Speech Recognisers.* ICASSP Proceedings, pp569-572, 1992

[10] A Ljolje. *New developments in phone recognition using an ergodic hidden markov model.* Technical memorandum TM-11222-910829-12, A T and T Bell Laboratories, 1991

[11] SE Fahlman. *An Empirical Study of Learning Speed in Back-Propagation Networks.* CMU-CS-88-162, 1988

[12] Peter F Brown *The Acoustic-Modelling problem in Automatic Speech Recognition.* Technical report RC 12750 (log #57380) Computer Science, IBM Thomas J. Watson Research Center, Distribution Services 73-F11, Post Office Box 218, Yorktown Heights, New York 10598, 1987