





MMIE training of large vocabulary recognition systems

V. Valtchev ¹, J.J. Odell ¹, P.C. Woodland ², S.J. Young *

University Engineering Department, Trumpington Street, Cambridge CB2 1PZ, United Kingdom Received 11 December 1996; revised 20 June 1997

Abstract

This paper describes a framework for optimising the structure and parameters of a continuous density HMM-based large vocabulary recognition system using the Maximum Mutual Information Estimation (MMIE) criterion. To reduce the computational complexity of the MMIE training algorithm, confusable segments of speech are identified and stored as word lattices of alternative utterance hypotheses. An iterative mixture splitting procedure is also employed to adjust the number of mixture components in each state during training such that the optimal balance between the number of parameters and the available training data is achieved. Experiments are presented on various test sets from the Wall Street Journal database using up to 66 hours of acoustic training data. These demonstrate that the use of lattices makes MMIE training practicable for very complex recognition systems and large training sets. Furthermore, the experimental results show that MMIE optimisation of system structure and parameters can yield useful increases in recognition accuracy. © 1997 Elsevier Science B.V.

Résumé

Cet article décrit un cadre théorique pour l'optimisation de la structure et des paramètres d'un système de reconnaissance de grands vocabulaires basé sur des HMMs avec des densités d'émission continues, utilisant le critère d'estimation de l'information mutuelle maximale (MMIE). Pour réduire la complexité calculatoire de l'algorithme d'apprentissage MMIE, les segments de parole pouvant être confondus sont identifiés et stockés sous forme de treillis de mots des diverses hypothèses de séquences. Une procédure itérative de division des gaussiennes est également employée pour ajuster, à chaque état, le nombre de composantes des mélanges pendant l'apprentissage afin d'obtenir le meilleur compromis entre le nombre de paramètres et le volume de données d'apprentissage disponible. On présente des résultats expérimentaux sur divers ensembles de test de la base de données "Wall Street Journal" qui utilisent jusqu'à 66 heures de données de parole pour l'apprentissage. Ces résultats démontrent que l'utilisation de treillis rend l'apprentissage MMIE utilisable pour des systèmes de reconnaissance très complexes et des ensembles d'apprentissage très grands. De plus, ils montrent que l'optimisation MMIE de la structure des systèmes et des paramètres peut aboutir à des améliorations utiles des performances de reconnaissance. © 1997 Elsevier Science B.V.

1. Introduction

Previous research has shown that the accuracy of an HMM-based speech recognition system trained using Maximum Likelihood Estimation (MLE) can often be improved further using discriminative train-

^{*} Corresponding author. E-mail: sjy@eng.cam.ac.uk.

1 Now with Entropic Cambridge Research Laboratory, Com-

pass House, 80-82 Newmarket Road, Cambridge CB5 8DZ, UK. E-mail: {vv1,jo}@entropic.co.uk.

² E-mail: pcw@eng.cam.ac.uk.

ing. For example, Maximum Mutual Information Estimation (MMIE) (Bahl et al., 1986) has been studied in the context of small vocabulary speech tasks and substantial gains in performance have been reported (Normandin et al., 1994a), especially where compact models are used with relatively few parameters (Kapadia et al., 1993). Other similar discriminative training methods such as corrective training (Bahl et al., 1993) and Minimum Classification Error/Generalised Probabilistic Descent (Juang and Katagiri, 1992; Rainton and Sagayama, 1992; McDermott and Katagiri, 1994) have shown similar promise.

Unfortunately, the discriminative optimisation of HMM parameters is much more complex than the conventional MLE framework. First, the discriminative nature of the objective function requires that acoustically confusable segments of speech be available to represent the errors made during recognition. Even in a small vocabulary task, the gathering of statistics about these confusable segments results in a dramatic increase in computational load compared to the corresponding MLE case. Second, given a state/frame alignment of the training data there are no closed form solutions for parameter estimates that maximise the objective function. Thus, whilst an MLE system can typically be trained in a few iterations, discriminative training may require considerably more.

The focus of the work reported here is to develop practical methods for applying discriminative training to large vocabulary continuous speech recognition (LVCSR) systems and study the effects of this training on model complexity and recognition accuracy. Since LVCSR systems typically have several million parameters and require very large training databases, the application of discriminative training techniques to LVCSR systems is computationally extremely challenging.

A recent trend in the design of LVCSR systems has been the inclusion of a mechanism to generate lattices encoding multiple recognition hypotheses (Aubert and Ney, 1995; Richardson et al., 1995; Woodland et al., 1995a). Although lattice generation was originally motivated by the need to allow multipass recognition strategies to be used, lattices are now widely used for system development where, used as a constraining word-graph, they allow rapid

recogniser testing. Since lattices contain the most likely word hypotheses as determined by a speech recogniser, they provide a compact encoding of confusion data and therefore offer a route towards making discriminative training of LVCSR systems practicable.

This paper explores the use of lattices for the discriminative optimisation of HMM-based LVCSR systems (Valtchev et al., 1996). The focus is on MMI training but the general computational framework is equally applicable to other forms of discriminative training such as those mentioned earlier. The experimental work is based around the HTK Large Vocabulary recogniser which is a continuous-density tied-state cross-word triphone system (Woodland et al., 1994, 1995a,b). The HTK LVCSR is built incrementally. First, a set of conventional single-Gaussian triphones are estimated, then the model states are clustered using phonetic decision trees. The output distributions of the resulting tied-states are then converted to Gaussian mixtures using an iterative mixture splitting procedure. Normally, all states have the same number of mixture components, but it is also possible to split mixtures using a discriminative criterion (Normandin, 1995) and this is also investigated here.

The paper is organised as follows. Section 2 reviews the MMI-based parameter re-estimation formulae and the discriminative mixture-splitting procedure. Section 3 then describes the implementation of this MMI training scheme using lattices. Section 4 presents experimental results on using MMI-training to optimise the parameters of the HTK LVCSR system using the Wall Street Journal SI-284 acoustic training data (Kubala et al., 1994). Performance is evaluated on the 1994 CSR NAB Hub 1 development and evaluation data (Pallett et al., 1995) and the SQALE American English evaluation data (Steeneken and van Leeuwen, 1995). Finally, Section 5 presents our overall conclusions on the work.

2. MMI-based parameter optimisations

2.1. Re-estimation of HMM parameters

Conventional Maximum Likelihood Estimation (MLE) attempts to increase the a posteriori probabil-

ity of the training data given the model sequence corresponding to the data. The models from other classes do not participate in the parameter re-estimation, hence, the MLE criterion does not relate directly to the objective of reducing the recognition error rate.

MMIE training was proposed in (Bahl et al., 1986) as an alternative to MLE. The method attempts to increase the a posteriori probability of the model sequence corresponding to the training data given the training data. For R training observations $\{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_r, \ldots, \mathcal{O}_R\}$ with corresponding transcriptions $\{w_r\}$, the MMIE objective function is given by

$$\mathscr{F}(\lambda) = \sum_{r=1}^{R} \log \frac{P_{\lambda}(\mathscr{O}_{r}|\mathscr{M}_{w_{r}})P(w_{r})}{\sum_{\hat{w}} P_{\lambda}(\mathscr{O}_{r}|\mathscr{M}_{\hat{w}})P(\hat{w})}, \tag{1}$$

where \mathcal{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 Eq. (1) is taken over all possible word sequences \hat{w} allowed in the task and it can be replaced by

$$P_{\lambda}(\mathscr{O}_{r}|\mathscr{M}_{gen}) = \sum_{\hat{w}} P_{\lambda}(\mathscr{O}_{r}|\mathscr{M}_{\hat{w}}) P(\hat{w}), \qquad (2)$$

where \mathcal{M}_{gen} is a model constructed such that for all paths in every $\mathcal{M}_{\hat{w}}$ there is a corresponding path of equal probability in \mathcal{M}_{gen} . It is usually assumed that the model used for recognition is a reasonable approximation of \mathcal{M}_{gen} .

MMIE training can thus be interpreted as a two stage optimisation process. The first stage is equivalent to performing MLE training such that the HMM parameters are adapted to increase the numerator term $P_{\lambda}(\mathcal{O}_r|\mathcal{M}_{w_r})$. In the second stage, the HMM parameters are changed in the opposite direction in order to minimise the denominator term $P_{\lambda}(\mathcal{O}|\mathcal{M}_{\text{gen}})$. The second step dominates the computation and this will depend on the size of the vocabulary, the grammar and any contextual constraints. In many practical situations, for example where cross-word context dependent acoustic models are used in conjunction with a long span language model, the construction of a complete model for \mathcal{M}_{gen} is intractable.

The MMIE objective function given in Eq. (1) can be optimised by any of the standard gradient methods, however, such approaches are often slow to

converge. Analogous to the Baum-Welch algorithm for MLE training, Gopalakrishnan et al. have shown that a re-estimation formulae of the form

$$\hat{\lambda}_{i,j} = \frac{\lambda_{i,j} \left(\frac{\partial \mathscr{F}}{\partial \lambda_{i,j}} + D \right)}{\sum_{k} \lambda_{i,k} \left(\frac{\partial \mathscr{F}}{\partial \lambda_{i,k}} + D \right)}$$
(3)

will converge to give a local optimum of $\mathcal{F}(\lambda)$ for a sufficiently large value of the constant D (Gopalakrishnan et al., 1989).

For a continuous density HMM system, Eq. (3) does not lead to a closed form solution for the means and variances. However, Normandin has shown that using Eq. (3) and a discrete approximation to a Gaussian distribution leads to the following re-estimation formulae where $\mu_{j,m}$ and $\sigma_{j,m}^2$ are the means and (diagonal) variances for each state j and mixture component m (Normandin, 1991):

$$\hat{\mu}_{j,m} = \frac{\left\{\theta_{j,m}(\mathscr{O}) - \theta_{j,m}^{\text{gen}}(\mathscr{O})\right\} + D\mu_{j,m}}{\left\{\gamma_{i,m} - \gamma_{i,m}^{\text{gen}}\right\} + D},\tag{4}$$

$$\hat{\sigma}_{j,m}^{\,2} = \frac{\left\{\theta_{j,m}(\mathscr{O}^2) - \theta_{j,m}^{\,\mathrm{gen}}(\mathscr{O}^2)\right\} + D\left(\sigma_{j,m}^2 + \mu_{j,m}^2\right)}{\left\{\gamma_{j,m} - \gamma_{j,m}^{\,\mathrm{gen}}\right\} + D}$$

$$-\hat{\mu}_{i,m}^2. \tag{5}$$

In these equations, $\theta_{j,m}(x)$ represents the sum of all training data x weighted by the probability of occupying component m of state j when x occurs, i.e.,

$$\theta_{j,m}(x) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} x^r(t) \gamma_{j,m}^r(t),$$
 (6)

where $\gamma_{j,m}^r(t)$ is the probability of occupying component m of state j at time t and $x^r(t)$ is the data from observation r at time t. Similarly, $\gamma_{j,m}$ represents the corresponding component occupation counts summed over all the data, i.e.,

$$\gamma_{j,m} = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{j,m}^r(t).$$
 (7)

The superscript gen indicates the corresponding quantities computed using the model \mathcal{M}_{gen} . Note that although there is no proof that these re-estimation formulae converge, they appear to work well in practice (Kapadia et al., 1993; Normandin, 1991).

A re-estimation formula for the mixture weight parameters $c_{i,m}$ follows directly from Eq. (3):

$$\hat{c}_{j,m} = \frac{c_{j,m} \left\{ \frac{\partial \mathscr{F}}{\partial c_{j,m}} + C \right\}}{\sum_{\hat{m}} c_{j,\hat{m}} \left\{ \frac{\partial \mathscr{F}}{\partial c_{j,\hat{m}}} + C \right\}}.$$
(8)

However, the derivative

$$\frac{\partial \mathscr{F}}{\partial c_{j,m}} = \frac{1}{c_{j,m}} \left(\gamma_{j,m} - \gamma_{j,m}^{\text{gen}} \right) \tag{9}$$

is extremely sensitive to small-valued parameters. As a consequence, Merialdo suggested the following more robust approximation (Merialdo, 1988):

$$\frac{\partial \mathscr{F}}{\partial c_{j,m}} \approx \frac{\gamma_{j,m}}{\sum_{\hat{m}} \gamma_{j,\hat{m}}} - \frac{\gamma_{j,m}^{\text{gen}}}{\sum_{\hat{m}} \gamma_{j,\hat{m}}^{\text{gen}}}$$
(10)

and this also appears to work well in practice (Kapadia et al., 1993; Normandin, 1991). The constant C is chosen such that all parameter derivatives are positive.

2.2. Setting the constant D

The speed of convergence of MMI-based optimisation using Eqs. (4) and (5) is directly related to the value of the constant D. Small D results in a larger step size and hence a potentially faster rate of convergence. However, using small values of D typically results in oscillatory behaviour which reduces the rate of convergence, and if D is very small, then the optimisation will become unstable. In practice a useful lower bound on D is the value which ensures that all variances remain positive.

Substituting Eq. (4) into Eq. (5) for the estimate of the mean gives

$$\hat{\sigma}_{j,m}^{2} = \frac{\overline{\theta}_{j,m}(\mathscr{O}^{2}) + D(\sigma_{j,m}^{2} + \mu_{j,m}^{2})}{\overline{\gamma}_{j,m} + D}$$
$$-\left\langle \frac{\overline{\theta}_{j,m}(\mathscr{O}) + D\mu_{j,m}}{\overline{\gamma}_{j,m} + D} \right\rangle^{2} > 0, \tag{11}$$

where for clarity the following substitutions have been used:

$$\begin{split} & \overline{\theta}_{j,m}(\mathscr{O}) = \theta_{j,m}(\mathscr{O}) - \theta_{j,m}^{\text{gen}}(\mathscr{O}), \\ & \overline{\theta}_{j,m}(\mathscr{O}^2) = \theta_{j,m}(\mathscr{O}^2) - \theta_{j,m}^{\text{gen}}(\mathscr{O}^2), \\ & \overline{\gamma}_{i,m} = \gamma_{i,m} - \gamma_{i,m}^{\text{gen}}. \end{split}$$

The above expression can be rearranged to give a quadratic inequality with respect to D in the form

$$aD^2 + bD + c > 0. (12)$$

Since a is positive, the inequality (12) is valid when $D \in (-\infty; D_1] \cup [D_2; +\infty)$, where D_1 and D_2 are the roots of the quadratic equation $aD^2 + bD + c = 0$. Hence, an appropriate value of D can be found by solving the system of quadratic inequalities for the given HMM set and choosing a small positive number from the resulting interval.

Preliminary experiments were performed in order to establish and tune the convergence properties of the MMI re-estimation algorithm. The use of a global constant *D* was compared with the use of phonespecific constants (47 different constants in total) such that the variance parameters for all triphones of each phone were positive. In these experiments, phone-specific constants were found to improve the convergence rate by a factor of two or more and, hence, phone-specific constants were used in all subsequent experiments.

2.3. Discriminative mixture splitting

As noted in the Introduction, the HTK LVCSR system uses mixtures of Gaussian densities to model the acoustic data. The use of such densities has several advantages, most important of which is the ability to model the data associated with each HMM state to arbitrary precision. Thus, the art of building a good acoustic model often translates into finding the right balance between the number of free parameters in the system and the amount of training data available. To achieve good recognition performance, a large number of mixture components are needed to adequately model the acoustic variability across different samples of the same speech sound (resolution). Conversely, the number of mixture components must be small enough to allow for reliable estimates of the

Gaussian parameters and mixture component weights (reliable parameter estimation).

In the HTK LVCSR system, the optimal balance between resolution and reliable parameter estimation is achieved through parameter sharing. Phonetic decision trees are used to cluster states and then the states in each cluster are tied to use the same mixture distribution (Young et al., 1994). In the standard HTK LVCSR, the number of mixture components per output distribution is fixed at a value determined empirically, by starting off with a single component per state and gradually increasing this number until the performance of the system on a validation test set no longer improves.

Alternatively, the number of mixture components in the output distributions can be varied. Within an MLE-based estimation framework, the number of components chosen to represent each state can only depend on the amount of available training data and in practice, this can lead to a sub-optimal system. ³ However, MMIE offers a means of determining the number of components per mixture using a criterion which directly relates to discrimination.

The splitting of mixture components according to the MMIE criterion was first proposed by Normandin (1995). His algorithm uses the derivatives of the objective function with respect to the mixture component weights to decide if increased resolution is needed. The parameter derivatives are given by

$$\frac{\partial \mathcal{F}}{\partial c_{j,m}} = \sum_{r=1}^{R} \left\{ \frac{1}{P_{\lambda}(\mathscr{O}_{r}|\mathscr{M}_{w_{r}})} \frac{\partial P_{\lambda}(\mathscr{O}_{r}|\mathscr{M}_{w_{r}})}{\partial c_{j,m}} - \frac{1}{P_{\lambda}(\mathscr{O}_{r}|\mathscr{M}_{gen})} \frac{\partial P_{\lambda}(\mathscr{O}_{r}|\mathscr{M}_{gen})}{\partial c_{j,m}} \right\}. (13)$$

The first term on the right-hand side of Eq. (13) is the occupancy count for mixture component m in state j of the model, produced by aligning the training utterances against the model sequence corresponding to the correct transcription. Similarly, the second term is the occupancy count from aligning

the training data against the recognition model. A positive $\partial \mathcal{F}/\partial c_{j,m}$ indicates that the component occupation accumulator was often updated during the alignment of the data against the correct transcription. At the same time, the equivalent path in the recognition model contributed very little due to its relatively low likelihood compared to other competing paths. This clearly constitutes a discrimination problem which can be corrected by increasing the number of mixture components.

The discriminative mixture splitting algorithm can be summarised as follows:

- 1. Compute the mixture component occupancy counts according to Eq. (13);
- 2. Re-estimate the parameters of the mixture distribution using Eqs. (4), (5) and (8);
- 3. Split the mixture components with the top n largest positive derivative values (occupancy counts). This is accomplished by cloning the reestimated distribution and perturbing each resulting pair of Gaussians by ± 0.2 standard deviations.

3. Discriminative training using lattices

3.1. Lattice generation and use

The HTK LVCSR system employs a time-synchronous one-pass decoder that uses a dynamically built tree-structured recognition network (Odell et al., 1994). This approach allows the integration of cross-word context-dependent acoustic models and any N-gram language model directly within the search. However, despite the relatively high efficiency of the decoder, the recognition process is still computationally expensive to perform. During development it is often necessary to perform several recognition trials on a development test set in order to determine the best recogniser set-up. System development such as this can be speeded-up greatly by producing a lattice of likely hypotheses for each utterance instead of finding the single most likely transcription.

In the HTK system (Woodland et al., 1995a), lattices can be generated as a by-product of the recognition process. Each lattice consists of a set of

³ Possibly because mixture densities with a large number of components may allocate many of them to model intricate parts of the interior regions of the underlying distributions, thus not directly aiding discrimination.

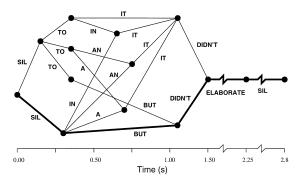


Fig. 1. An example lattice. The path shown in bold corresponds to the correct transcription of the utterance.

nodes that correspond to particular instants in time, and arcs connecting these nodes to represent possible word hypotheses (see Fig. 1). Associated with each arc is an acoustic log likelihood and a language model log likelihood.

In subsequent recognition passes, the lattices can be used to constrain the decoder in terms of limiting the number of words needed to be hypothesised at any decision point in the recognition. For instance, an initial pass using triphone models and a bigram language model can be used to generate a set of lattices. In subsequent evaluations, these lattices can be re-scored using a more complex language model and more detailed acoustic models which may require too much computation to use in a single pass.

Similar considerations concerning computational efficiency apply to any discriminative training procedure. Several training iterations are performed over the same training set in order to optimise the chosen objective criterion. A word lattice forms a compact representation of many different sentence hypotheses and hence provides an efficient representation of the confusion data needed for discriminative training (Normandin et al., 1994b).

3.2. MMI training using lattices

Referring again to Eqs. (4), (5) and (8), it can be seen that the main computational requirement is to compute the probability of the model being in each state j and component m at each time t. In conventional Baum–Welch re-estimation for MLE training,

these so called *occupation* probabilities are computed using the forward–backward algorithm and the composite HMM \mathcal{M}_{w_r} constructed from the transcription of each training utterance r. These occupation probabilities are also needed for MMI training. However, in MMI training, the analogous probabilities computed using the recognition model \mathcal{M}_{gen} are also needed.

If a lattice is assumed to contain word sequences corresponding to all of the high likelihood state/component alignments, then the forward-backward algorithm can be applied to the lattice to compute an estimate of the occupation probabilities. Furthermore, if lattices are produced for both \mathcal{M}_{w_p} and \mathcal{M}_{gen} , then the same computational procedure can be used for both. Owing to their role in the objective function given in Eq. (1), these are referred to as the numerator and denominator lattices, respectively. Note that using a lattice for the numerator allows pronunciation alternatives to be included in \mathcal{M}_{w_p} .

Using lattices, the discriminative training algorithm can be summarised as follows:

- 1. Generate a pair of *numerator* and *denominator* lattices for each utterance in the training data, these correspond to \mathcal{M}_{w_r} and \mathcal{M}_{gen} , respectively. The numerator lattice is produced by aligning the acoustic data against a network of HMMs built according to the known transcription. The denominator lattice corresponds to running an unconstrained recognition pass. In both cases an appropriate N-gram language model is used.
- 2. For each training utterance, the numerator or denominator lattice is loaded into the recogniser and reduced to a word graph in which the acoustic scores are discarded and the network compressed by removing duplicate word sequences. Constrained recognition is performed using the current HMM set and the language model scores from the word graph. A new output lattice is then produced containing the original language model scores and new acoustic scores.
- 3. For each node in the lattice, the forward $(\overline{\alpha})$ and the backward $(\overline{\beta})$ lattice probabilities are computed. In analogy to Baum–Welch re-estimation, the forward probabilities are computed in a recursive fashion starting from the beginning of the lattice. For node l and preceding words $w_{k,l}$

spanning nodes k to l, the forward probability is given by

$$\overline{\alpha}_{l} = \sum_{k} \overline{\alpha}_{k} P_{\text{acoust}}(w_{k,l}) P_{\text{lang}}(w_{k,l}), \qquad (14)$$

where P_{acoust} is the maximum likelihood of word $w_{k,l}$ hypothesised between the time instances corresponding to nodes k and l, and P_{lang} is the language model probability of $w_{k,l}$. The backward probabilities $\overline{\beta}_k$ are computed in a similar fashion starting from the end of the lattice.

4. For each pair of nodes k and l, the corresponding $\overline{\alpha}_k$ and $\overline{\beta}_l$ are propagated into the sequence of model instances corresponding to word $w_{k,l}$, and statistics are accumulated. For simplicity, statistics are accumulated across the best state sequence between nodes k and l. Thus, the posterior occupation probability remains constant within each word and is given by

$$\overline{\gamma}_{k,l} = \frac{\overline{\alpha}_k P_{\text{acoust}}(w_{k,l}) P_{\text{lang}}(w_{k,l}) \overline{\beta}_l}{\mathscr{L}}, \qquad (15)$$

where \mathscr{L} is the total lattice likelihood computed by summing the forward probabilities over all hypothesised end words. Given $\overline{\gamma}_{k,l}$, a Viterbi alignment is then performed for the arc spanning node k to node l using the acoustic model for word $w_{k,l}$ to give the required component occupation probabilities $\gamma_{j,m}(t)$ for each HMM state jand component m within the model.

- 5. The state/component occupation probabilities computed in step 4 for each of the numerator and denominator lattices enable the statistics $\theta_{j,m}(\mathscr{O})$, $\theta_{j,m}^{\text{gen}}(\mathscr{O})$, $\theta_{j,m}^{\text{gen}}(\mathscr{O}^2)$, $\theta_{j,m}^{\text{gen}}(\mathscr{O}^2)$, $\gamma_{j,m}$ and $\gamma_{j,m}^{\text{gen}}$ defined in Section 2.1 to be accumulated over all training utterances.
- 6. When all training utterances have been processed, new parameter estimates are calculated according to Eqs. (4), (5) and (8) with *D* being adjusted on a per-phone basis according to the procedure described in Section 2.2.

In order to save computation, each successive reestimation cycle is repeated from step 2 rather than from step 1 on the assumption that the high likelihood state/component alignments encoded within the initial set of lattices do not change during training.

The computational steps involved in the imple-

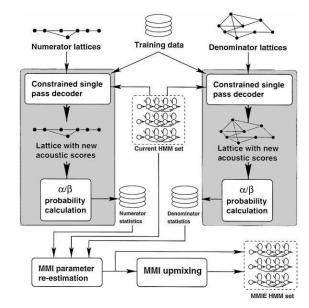


Fig. 2. Discriminative training framework for LVCSR systems.

mentation of this discriminative training procedure are illustrated in Fig. 2. The top left and right branches of the diagram show the calculation of statistics for the numerator/denominator parts of the MMIE objective function, respectively. For each training utterance, the numerator or denominator lattice is loaded into the recogniser and reduced to a word graph. Recognition is performed using the current HMM set and the language model scores from the word graph. A new output lattice is then produced containing the original language model scores and new acoustic scores. This is followed by the computation of forward $(\bar{\alpha})$ and backward $(\bar{\beta})$ probabilities for each node in the lattice. In a postprocessing step, the two sets of statistics are combined to calculate new parameter values according to Eqs. (4), (5) and (8). This is optionally followed by an up-mixing procedure whereby component occupancy statistics are used to split selected mixture components as described in Section 2.3.

4. Experiments

4.1. Lattice generation

For the experiments reported here, lattices were generated by the HTK LVCSR system using state-

clustered, mixture-Gaussian, cross-word triphones and a back-off bigram language model. Each frame of speech is represented by a 39-dimensional feature vector that consists of 12 mel-frequency cepstral coefficients, normalised log energy and the first and second differentials of these values. The state clustering algorithm uses decision trees built for every monophone HMM state to determine equivalence classes between sets of triphone contexts. This is followed by the application of an iterative uniform mixture splitting and retraining sequence which allows the optimal match between system complexity and available training data to be found (Young et al., 1994).

For the lattice generation on the training data, a 65k word list was created by adding the words occurring in the training set to our standard WSJ recognition lexicon (Woodland et al., 1995b). A corresponding bigram back-off language model was then constructed to accommodate the SI-284 training set which contains utterances with both verbalised and non-verbalised punctuation. The language model (train_bg65k) contained 4.2 million bigrams estimated from the 1994 North American Business News text corpus (NAB94) of 227 million words. The gender independent HMMs used for lattice generation were trained on the combined WSJ0 and WSJ1 portions (~66 hours total) of the WSJ database using the 1993 LIMSI WSJ Lexicon and phone set. They had a total of 6399 tied-states and each output distribution had 12 mixture components. This model set is referred to as HMM1. More details of its construction and its performance on various WSJ test sets are given in (Woodland et al., 1995a).

Lattices were generated for all of the 36,441 training utterances used from the SI-284 set. Table 1 gives an indication of the quality of these lattices in terms of word/sentence error rate and lattice density. The lattice density figure is the average number of lattice arcs (representing words) per spoken word.

Table 1 Lattice densities and % lattice sentence/word error rates

Lattice set	Lattice density	Lattice %SER	Lattice %WER
Numerator	1.7	0.0	0.0
Denominator	14.9	14.8	1.2

The lattice sentence error rate (%SER) relates to whether a path corresponding to the correct sentence transcription exists in the lattice. The lattice word error rate (%WER) is a lower bound on the word error rate which can be obtained by rescoring the lattice. To speed up development time, a tight pruning beam was used resulting in the average lattice density figure of 14.9 shown for the denominator lattices.

The relatively large density of 1.7 for the numerator lattices is due to the existence of multiple pronunciations for many of the most common words and due to the fact that two types of inter-word silence are allowed: one short silence which does not affect cross-word triphone context and one longer silence which blocks cross-word triphone context. However, the density of the numerator lattice has no effect on the training algorithm since it is reduced to a word graph prior to recomputing the acoustic scores, and hence it is effectively equivalent to doing a forced alignment as in our conventional MLE training.

Finally note that, as indicated by the lattice %SER figure in Table 1, some of the denominator lattices were found not to contain the correct transcription of the utterance. To solve this problem, the corresponding numerator and denominator lattices were merged together to form a new set of denominator lattices which were then used for the subsequent MMIE training.

4.2. Test sets used

Recognition experiments were performed on the 1994 ARPA Hub-1 development and evaluation test sets, and on the 1995 European SQALE project American English evaluation test set. Details of these test sets and the Out-Of-Vocabulary (OOV) rates obtained with a 65k word vocabulary are given in Table 2

Recognition results using the csrnabl_dt and csrnabl_et test sets were computed using the non-adjudicated reference transcriptions and word-mediated string alignment. The results for the sqale_et test set used the adjudicated transcriptions.

Table 2
Details of the recognition test sets used in the MMIE experiments, the OOV rate corresponds to the 65k word vocabulary throughout

Test set	Task	#Utter.	#Spkrs.	OOV rate
csrnab1_dt	1994 Hub-1 development	310	20	0.31%
csrnabl_et	1994 Hub-1 evaluation	316	20	0.65%
sqale_et	SQALE American Eng.	200	20	0.39%

4.3. Performance of MMI-trained systems

The HMM1 system was tested with bigram (bg), trigram (tg) and fourgram (fg) language models estimated from the NAB94 text corpus. Initially, two versions of the system were evaluated, HMM1-2 and HMM1-12, with 2 and 12 mixture components per state respectively. Both systems were optimised using four iterations of MMIE training. Table 3 gives results on the WSJ test sets obtained using the HMM1-12 system. The first row in Table 3 gives the performance of the system on the training set using the train bq65k bigram LM. Consistent with previous MMIE results, the word error rate on the training data reduces substantially from 8.1% to 3.5%. The following three rows show the performance of the system on the sqale_et test set with bigram, trigram and fourgram language models, respectively. In all cases, the MMIE training has resulted in improved recognition performance.

Table 3 Word (%WER) and sentence (%SER) error rates on the SI-284 training set and the WSJ test sets using the HMM1-12 system trained using MLE or MMIE

united using type of thirties					
		MLE		MMIE	
Data set	LM	%WER	%SER	%WER	%SER
SI-284	bg	8.1	58.8	3.5	34.1
sqale_et	bg	12.6	77.0	11.9	73.5
sqale_et	tg	9.0	60.0	8.2	59.5
sqale_et	fg	7.9	56.0	7.4	55.0
csrnab1_dt	bg	12.75	76.77	12.49	75.16
csrnab1_dt	tg	9.49	68.39	9.43	65.16
csrnab1_et	bg	13.43	82.28	12.64	79.75
csrnabl_et	tg	10.01	72.47	10.13	72.15

Table 4
Word and sentence error on the WSJ test sets using the HMM1-2
system trained using MLE or MMIE with bigram and trigram
language models

		MLE		MMIE	
Data set	LM	%WER	%SER	%WER	%SER
sqale_et	bg	17.4	82.0	15.7	78.0
sqale_et	tg	12.7	71.0	10.8	61.0
csrnab1_dt	bg	15.6	83.9	14.3	81.3
csrnab1_dt	tg	11.9	74.8	10.9	70.7
csrnab1_et	bg	17.4	86.4	16.0	84.2
csrnab1_et	tg	13.7	80.7	12.5	78.2

The remaining part of Table 3 gives recognition results on the csrnab1_dt and csrnab1_et test sets with bigram and trigram LMs. A small improvement in recognition accuracy is visible on the csrnab1_dt set for both the bigram and the trigram LMs. On the csrnab1_et set, the word error rate is reduced by 6% in the bigram case, however, with the trigram LM the MMIE result is marginally worse than the corresponding MLE case. The latter result suggests that performance gains from MMIE can diminish with the use of a more complex language model which does not match the one used during training.

Table 4 gives results on the three test sets obtained using the HMM1-2 system with bigram and trigram LMs. In all cases MMIE has provided improved recognition performance. Table 5 summarises the performance of variants of the HMM1 system using different number of mixture components per state and a bigram language model.

Overall, the results demonstrate that MMIE can provide a worthwhile improvement in the perfor-

Table 5 % word error rates on the various test sets using MLE and MMIE-trained variants of the HMM1 system with a varying number of mixture components and a bigram language model

No. mix.	sqale	sqale_et		csrnab1_dt		csrnabl_et	
comps	MLE	MMIE	MLE	MMIE	MLE	MMIE	
1	19.12	16.46	18.22	15.24	20.44	18.05	
2	17.36	15.70	15.58	14.31	17.44	15.96	
4	15.26	14.38	14.98	14.06	15.32	14.54	
12	12.60	11.90	12.75	12.49	13.43	12.64	

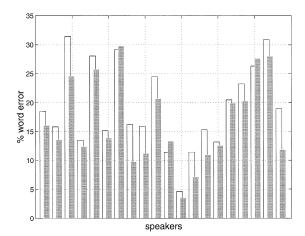


Fig. 3. Per speaker word error rate for the HMM1-2 system and a bigram LM on the sqale_et test set. The bars in white/grey correspond to the MLE/MMIE case, respectively.

mance of all systems but larger performance gains will be observed on the smaller model sets. Fig. 3 gives per speaker results for the two mixture component HMM1-2 system. The graph shows that for three speakers the word error rate is marginally worse than the corresponding MLE-trained system.

The plot in Fig. 4 shows the typical change in the value of the objective function at each iteration for the single Gaussian HMM1-1 system together with the recognition performance in terms of word error rate on the training set SI-284 and the sqale_et

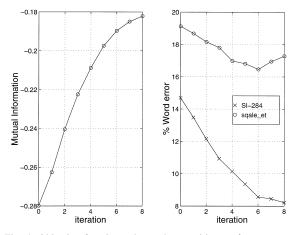


Fig. 4. Objective function value and recognition performance on the SI-284 training set and the sqale_et test set using the single Gaussian HMM1-1 system with bigram LMs.

test set with bigram LM. As the value of the objective function improves so does the performance on the training set. However, the performance on the test sets starts to deteriorate after the 6th iteration which can be attributed to an over-adaptation to the training data.

Finally, it may be of interest to note that the computation required for MMIE training with a denominator lattice density of approximately 15 was found to be around 10–12 times greater per iteration than that needed for standard MLE training.

4.4. MMI-based mixture splitting experiments

To test the effectiveness of MMI-based mixture component splitting, the single Gaussian HMM1-1 system was up-mixed in stages to produce two variable mixture variants HMM1-V4 (up to 4 mixture components per state) and HMM1-V16 (up to 16 mixture components per state). The total number of parameters in each of these two HMM sets are equal to the fixed 2 and 4 mixture component systems listed in Table 5 and reproduced in Table 6 as HMM1-F2 and HMM1-F4, respectively. The training patterns for these systems were HMM1-V4: tut-tu-t-t and HMM1-V16: tu-t-tu-t-tut-tu-t-t, where t denotes a MMIE pass and tu denotes an MMIE pass followed by up-mixing. Table 6 gives the performance of these systems on the WSJ test sets.

On all test sets the performance of the HMM1-V4 system is consistently better than the MMIE-trained HMM1-2 which is equivalent in the sense that it has the same number of parameters overall. Similarly, the HMM1-V16 system has provided improved recognition performance of 3–5% when compared to a 4 mixture component system with the same number of parameters. Thus it would appear that MMI-

Table 6 MMIE-trained versions of the HMM1 system and a bigram language model

System	Equiv.	sqale_et	csrnab1_dt	csrnabl_et
HMM1-F2	2 mix	15.70	14.31	15.96
HMM1-V4	2 mix	14.85	13.94	15.16
HMM1-F4	4 mix	14.38	14.06	14.54
HMM1-V16	4 mix	13.65	13.44	14.04

based discriminative mixture splitting is an effective way of efficiently allocating Gaussian components to states in a continuous density HMM system.

5. Conclusions

This paper has described an implementation of the MMIE discriminative training and mixture splitting algorithm based on the use of lattices to compactly represent confusable segments of data. It has been demonstrated that this approach makes it feasible to apply MMIE training to very large HMM-based recognition systems. Furthermore, the re-estimation formulae used previously for small systems have been shown to give good convergence on large systems provided that the learning rate constants *D* for the mean and variance parameters are set on a per phone basis.

Experimental results using the full SI-284 speaker independent training set from the Wall Street Journal database show, as expected, that the proposed method is very effective in reducing the word error rate on the training set. Results on unseen test data show a reduction in word error rate of 5–16% following MMIE training. In addition, the splitting of mixture components based on the MMI criterion can yield further gains in recognition accuracy and/or provide an effective mechanism for constructing compact and "parameter-efficient" HMM sets.

Acknowledgements

This work is in part supported by a UK Engineering and Physical Sciences Research Council grant GR/J10204. Additional computational resources were provided by the ARPA CAIP computing facility at Rutgers University, NJ.

References

- Aubert, X., Ney, H., 1995. Large vocabulary continuous speech recognition using word graphs. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Detroit, Vol. 1, pp. 49–52.
- Bahl, L.R., Brown, P.F. de Souza, P.V., Mercer, R.L., 1986.

- Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Tokyo, pp. 49–52.
- Bahl, L.R., Brown, P.F., de Souza, P.V., Mercer, R.L., 1993. Estimating hidden Markov model parameters so as to maximise speech recognition accuracy. Trans. IEEE Speech Audio Process. 1 (1), 77–83.
- Gopalakrishnan, P.S., Kanevsky, D., Nadas, A., Nahamoo, D., 1989. A generalisation of the Baum algorithm to rational objective functions. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Glasgow, S12.9, pp. 631–634.
- Juang, B.-H., Katagiri, S., 1992. Discriminative training. J. Acoust. Soc. Japan 13 (6), 333–340.
- Kapadia, S., Valtchev, V., Young, S.J., 1993. MMI training for continuous parameter recognition of the TIMIT database. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Minneapolis, pp. 491–494.
- Kubala, F., Bellegarda, J., Cohen, J., Pallett, D., Paul D., Phillips, M., Rajesekaran, R., Richardson, F., Riley, M., Rosenfeld, R., Roth, R., Weintraub, M., 1994. The hub and spoke paradigm for CSR evaluation. In: Proc. Spoken Language Technology Workshop. Morgan Kaufmann, Plainsboro, NJ, pp. 37–42.
- McDermott, E., Katagiri, S., 1994. Prototype-based minimum classification error/generalised probabilistic descent training for various speech units. Computer Speech and Language 8 (4), 351–368.
- Merialdo, B., 1988. Phonetic recognition using hidden Markov models and maximum mutual information training. In: Proc. Internat. Conf. Acoust. Speech Signal Process., New York, Vol. 1, pp. 111–114.
- Normandin, Y., 1991. Hidden Markov models, maximum mutual information estimation, and the speech recognition problem. Ph.D. Thesis, Dept. of Elect. Eng., McGill University, Montreal.
- Normandin, Y., 1995. Optimal splitting of HMM Gaussian mixture components with MMIE training. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Detroit, Vol. 1, pp. 449–452.
- Normandin, Y., Cardin, R., De Mori, R., 1994a. High-performance digit recognition using maximum mutual information. IEEE Trans. Speech Audio Process. 2 (2), 299–311.
- Normandin, Y., Lacoutre, R., Cardin, R. 1994b. MMIE training for large vocabulary continuous speech recognition. In: Proc. ICSLP, Yokohama, Vol. 3, pp. 1367–1370.
- Odell, J.J., Valtchev, V., Woodland, P.C., Young, S.J., 1994. A one-pass decoder design for large vocabulary recognition. In: Proc. Human Language Technology Workshop. Morgan Kaufman, Plainsboro, NJ, pp. 405–410.
- Pallett, D.S., Fiscus, J.G., Fisher, W.M., Garofolo, J.S., Lund, B.S., Martin, A., Pryzybocki, M., 1995. The 1994 benchmark tests for the ARPA spoken language program. In: Proc. Spoken Language Technology Workshop. Morgan Kaufmann, Austin, TX, pp. 5–38.
- Rainton, D., Sagayama, S., 1992. Appropriate error criterion for continuous speech HMM minimum error training. In: Proc. Internat. Conf. on Spoken Language Processing, Banff, Canada, pp. 233–236.
- Richardson, F., Ostendorf, M., Rohlicek, J.R., 1995. Lattice-based

- search strategies for large vocabulary recognition. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Detroit, Vol. 1, pp. 576–579.
- Steeneken, H.J.M., van Leeuwen, D.A., 1995. Multilingual assessment of speaker independent large vocabulary speech recognition systems: The SQALE project. In: Proc. Eurospeech, Madrid, pp. 1271–1274.
- Valtchev, V., Woodland, P.C., Young, S.J., 1996. Lattice-based discriminative training for large vocabulary speech recognition. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Atlanta, Vol. 2, pp. 605–608.
- Woodland, P.C., Odell, J.J., Valtchev, V., Young, S.J., 1994.
 Large vocabulary continuous speech recognition using HTK.
 In: Proc. Internat. Conf. Acosut. Speech Signal Process.,
 Adelaide, Vol. 2, pp. 125–128.
- Woodland, P.C., Leggetter, C.J., Odell, J. Valtchev, V., Young, S.J., 1995a. The 1994 HTK large vocabulary speech recognition system. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Detroit, Vol. 1, pp. 73–76.
- Woodland, P.C., Leggetter, C.J., Odell, J.J., Valtchev, V., Young, S.J., 1995b. The development of the 1994 HTK large vocabulary speech recognition system. In: Proc. Spoken Language Technology Workshop. Morgan Kaufmann, Austin, TX, pp. 104–109.
- Young, S.J., Odell, J.J., Woodland, P.C., 1994. Tree-based state tying for high accuracy acoustic modelling. In: Proc. Human Language Technology Workshop. Morgan Kaufmann, Plainsboro, NJ, pp. 307–312.