FLEXIBLE SPEAKER ADAPTATION USING MAXIMUM LIKELIHOOD LINEAR REGRESSION

C.J. Leggetter & P.C. Woodland

Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ. UK.

ABSTRACT

The maximum likelihood linear regression (MLLR) approach for speaker adaptation of continuous density mixture Gaussian HMMs is presented and its application to static and incremental adaptation for both supervised and unsupervised modes described. The approach involves computing a transformation for the mixture component means using linear regression. To allow adaptation to be performed with limited amounts of data, a small number of transformations are defined and each one is tied to a number of component mixtures. In previous work, the tyings were predetermined based on the amount of available data. Recently we have used dynamic regression class generation which chooses the appropriate number of classes and transform tying during the adaptation phase. This allows complete unsupervised operation with arbitrary adaptation data. Results are given for static supervised adaptation for non-native speakers and also unsupervised incremental adaptation. Both show the effectiveness and flexibility of the MLLR approach.

1. INTRODUCTION

Over the last few years much progress has been made in speaker independent (SI) recognition system performance. However, even with good speaker independent systems some speakers are modelled poorly, and it is still the case that speaker dependent (SD) systems can give significantly better performance with sufficient speaker-specific training data. In many cases it is undesirable to train an SD system due to the large amount of training data needed and hence the required enrollment time. Therefore speaker adaptation (SA) techniques which tune an existing speech recognition system to a new speaker are of great interest.

Adaptation methods require a sample of speech (adaptation data) from the new speaker so that the models can be updated. The amount of adaptation data needed depends on the way the SA technique uses the data and on the type of system to be adapted. For example MAP estimation [1] requires a relatively large amount of data since it updates only those models for which examples are present in the data. This problem becomes particularly severe when HMM systems that contain a very large numbers of parameters are used.

This paper considers the maximum likelihood linear regression approach (MLLR) [3] which is a parameter transformation technique that has proved successful while using only small amounts of adaptation data. The method is extended to be more flexible and suitable for use in unsupervised adaptation in both static and incremental modes.

In MLLR adaptation an initial set of speaker independent models are adapted to the new speaker by transforming the mean parameters of the models with a set of linear transforms. By using the same transformation across a number of distributions and pooling the transformation training data, maximum use is made of the adaptation data, and the parameters of all state distributions can be adapted. The set of Gaussians that share the same transformation is referred to as a *regression class*. The transformations are trained so as to maximise the likelihood of the adaptation data with the transformed model set.

In previous work [3], the tying of the transformations was determined before adaptation. Here the adaptation procedure is enhanced by calculating the number and membership of the regression classes during the adaptation procedure. Using this dynamic approach allows all modes of adaptation to be performed in a single framework. This approach is evaluated on data from the 1994 ARPA CSR S3 and S4 "spoke" tests. Experiments on S3 demonstrates the effectiveness on static supervised adaptation for non-native speakers, and experiments with S4 show that using the same framework, incremental unsupervised adaptation can be easily implemented.

The structure of the paper is as follows: first the MLLR approach is reviewed, and the extension to incremental adaptation discussed; Sec. 3 describes fixed and dynamic approaches to regression class definition; Sec. 4 compares static supervised and unsupervised adaptation. The experimental evaluation on the 1994 CSR data is given in Sec. 5 and presents adaptation results for the S3 and S4 tests as well as discussing how speaker adaptation was integrated into the 1994 HTK system for the H1-P0 test [6].

2. MLLR OVERVIEW

This section briefly reviews the MLLR approach, and gives equations for the estimation of the MLLR transformations. This information is covered in much greater detail in [2]. Sec. 2.3 then shows how the approach can be extended for incremental adaptation.

2.1. MLLR Basis

Each state in a continuous density mixture Gaussian HMM has an output distribution made up of a number of component densities. A state distribution with m components can be expanded to m parallel single Gaussian states. Therefore in the mathematical description in this section, the case of single Gaussian output distribution states is described, and the extension to mixture Gaussians is straightforward.

Each output distribution is characterised by a mean μ_j and a covariance Σ_j. In the adaptation procedure the SI means are mapped to an estimate of the unknown SD means ($\hat{\mu}_j$) by a linear regression-based transform estimated from the adaptation data

$$\hat{\mu}_j = W_j \nu_j$$

where W_j is the $n \times (n+1)$ transformation matrix and ν_j is the extended mean vector

$$\nu_j = [1, \mu_{j_1}, \ldots, \mu_{j_n}]' \ .$$

The regression transformation is estimated so as to maximise the likelihood of the adptatation data. If a separate regression matrix is trained for each distribution then this becomes equivalent to standard Baum-Welch retraining using the adaptation data.

To allow the approach to be effective with small amounts of adaptation data, each regression matrix is associated with a number of state distributions and estimated from the combined data. By tying in this fashion, the transforms required for each component mean produce a general transformation for all the tied components, and hence parameters not represented in the training data can be updated. This use of tying also means that the transformation matrices can be estimated robustly and hence the method is effective even for unsupervised adaptation.

After transformation, the probability density function of state j generating a speech observation vector $\mathbf{o}$ of dimension n is

$$b_j(\mathbf{o}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{o} - W_j \nu_j)' \Sigma_j^{-1} (\mathbf{o} - W_j \nu_j)}$$

2.2. Estimation of MLLR Matrices

The transformations are computed to maximise the likelihood of the adaptation data. Given a set of T frames of adaptation data $O = \mathbf{o}_1 \ldots \mathbf{o}_T$, the probability of occupying state j at time t while generating O, $\gamma_j(t)$, using the current parameter set, is given by

$$\gamma_j(t) = \frac{f(O, \theta_t = j|\lambda)}{f(O|\lambda)}$$

where $f(O, \theta_t = j|\lambda)$ is the likelihood of occupying state j at time t and generating O, and $f(O|\lambda)$ is the total likelihood of the model generating the observation sequence. The $\gamma_j(t)$ are computed using the forward-backward algorithm.

Assuming that all the Gaussian covariance matrices are diagonal and that W_j is tied between R Gaussians $j_1 \ldots j_R$, then it can be shown that W_j can be computed column by column by

$$\mathbf{w}_i = G_i^{-1} \mathbf{z}_i. \tag{1}$$

In (1) $\mathbf{z}_i$ is the i^{th} column of the matrix Z given by

$$Z \;\; = \;\; \sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{j_r}(t) \Sigma_{j_r}^{-1} \mathbf{o}_t \nu_{j_r}' \tag{2}$$

and G_i is given by

$$G_i = \sum_{r=1}^{R} c_{ii}^{(r)} \nu_{j_r} \nu_{j_r}'$$

where $c_{ii}^{(r)}$ is the i^{th} diagonal element of the r^{th} tied state covariance scaled by the total state occupation probability:

$$C^{(r)} \;\; = \;\; \sum_{t=1}^{T} \gamma_{j_r}(t) \Sigma_{j_r}^{-1} \tag{3}$$

A full derivation of this result is given in [2].

Updating the parameters using the above equations constitutes one iteration of MLLR adaptation. If the change in parameters results in a different posterior probability of state occupation then the likelihood of the adaptation data can be further increased by further MLLR iterations.

It should be noted that the above assumes that the transformations are "full" regression matrices and a simplified form is obtained if the matrices are assumed to be diagonal. However we have previously found that full matrices give superior performance and hence all experiments reported in this paper assume the use of full regression matrices.

2.3. Incremental Adaptation

The basic equations for MLLR assume that all adaptation data is available before the means are updated (static adaptation). By simple manipulation of equations (2) and (3) the time dependent components can be accumulated separately so that the following is obtained

$$Z = \sum_{r=1}^{R} \Sigma_{j_r}^{-1} \left[\sum_{t=1}^{T} \gamma_{j_r}(t) \mathbf{o}_t \right] \nu'_{j_r} \quad (4)$$

$$C^{(r)} = \left[\sum_{t=1}^{T} \gamma_{j_r}(t) \right] \Sigma_{j_r}^{-1}. \quad (5)$$

By accumulating the observation vectors associated with each Gaussian and the associated occupation probabilities, the MLLR equations can be applied at any point in time with the current values of the mean vectors, and hence adaptation may be performed incrementally. For each adaptation update, all the data associated with each state in the regression class is used to generate the transformation matrix. It should be noted that for this incremental form to be equivalent to static adaptation it is assumed that updating does not change the observation vector/state alignment of previously seen utterances.

3. REGRESSION CLASSES

The tying of transformation matrices between mixture components is achieved by defining a set of regression classes. Each regression class has a single transformation matrix associated with it, and all the mixture components within that class are transformed by the same matrix. The matrix is estimated using data allocated to the mixture components within the class.

3.1. Fixed Regression Classes

In previous work on MLLR [3] the class definitions were predetermined by assessing the amount of adaptation data available, and then using a mixture component clustering procedure based on a likelihood measure to generate an appropriate number of classes. Experiments using mixture Gaussian tied state cross word triphones using the ARPA Resource Management (RM) database confirmed that the optimal number of regression classes was roughly proportional to the amount of adaptation data available (see Table 1).

No. Adapt Utts.	Optimal No. Classes
20	4
40	15
100	40
600	200

Table 1: Optimal number of fixed regression classes for adaptation data, results on static adaptation using RM

3.2. Dynamic Regression Classes

The use of predetermined class definitions assumes that the amount of adaptation data available is known in advance, and that a sufficient amount of data will be assigned to each regression class. Classes with insufficient data assigned to them will result in poor estimates of the transformations or the class may be dominated by a specific mixture component. Hence, computing the number of classes and the appropriate tying during the adaptation phase after the data has been observed is desirable.

To facilitate dynamic regression class definition, the mixture components in the system are arranged into a tree. For a small HMM system, the leaves of the tree would represent individual mixture components and at higher levels in the tree the mixture components are merged into groups of similar components based on a distance measure between components. The tree root node represents a single group containing all mixture components. The tree is used so that the most specific set of regression classes is generated for which there are sufficient adaptation data.

When HMM systems with very large numbers of mixture components (the systems described later have 77,000 or more mixture components) it may not feasible to construct a tree with a single mixture component at each leaf node. Instead the leaves are based on an initial clustering into *base classes*. Each base class contains a (reasonably small) set of components which are deemed similar using a distance measure between components.

To accumulate the statistics required for the adaptation process, accumulators are associated with the mixture components. The summed state occupation probability and the observation vectors associated with each component during the forward-backward alignment are recorded. When the adaptation alignment is complete, the total amount of data allocated to each mixture component is known. A search is then made through the tree starting at the root node to find the set of regression class definitions. A separate regression class is created at the lowest level in the tree for which there is sufficient data. This search allows the data to be used in more than one regression class to ensure that the mixture component means are updated using the most specific regression transforms.

4. UNSUPERVISED STATIC ADAPTATION

Implementation of static supervised and static unsupervised adaptation schemes using MLLR are very similar. Supervised adaptation uses a known word sequence for each sentence whereas unsupervised adaptation uses the output of a recogniser to label to data. The labelled data is passed to the forward-backward procedure where the appropriate statistics are gathered and the MLLR transforms generated. The model parameters are then updated. Previously [3] we have reported results using the RM corpus using fixed regression classes and showed that supervised and unsupervised adaptation result in similar performance. This is due in large part to the use of general regression classes which reduce the effects of misalignments and poor labelling of data, giving good performance with unsupervised adaptation.

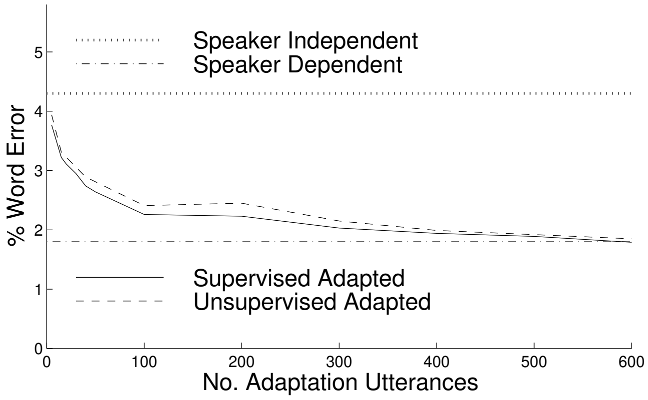

Figure 1: Supervised vs Unsupervised Adaptation using RM

The RM experiments [3] were based on a gender independent cross word triphone system with 1778 tied states and a 6-component mixture distribution per state. This was trained using the standard RM SI-109 training set. A speaker dependent version was also trained for each of the 12 RM SD speakers using the 600 SD training sentences. All testing was on the 100 sentences of SD test data for each speaker using the standard word-pair grammar. Static supervised and unsupervised recognition experiments with varying amounts of the speaker specific training data used for adaptation were performed. Figure 1 shows these results and also the performance of the SI and SD systems for comparison.

5. EVALUATION ON WSJ DATA

This section describes evaluation of the MLLR adaptation approach with both static supervised adaptation for non-native speakers (S3 test) and incremental unsupervised adaptation to improve the performance on native speakers (S4 test). Both types of adaptation used the same baseline speaker independent system, and the same regression class tree. In all cases the dynamic tree-based approach to regression class definition was used. The recognition results were all computed using the final adjudicated reference transcriptions and phone-mediated alignments.

5.1. Baseline SI System

The baseline speaker independent system used for the S3 and S4 experiments was a gender independent cross word triphone mixture Gaussian tied state HMM system (HMM-1 system of [6]), and is similar to the system described in [5]. In the HMM-1 system speech is parameterised using 12 MFCCs, normalised log energy and the first and second differentials of these parameters to give a 39 dimensional acoustic vector. Decision tree-based state clustering [7] was used to define 6399 speech states, and then a 12 component mixture Gaussian distribution trained for each tied state (a total of about 6 million parameters). The acoustic training data consisted of 36493 sentences from the SI-284 WSJ0+1 set, and the 1993 LIMSI WSJ lexicon and phone set was used. The recognition tests for S3 and S4 used a 5k (4986) word vocabulary and the standard MIT Lincoln Labs 5k trigram language model. Decoding used the single pass dynamic network decoder described in [4].

5.2. Regression Class Tree

The regression class tree was built using the divergence between mixture components as the distance measure. 750 base classes were generated using a simple clustering algorithm. Initially, 750 mixture components were chosen, and the nearest 10 components to each one were assigned to the same base class. Every other component was then assigned to the appropriate base class using an average distance from all the existing members. This technique was efficient and assigned a reasonable number (mostly around 100) of mixture components to each base class. A regression tree was then built using a similar distance measure.

The base classes were compared on a pairwise basis using an average divergence between all members of each class. To speed up processing, the search space was pruned by computing the average distributions of each class and only considering the closest 10 in the detailed match. At each node the two closest classes were combined, and any class remaining was given a separate node. After 2 levels of such combination the remainder of the tree was built using the average distributions of each node for comparison. This created a tree with 11 levels and 1502 separate nodes.

5.3. Spoke S3 Results

The aim of the S3 spoke was to investigate the use of static supervised adaptation to improve performance with non-native speakers. Each speaker supplies utterances of the standard set of 40 adaptation sentences which were recorded for all speakers in the corpus.

For use with MLLR these 40 sentences were first used in a Viterbi alignment procedure to select the appropriate pronunciation of each word and any inter-word silences etc. The resulting phone string was then used and a number of iterations of MLLR were performed to obtain an adapted model set for the current speaker. Several iterations of MLLR may be required in the case of non-native speakers since the original models are poor and hence the state/frame alignments may change after adaptation. The word error rate with the SI HMM-1 models and native speaker recogniser settings was 27.14% for the S3 1994 development test data and 20.72% for the S3 1994 evaluation test data.

Table 2 gives the results for systems with recogniser settings tuned for non-native speakers. The effect of multiple iterations of MLLR and adaptation using a single global regression matrix are shown.

Regression Classes	Iterations MLLR	% Word Error	
		S3-dev'94	S3 Nov'94
baseline	baseline	20.82	16.67
tree	1	12.80	11.52
tree	2	12.34	10.99
tree	3	12.20	10.99
global	2	16.10	13.81

Table 2: % word error rates for S3 non-native speakers with MLLR static supervised adaptation.

For native speakers the average error rate with the HMM-1 system is about 5%, and without any adaptation the error rate is a factor of four to five higher for non-natives. It can be seen from Table 2 that with multiple iterations of MLLR and the dynamic tree-based regression class definitions (and revised set-up) the error rate is reduced by an average of 55% from the SI system. The use of multiple regression classes gives on average a 22% reduction in error rate over a single global class and the use of multiple iterations of MLLR gives a worthwhile reduction in error.

5.4. Spoke S4 Results

The aim of the S4 test was to improve the performance of native speakers using unsupervised incremental adaptation. The HMM-1 system was used and incremental MLLR integrated into the dynamic network decoder. Each S4 test-set contained about 100 sentences from each of 4 speakers, and in fact both the 1994 S4 development data and the evaluation data contained speakers with high error rates.

When performing incremental adaptation as described in Sec. 2.3 the parameters can be updated at any time. In the tests performed here there was no update until 3 sentences had been recognised and then the interval between successive updates was varied (every sentence, every 5 sentences, every 10 sentences). Furthermore the use of a global regression class updating every sentence was also investigated.

Regression Classes	Update Interval	% Word Error	
		S4-dev'94	S4 Nov'94
baseline	baseline	9.08	7.76
tree	1	6.66	6.43
tree	5	6.69	6.58
tree	10	6.76	6.62
global	1	7.27	7.04

Table 3: % word error rates for S4 with MLLR unsupervised incremental adaptation.

It can be seen from Table 3 that a worthwhile decrease in error rate is obtained with unsupervised adaptation (average of 22%). Indeed the speaker with the highest initial error rate improved from 21.5% to 14.8%, and all speakers yielded a lower rate with all adapted systems (including the global regression class). The computational overhead of adaptation is approximately inversely proportional to the update interval. If the update interval is increased to 10 sentences there is only a small drop in performance and a large reduction in computation due to adaptation.

The operation of the tree-based dynamic regression class definition is illustrated in Fig. 2, and shows that the number of classes defined is approximately linear in the number of sentences available for accumulation of adaptation statistics. The differing slopes are mainly due to the different speaking rates of different speakers.

5.5. Adaptation In Nov'94 H1 System

The same approach used in S4 for unsupervised speaker adaptation was also used for the November 1994 H1-P0 HTK system [6]. In this test there were only about 15 sentences from each speaker, speaking sentences from unfiltered newspaper articles. The recogniser used for the test had a 65k word vocabulary and a 4-gram lan-

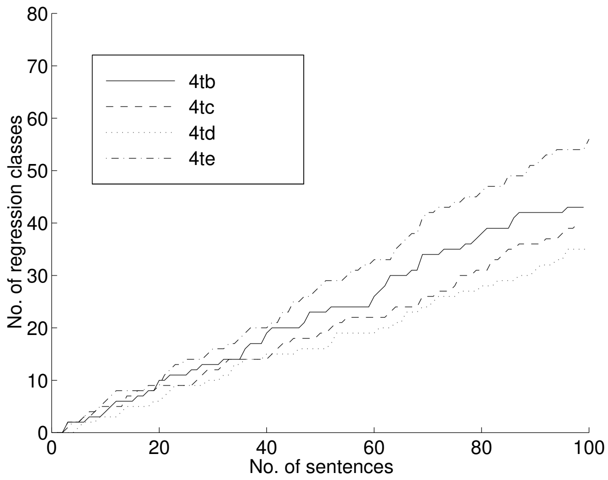

Figure 2: Variation of the number of regression classes used as recognition proceeds for each speaker of the Nov'94 S4 test set

guage model. The acoustic models to be adapted were a gender dependent set built using a decision tree with a wider phonetic context than the HMM-1 set described above. In total there were about 15 million parameters in this HMM set (HMM-2). Further details of this system are given in [6]. A regression class tree with 750 base classes was also built for the HMM-2 set, and the gender of the speaker was identified automatically by the system based on the first 2 sentences. The models for the identified gender were adapted using MLLR and then adapted again after every second sentence. The results from the system on both the 1994 H1 development and evaluation test data, with and without unsupervised incremental speaker adaptation are shown in Table 4.

Adaptation	% Word Error	
	H1-dev'94	H1 Nov'94
N	8.30	7.93
Y	7.28	7.18

Table 4: % word error rates for 1994 HTK H1-P0 evaluation system with and without unsupervised incremental speaker adaptation.

On the development data the error rate reduced by 12% with adaptation and on the evaluation data by 9%. An analysis of the error rate change on a speaker by speaker basis showed that for the development data 17 of the 20 speakers had a reduced error rate with adaptation and the error rate for only one speaker increased. For the evaluation data only 11 speakers improved and 7 performed more poorly. However the speakers that did improve tended to be those that initially performed poorly and in some cases the improvements were quite large. In cases where the performance deteriorated this was usually by only a small amount. The HTK H1-P0 system used for the evaluation was configured with incremental unsupervised adaptation and returned the lowest reported error rate in the test.

6. CONCLUSION

The MLLR approach for adapting a speaker independent model system has been extended to allow incremental adaptation and dynamic allocation of regression classes. The framework is therefore useful in static and incremental adaptation in both unsupervised and supervised modes with minimal changes to the system. The approach has been applied to a number of different problems with success, including unsupervised incremental adaptation of a large state-of-the-art HMM system.

Acknowledgements

C.J. Leggetter was funded by an EPSRC studentship. ARPA provided access to the CAIP computing facility which was used for some of this work. LIMSI kindly provided their 1993 WSJ Lexicon. We would like to thank the other members of the Cambridge HTK group for their help, in particular Julian Odell.

References

1. Gauvain J-L. & Lee C-H. (1994). Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *IEEE Trans. SAP*, Vol. 2, No. 2, 291–298.
2. Leggetter C.J. & Woodland P.C. (1994). Speaker Adaptation Using Linear Regression. *Technical Report CUED/F-INFENG/TR.181.* Cambridge University Engineering Department, June 1994.
3. Leggetter C.J. & Woodland P.C. (1994). Speaker Adaptation of Continuous Density HMMs Using Linear Regression. *Proc. ICSLP'94*, Vol. 2, pp. 451-454, Yokohama.
4. Odell J.J., Valtchev V., Woodland P.C. & Young S.J. (1994). A One Pass Decoder Design For Large Vocabulary Recognition. *Proc. ARPA Human Language Technology Workshop, March 1994*, pp. 405-410, Morgan Kaufmann.
5. Woodland P.C., Odell J.J., Valtchev V. & Young S.J. (1994). Large Vocabulary Continuous Speech Recognition Using HTK. *Proc. ICASSP'94*, Vol. 2, pp. 125-128, Adelaide.
6. Woodland P.C., Leggetter C.J., Odell J.J., Valtchev V. & Young S.J. (1995). The Development of the 1994 HTK Large Vocabulary Speech Recognition System. *Proc. ARPA 1995 Spoken Language Technology Workshop*, Barton Creek.
7. Young S.J., Odell J.J. & Woodland P.C. (1994). Tree-Based State Tying for High Accuracy Acoustic Modelling. *Proc. ARPA Human Language Technology Workshop, March 1994*, pp. 307-312, Morgan Kaufmann.