SPEAKER ADAPTATION OF HMMS
USING LINEAR REGRESSION

C. J. Leggetter & P. C. Woodland

CUED/F-INFENG/TR. 181

June 1994

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

SPEAKER ADAPTATION OF HMMS USING LINEAR REGRESSION

C. J. Leggetter & P. C. Woodland

CUED/F-INFENG/TR. 181

June 1994

Abstract

A method of speaker adaptation for continuous density HMMs is presented. The model parameters of a general speaker independent system are adapted to a new speaker using a transformation of the mean vectors based on linear regression. The method uses the same maximum likelihood optimisation criteria as Baum-Welch training of model parameters, and can be implemented using the forward-backward algorithm. A full derivation of the transformation is given.

To allow adaptation to be performed on small amounts of data a set of regression classes are defined. The data within each class is pooled to calculate a general regression transformation for that class, and the same transformation is applied to a number of model parameters.

Experiments have been performed on the ARPA RM1 database using a triphone HMM system with mixture Gaussian output distributions. Results show that a 42% reduction in error from the speaker independent system can be achieved by using 40 adaptation utterances from the new speaker.

Contents

1 Introduction

Rapid advances in speech recognition research have resulted in high performance speaker independent recognition systems [14]. These systems perform well because they have used large amounts of data to provide detailed modelling of speech patterns.

It is still the case however, that training a system for a specific speaker will result in better recognition performance on that speaker than using a speaker independent system. Typically, a speaker independent system has 2 to 3 times the error rate of a speaker dependent system [7]. A speaker dependent system is tuned precisely to that one speaker, and the characteristics of the speaker can be modelled very well. The drawback of implementing speaker dependent systems is that they require the speaker to provide large amounts (hours) of training data to get sufficient modelling accuracy.

Using model adaptation techniques, a compromise between the comprehensive modelling of all speech phenomena in speaker independent systems and the specific modelling of the speaker dependent system can be achieved. By starting with a good speaker independent system and using a fairly small amount of data from a new speaker, the model set can be adapted to improve the modelling for that speaker.

This report details an approach to speaker adaptation using continuous density HMMs. The method uses a scheme of adapting the parameters of the HMM system using a linear regression approach. The initial speaker independent system is used to generate statistics of how the new speaker characteristics are different from those modelled by the system. These statistics are then used to update the model parameters to give an improved system for the new speaker.

The method is based on that used by Hewett [6] in which least squares regression is applied to adapt templates in dynamic time warping. The theory has been extended to HMMs and uses a maximum likelihood optimisation criteria for regression. The least squares criteria can be shown to be a specific case of maximum likelihood optimisation. The data is used in a flexible manner enabling the method to be applied using only a very small amount of data (adaptation data) from the new speaker to generate a global adaptation transform. If more data is available, better adaptation can be performed by increasing the number of transforms and making each one more specific. This method is referred to as MLLR (maximum likelihood linear regression) adaptation.

The basic premise behind MLLR adaptation is the same as that used in many VQ/HMM adaptation methods [5][13], i.e. that the main difference between speakers is characterised by the position of phones in acoustic space. In VQ/HMM adaptation techniques the codebook is usually updated to fit the new speaker, in the MLLR approach the means of the HMM state distributions are transformed to be more representative of the speaker.

Many previous uses of transforms for adaptation have used least squares error as the optimisation criteria for the transform. Such approaches have shown promising results on both discrete HMMs [3] and continuous density HMMs [2]. These approaches also favour using the transform to transform incoming speech vectors for the adaptation instead of the system parameters.

Many of the approaches which update the system parameters during adaptation are based on maximum *a posteriori* (MAP) estimation [4] [8]. These methods are generally implemented so that only those parameters in models for which data is available are updated. This contrasts with the MLLR method which adapts all models even if no model-specific data is available.

The report is organised as follows. Section 2 gives an outline of the MLLR approach. The actual MLLR transform is derived in section 3 with the special cases of MLLR, including a comparison with least squares regression, in section 4. Implementation issues are discussed in section 5 and the method evaluated using word-internal triphone models on the Resource Management RM1 database in section 6.

2 Outline of Adaptation Approach

The MLLR approach to speaker adaptation assumes that the initial speaker independent system is a continuous density HMM system which has been well trained. The system is adapted to a new speaker by transformation of the system parameters as outlined in the following sections.

2.1 Adaptation of Model Parameters

The aim of the method is to use the adaptation data provided by the speaker to improve the modelling of a previously trained system. It is assumed that the amount of data for adaptation is small, so adapting all the model parameters individually is not possible. For many models there would be no adaptation data, and other model parameters would have insufficient examples of speaker data. In such cases individual parameter adaptation would be poor due to the small sample of observed examples.

This method limits itself to simply updating the means of the mixture components making up the state output distributions. The rationale behind this is that the differences between speakers is mainly characterised in the estimates of the mixture component means. The speaker independent mixture component means are transformed to improve the modelling of the new speaker. As there is no adaptation of the transition probabilities, mixture component weights, or mixture component covariances, these parameters all take their values from the original model set.

2.2 HMM notation

A continuous density HMM is characterised by the following parameters:

N - the number of states in the HMM where states 1 and N are non-emitting states used to connect models together.

A - an $N \times N$ transition matrix where a_{ij} represents the probability of moving from state i to state j.

b_i - the output distribution associated with state i $(1 < i < N)$

The output mixture distribution of a state is made up by combining a number of component densities. Assuming each component density is a Gaussian and there are M component densities in each state, the output distribution can be described using the parameters:

c_{ik} - the mixture weight of mixture component k.

μ_{ik} - the mean of mixture component k (vector of length n).

C_{ik} - the $n \times n$ covariance matrix of mixture component k.

where $1 \leq k \leq M$, and n is the dimension of the speech vector.

Given a speech frame vector $\mathbf{o}$, the probability density function of that vector being generated by mixture component k, $b_{ik}(\mathbf{o})$ is

$$b_{ik}(\mathbf{o}) = \frac{1}{(2\pi)^{\frac{n}{2}}|C_{ik}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{o}-\mu_{ik})'C_{ik}^{-1}(\mathbf{o}-\mu_{ik})} \tag{1}$$

The component densities are combined to give the state probability density:

$$b_i(\mathbf{o}) = \sum_{k=1}^{M} c_{ik} b_{ik}(\mathbf{o}) \tag{2}$$

[17].

2.3 Adaptation of the Mean Vector

The mean vector of a Gaussian distribution is adapted using a linear transform W (an $n \times (n+1)$ matrix) which optimises a maximum likelihood objective function.

If μ is the mean, define $\hat{\mu}$ as

$$\hat{\mu} = \begin{bmatrix} \omega \\ \mu_1 . \\ . \\ . \\ \mu_n \end{bmatrix} \tag{3}$$

where ω is the offset term for the regression ($\omega = 1$ for the standard offset). The estimate of the adapted mean is given by

$$\mu_{new} = W\hat{\mu} \tag{4}$$

and the probability density function for the Gaussian distribution in the adapted system is:

$$b(\mathbf{o}) = \frac{1}{(2\pi)^{\frac{n}{2}}|C|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{o}-W\hat{\mu})'C^{-1}(\mathbf{o}-W\hat{\mu})} \tag{5}$$

If W is the extended identity matrix (column 1 a zero vector, columns $2 \rightarrow n+1$ the identity matrix), then this adapted mean is identical to the mean in the original system. One approach to adaptation is to initialise a matrix W to the extended identity matrix for each distribution in the system, and calculate an estimate of W only when there is sufficient data. However, this is a poor approach for several reasons. First, any unseen mixtures are never updated, secondly if enough data is available to robustly estimate W then there is probably enough data to re-estimate the weights and covariances.

A better approach is to pool the information from a number of distributions into a single matrix W_p, which is then used to transform all the means of the contributing distributions. This is similar in essence to the use of tying within HMMs for estimating model parameters [16]. The degree of tying is determined by the amount of adaptation data available. The case of small amounts of training data is of special interest, and this obviously necessitates a large degree of tying.

3 Derivation of Regression Matrix

Since state distributions using mixture Gaussians can be expanded into a set of parallel single Gaussian states, the regression transform is first derived for the single Gaussian distribution per state, and later extended to the general case of Gaussian mixtures.

Considering the general case of state s with a single Gaussian output density and associated regression matrix W_s. Defining $\hat{\mu}_s$ as the extended mean vector:

$$\hat{\mu}_s = \begin{bmatrix} \omega \\ \mu_{s_1}. \\ . \\ . \\ \mu_{s_n} \end{bmatrix} \tag{6}$$

where ω is the offset term for the regression, the probability density function for the state is:

$$b_s(o) = \frac{1}{(2\pi)^{\frac{n}{2}}|C_s|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{O}-W_s\hat{\mu}_s)'C_s^{-1}(\mathbf{O}-W_s\hat{\mu}_s)} \tag{7}$$

The W_s are chosen to maximise the likelihood of the adapted models generating the adaptation data.

3.1 Definition of Auxiliary Function

Assume the adaptation data is a series of T observations generated by a stochastic process O,

$$O = \mathbf{o}_1 \ldots \mathbf{o}_T \tag{8}$$

Denote the current set of model parameters by λ and a re-estimated set of model parameters as $\bar{\lambda}$. The sequence of states used to generate O is given by

$$\theta = (\theta_0 \theta_1 \ldots \theta_T) \tag{9}$$

where $\theta_0 = 1$. The likelihood of generating the observed speech frames while following the state sequence θ is

$$\mathcal{F}(O, \theta|\lambda) = a_{\theta_T N} \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} b_{\theta_t}(\mathbf{o}_t) \tag{10}$$

If all possible state sequences of length T are denoted by the set Θ, the total likelihood of the model set generating the observation sequence is

$$\mathcal{F}(O|\lambda) = \sum_{\theta\in\Theta} \mathcal{F}(O, \theta|\lambda) \tag{11}$$

This is the objective function to be maximised during adaptation. It is convenient to define an auxiliary function $Q(\lambda, \bar{\lambda})$:

$$Q(\lambda, \bar{\lambda}) = \sum_{\theta\in\Theta} \mathcal{F}(O, \theta|\lambda) \log(\mathcal{F}(O, \theta|\bar{\lambda})) \tag{12}$$

Choosing model parameters to maximise the auxiliary function increases the value of the objective function (unless it is at a maximum). Therefore successively forming a new auxiliary function with improved parameters iteratively maximises the objective function. A proof of this is given by Baum [1] and extended to mixture distributions and vector observations by Liporace [12] and Juang [9].

3.2 Maximisation of Auxiliary Function

Using the re-estimated parameters (indicated with a bar e.g. $\bar{x}$) in the output density function

$$\log(\mathcal{F}(O,\theta|\bar{\lambda})) = \log\left[\bar{a}_{\theta_T N}\prod_{t=1}^{T}\bar{a}_{\theta_{t-1}\theta_t}\bar{b}_{\theta_t}(\mathbf{o}_t)\right] \tag{13}$$

$$= \log\bar{a}_{\theta_T N} + \sum_{t=1}^{T}\log\bar{a}_{\theta_{t-1}\theta_t} + \sum_{t=1}^{T}\log\bar{b}_{\theta_t}(\mathbf{o}_t) \tag{14}$$

The auxiliary function (12) can be split into components based on the different parameters

$$Q(\lambda,\bar{\lambda}) = \sum_{\theta\in\Theta}\mathcal{F}(O,\theta|\lambda)\left[\log\bar{a}_{\theta_T N} + \sum_{t=1}^{T}\log\bar{a}_{\theta_{t-1}\theta_t} + \sum_{t=1}^{T}\log\bar{b}_{\theta_t}(\mathbf{o}_t)\right] \tag{15}$$

$$= \sum_{i=1}^{N}Q_{a_i}[\lambda,\{\bar{a}_{ij}\}_{j=1}^{N}] \quad + \quad \sum_{j=1}^{N}Q_b(\lambda,\bar{b}_j) \tag{16}$$

Where

$$Q_{a_i}[\lambda,\{\bar{a}_{ij}\}_{j=1}^{N}] = \sum_{\theta\in\Theta}\mathcal{F}(O,\theta_T=i|\lambda)\log\bar{a}_{iN} + \sum_{\theta\in\Theta}\sum_{t=1}^{T}\sum_{j=1}^{N}\mathcal{F}(O,\theta_{t-1}=i,\theta_t=j|\lambda)\log\bar{a}_{ij} \tag{17}$$

$$Q_b(\lambda,\bar{b}_j) = \sum_{\theta\in\Theta}\sum_{t=1}^{T}\mathcal{F}(O,\theta_t=j|\lambda)\log\bar{b}_j(\mathbf{o}_t) \tag{18}$$

Since only the transformation W_s is re-estimated, only $Q_b(\lambda,\bar{b}_s)$ needs to be maximised. Define S as the set of all state distributions in the system, and $\gamma_s(t)$ as the total occupation probability of state s at time t given that the observation sequence O is generated.

$$\gamma_s(t) = \frac{1}{\mathcal{F}(O|\lambda)}\sum_{\theta\in\Theta}\mathcal{F}(O,\theta_t=s|\lambda) \tag{19}$$

So equation (18) can be written:

$$Q_b(\lambda,\bar{b}_s) = \mathcal{F}(O|\lambda)\sum_{t=1}^{T}\gamma_s(t)\log\bar{b}_s(\mathbf{o}_t) \tag{20}$$

Expanding $\log\bar{b}_s(\mathbf{o}_t)$:

$$\log\bar{b}_s(\mathbf{o}_t) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}log|C_s| - \frac{1}{2}(\mathbf{o}_t - \overline{W}_s\hat{\mu}_s)'C_s^{-1}(\mathbf{o}_t - \overline{W}_s\hat{\mu}_s) \tag{21}$$

the auxiliary function is

$$Q_b(\lambda,\bar{b}_s) = -\frac{1}{2}\mathcal{F}(O|\lambda)\sum_{t=1}^{T}\gamma_s(t)[n\log(2\pi) + \log|C_s| + (\mathbf{o}_t - \overline{W}_s\hat{\mu}_s)'C_s^{-1}(\mathbf{o}_t - \overline{W}_s\hat{\mu}_s)] \tag{22}$$

To maximise $Q(\lambda,\bar{\lambda})$ with respect to $\overline{W}_s$ it is necessary to differentiate $Q(\lambda,\bar{\lambda})$ with respect to $\overline{W}_s$ and equate to zero. Thus for a maximum

$$\frac{dQ(\lambda,\bar{\lambda})}{d\overline{W}_s} = \frac{d}{d\overline{W}_s}\left[\sum_{k\in S}Q_b(\lambda,\bar{b}_k)\right] = \frac{d}{d\overline{W}_s}Q_b(\lambda,\bar{b}_s) = 0 \tag{23}$$

$$\frac{d}{d\overline{W}_s} Q_b(\lambda, \bar{b}_s) = -\frac{1}{2}\mathcal{F}(O|\lambda)\frac{d}{d\overline{W}_s}\sum_{t=1}^{T}\gamma_s(t)[n\log(2\pi)+\log|C_s|+h(\mathbf{o}_t,s)] \tag{24}$$

Where $$h(\mathbf{o}_t, s) = (\mathbf{o}_t - \overline{W}_s\hat{\mu}_s)'C_s^{-1}(\mathbf{o}_t - \overline{W}_s\hat{\mu}_s)$$

and differential of $h(\mathbf{o}_t, s)$ is

$$\frac{d}{d\overline{W}_s}h(\mathbf{o}_t, s) = -2C_s^{-1}\left[\mathbf{o}_t - \overline{W}_s\hat{\mu}_s\right]\hat{\mu}_s' \tag{25}$$

Therefore equation (23) becomes

$$\frac{d}{d\overline{W}_s} Q_b(\lambda, \bar{b}_s) = \mathcal{F}(O|\lambda)\sum_{t=1}^{T}\gamma_s(t)C_s^{-1}\left[\mathbf{o}_t - \overline{W}_s\hat{\mu}_s\right]\hat{\mu}_s' = 0 \tag{26}$$

and hence

$$\sum_{t=1}^{T}\gamma_s(t)C_s^{-1}\mathbf{o}_t\hat{\mu}_s' = \sum_{t=1}^{T}\gamma_s(t)C_s^{-1}\overline{W}_s\hat{\mu}_s\hat{\mu}_s' \tag{27}$$

Equation (27) gives the general form of the optimisation of $\overline{W}_s$.

When each distribution has a separate regression transform a value for the adapted mean is easily derived. Noting that the adapted mean is $\bar{\mu}_s = W_s\hat{\mu}_s$ and re-arranging equation (27):

$$\bar{\mu}_s = \overline{W}_m\hat{\mu}_s = \frac{\sum_{t=1}^{T}\gamma_s(t)\mathbf{o}_t}{\sum_{t=1}^{T}\gamma_s(t)} \tag{28}$$

which is the standard maximum likelihood re-estimation formula for the mean vector.

3.3 Tied Regression Matrices

The formula given in equation (28) assume state specific regression matrices. To generalise to tied regression matrices (e.g. a global regression matrix) the summation should be performed over all tied distributions. If W_s is shared by R states $\{s_1, s_2 \dots s_R\}$ equation (27) becomes:

$$\sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)C_{s_r}^{-1}\mathbf{o}_t\hat{\mu}_{s_r}' = \sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)C_{s_r}^{-1}\overline{W}_s\hat{\mu}_{s_r}\hat{\mu}_{s_r}' \tag{29}$$

3.4 Re-estimation Formula for Tied Case

To derive a re-estimation formula for the tied case, equation (29) is rewritten as

$$\sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)C_{s_r}^{-1}\mathbf{o}_t\hat{\mu}_{s_r}' = \sum_{r=1}^{R}V^{(r)}\overline{W}_s D^{(r)} \tag{30}$$

where $V^{(r)}$ is the state distribution inverse covariance matrix scaled by the state occupation probability:

$$V^{(r)} = \sum_{t=1}^{T}\gamma_{s_r}(t)C_{s_r}^{-1} \tag{31}$$

and $D^{(r)}$ is the outer product of the extended mean vectors

$$D^{(r)} = \hat{\mu}_{s_r}\hat{\mu}_{s_r}' \tag{32}$$

If the individual matrix elements of $V^{(r)}$,W_s and $D^{(r)}$ are denoted by $v_{ij}^{(r)}$, w_{ij} and $d_{ij}^{(r)}$ respectively, the right hand side of equation (30) is an $n \times (n+1)$ matrix Y with individual elements y_{ij} given by:

$$y_{ij} = \sum_{p=1}^{n} \sum_{q=1}^{n+1} w_{pq} \left[\sum_{r=1}^{R} v_{ip}^{(r)} d_{qj}^{(r)} \right] \tag{33}$$

If all covariances are diagonal, and using the fact that D is symmetric

$$\sum_{r=1}^{R} v_{ip}^{(r)} d_{qj}^{(r)} = \begin{cases} \sum_{r=1}^{R} v_{ii}^{(r)} d_{jq}^{(r)} & \text{when} \quad i = p \\ 0 & \text{when} \quad i \neq p \end{cases} \tag{34}$$

so

$$y_{ij} = \sum_{q=1}^{n+1} w_{iq} g_{jq}^{(i)} \tag{35}$$

where $g_{jk}^{(i)}$ are the elements of the $(n+1) \times (n+1)$ matrix $G^{(i)}$ which is the sum of the outer products of the extended means scaled by the i^{th} diagonal component of the weighted mixture component inverse variance of all shared mixture components ($D^{(r)}$ scaled by $v_{ii}^{(r)}$)

$$g_{jq}^{(i)} = \sum_{r=1}^{R} v_{ii}^{(r)} d_{jq}^{(r)} \tag{36}$$

The left hand side of equation (30) is an $n \times (n+1)$ matrix Z with elements z_{ij}, so $Z = Y$ and

$$z_{ij} = y_{ij} = \sum_{q=1}^{n+1} w_{iq} g_{jq}^{(i)} \tag{37}$$

z_{ij} and $g_{jq}^{(i)}$ are not dependent on $\overline{W}_s$ and can both be computed from the observation vectors and the model parameters. Hence, a system of simultaneous equations is generated:

$$\mathbf{w}_i' = G^{(i)-1} \mathbf{z}_i' \tag{38}$$

where $\mathbf{w}_i$ and $\mathbf{z}_i$ are the i^{th} rows of $\overline{W}_s$ and Z respectively.
These can be solved using Gaussian elimination or LU decomposition methods to calculate $\overline{W}_s$ on a row by row basis.

3.5 Extension to Mixture Distributions

For states with output distributions made up of M mixture components the probability density function for the state is given by

$$b_s(\mathbf{o}) = \sum_{k=1}^{M} c_{sk} b_{sk}(\mathbf{o}) \tag{39}$$

where $b_{sk}(\mathbf{o})$ is the k^{th} Gaussian mixture component of state s and c_{sk} the associated component weighting. Thus the likelihood function becomes

$$\begin{aligned} \mathcal{F}(O, \theta|\lambda) &= a_{\theta_T,N} \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} \left[\sum_{k=1}^{M} c_{\theta_t k} b_{\theta_t k}(\mathbf{o}_t) \right] & (40) \\ &= \sum_{k_1=1}^{M} \sum_{k_2=1}^{M} \cdots \sum_{k_T=1}^{M} \left[a_{\theta_T,N} \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} c_{\theta_t k_t} b_{\theta_t k_t}(\mathbf{o}_t) \right] & (41) \end{aligned}$$

Defining Ω_b as the set of all possible branch sequences (mixture component sequences) of length T, where such a sequence is $K = (k_1, k_2 \ldots, k_T)$, the joint density of the stochastic process is

$$\mathcal{F}(O|\lambda) = \sum_{\theta \in \Theta} \sum_{K \in \Omega_b} \mathcal{F}(O, \theta, K|\lambda) \tag{42}$$

where

$$\mathcal{F}(O, \theta, K|\lambda) = a_{\theta_T N} \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} c_{\theta_t k_t} b_{\theta_t k_t}(\mathbf{o}_t) \tag{43}$$

This can be interpreted by considering each set of state sequences which generates O as being a superposition of M^T branch layers.

The auxiliary function is now redefined to take the branch layers into consideration:

$$\begin{aligned} Q(\lambda, \bar{\lambda}) &= \sum_{\theta \in \Theta} \sum_{K \in \Omega_b} \mathcal{F}(O, \theta, K|\lambda) \log \mathcal{F}(O, \theta, K|\bar{\lambda}) &(44) \\ &= \sum_{\theta \in \Theta} \sum_{K \in \Omega_b} \mathcal{F}(O, \theta, K|\lambda) \left\{ \log \bar{a}_{\theta_T N} + \sum_{t=1}^{T} \log \bar{a}_{\theta_{t-1}\theta_t} + \sum_{t=1}^{T} \log \bar{b}_{\theta_t k_t}(\mathbf{o}_t) + \sum_{t=1}^{T} \log \bar{c}_{\theta_t k_t} \right\} &(45) \end{aligned}$$

Separating into partial auxiliary functions:

$$Q(\lambda, \bar{\lambda}) = \sum_{i=1}^{N} Q_{a_i}[\lambda, \{\bar{a}_{ij}\}_{j=1}^{N}] + \sum_{j=1}^{N} \sum_{k=1}^{M} Q_b(\lambda, \bar{b}_{jk}) + \sum_{j=1}^{N} Q_{c_j}[\lambda, \{\bar{c}_{jk}\}_{k=1}^{M}] \tag{46}$$

where

$$\begin{aligned} Q_{a_i}[\lambda, \{\bar{a}_{ij}\}_{j=1}^{N}] &= \sum_{\theta \in \Theta} \sum_{K \in \Omega_b} \mathcal{F}(O, \theta_T = i, K|\lambda) \log \bar{a}_{iN} + \\ &\quad \sum_{\theta \in \Theta} \sum_{K \in \Omega_b} \sum_{t=1}^{T} \sum_{j=1}^{N} \mathcal{F}(O, \theta_{t-1} = i, \theta_t = j, K|\lambda) \log \bar{a}_{ij} &(47) \\ Q_b(\lambda, \bar{b}_{jk}) &= \sum_{\theta \in \Theta} \sum_{K \in \Omega_b} \sum_{t=1}^{T} \mathcal{F}(O, \theta_t = j, k_t = k|\lambda) \log \bar{b}_{jk}(\mathbf{o}_t) &(48) \\ Q_{c_j}(\lambda, \{\bar{c}_{jk}\}_{k=1}^{M}) &= \sum_{\theta \in \Theta} \sum_{K \in \Omega_b} \sum_{k=1}^{M} \sum_{t=1}^{T} \mathcal{F}(O, \theta_t = j, k_t = k|\lambda) \log \bar{c}_{jk}(\mathbf{o}_t) &(49) \end{aligned}$$

Again, only $Q_b(\lambda, \bar{b}_{jk})$ is dependent on the regression transform, so this is the only function to be considered. Defining $\gamma_{jk}(t)$ as the total occupation probability of mixture component k of state j at time t given that the observation sequence O is generated:

$$\gamma_{jk}(t) = \frac{1}{\mathcal{F}(O|\lambda)} \sum_{\theta \in \Theta} \sum_{K \in \Omega_b} \mathcal{F}(O, \theta_t = j, k_t = k|\lambda) \tag{50}$$

reduces equation (49) to

$$Q_b(\lambda, \bar{b}_{jk}) = \mathcal{F}(O|\lambda) \sum_{t=1}^{T} \gamma_{jk}(t) \log \bar{b}_{jk}(\mathbf{o}_t) \tag{51}$$

which is equivalent to equation (20) for the single Gaussian case.

Thus by substituting mixture component occupation probabilities for state occupation probabilities transformations for individual mixture components can be derived.

3.6 Multiple Observation Sequences

The transform has been derived for the case of a single observation sequence. The extension to the more general case of multiple observation sequences is trivial.

Given a set of Q observation sequences $O^{(1)} \ldots O^{(Q)}$ with observation sequence $O^{(q)}$ having T_q observation frames ($O^{(q)} = \mathbf{o}_1^{(q)} \ldots \mathbf{o}_{T_q}^{(q)}$), and considering a single component mixture per state, equation (27) becomes:

$$\sum_{q=1}^{Q} \sum_{t=1}^{T_q} \gamma_s^{(q)}(t) C_s^{-1} \mathbf{o}_t^{(q)} \hat{\mu}_s' = \sum_{q=1}^{Q} \sum_{t=1}^{T_q} \gamma_s^{(q)}(t) C_s^{-1} \overline{W}_s \hat{\mu}_s \hat{\mu}_s' \tag{52}$$

where

$$\gamma_s^{(q)}(t) = \frac{1}{\mathcal{F}(O^{(q)}|\lambda)} \sum_{\theta \in \Theta_{T_q}} \mathcal{F}(O^{(q)}, \theta_t = s|\lambda) \tag{53}$$

and Θ_{T_q} is the set of all possible state sequences of length T_q.

The tied regression matrix formula (equation (30)) then becomes

$$\sum_{q=1}^{Q} \sum_{t=1}^{T_q} \sum_{r=1}^{R} \gamma_{s_r}(t) C_{s_r}^{-1} \mathbf{o}_t^{(q)} \hat{\mu}_{s_r}' = \sum_{r=1}^{R} V^{(r)} \overline{W}_s D^{(r)} \tag{54}$$

where $D^{(r)}$ is the extended mean outer product matrix as before and $V^{(r)}$ is the state distribution inverse covariance matrix scaled by the state occupation probability:

$$V^{(r)} = \sum_{q=1}^{Q} \sum_{t=1}^{T} \gamma_{s_r}^{(q)}(t) C_{s_r}^{-1} \tag{55}$$

Hence $G^{(i)}$ and Z can be calculated and the transform estimated as before.

4 Special Cases of MLLR

This section considers special cases of the regression matrix. The simplification of the maximum likelihood regression transform to least squares regression is presented first. This is followed by possible approaches to reducing the number of parameters required to estimate the regression transform.

4.1 Least Squares Regression

Least squares regression optimisation, as used by Hewett [6], can be shown to be a special case of the maximum likelihood regression. If all the covariance matrices of the distributions assigned to the same class are the same ($C_{s_1} = C_{s_2} = \ldots = C_{s_R}$) equation (29) becomes:

$$\sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)\mathbf{o}_t\hat{\mu}'_{s_r} = \sum_{t=1}^{T}\sum_{r=1}^{R}\gamma_{s_r}(t)\overline{W}_s\hat{\mu}_{s_r}\hat{\mu}'_{s_r} \tag{56}$$

If each speech frame is assigned to exactly one distribution (e.g. by Viterbi alignment) so that

$$\gamma_{s_r}(t) = \begin{cases} 1 & \text{if } \mathbf{o}_t \text{ is assigned to state distribution } s_r \\ 0 & \text{otherwise} \end{cases} \tag{57}$$

then equation (56) becomes

$$\sum_{t=1}^{T}\mathbf{o}_t\hat{\mu}'_{\theta_t}\delta_{s\theta_t} = \overline{W}_s\sum_{t=1}^{T}\hat{\mu}_{\theta_t}\hat{\mu}'_{\theta_t}\delta_{s\theta_t} \tag{58}$$

where

$$\delta_{s\theta_t} = \begin{cases} 1 & \text{if } \theta_t \in \{s_1 \ldots s_R\} \\ 0 & \text{otherwise} \end{cases} \tag{59}$$

Defining the matrices X and Y as

$$\begin{aligned} X &= \left[\begin{array}{cccc} \hat{\mu}_{\theta_1} & \hat{\mu}_{\theta_2} & \ldots & \hat{\mu}_{\theta_T} \end{array}\right] \\ Y &= \left[\begin{array}{cccc} \mathbf{o}_1\delta_{\theta_1 m} & \mathbf{o}_2\delta_{\theta_2 m} & \ldots & \mathbf{o}_T\delta_{\theta_T m} \end{array}\right] \end{aligned} \tag{60}$$

equation (58) becomes

$$YX' = \overline{W}_s XX' \tag{61}$$

so the estimate of the regression matrix is

$$\overline{W}_s = YX'(XX')^{-1} \tag{62}$$

which is the least squares estimate.

4.2 Reducing the Number of Regression Parameters

The $n \times (n+1)$ regression matrix effectively changes the mean in two ways. The first column provides an offset element, and the remaining $n \times n$ portion provides a scaling based on the current mean values.

$$\begin{aligned} \mu_{new} &= W\hat{\mu} \\ \mu_{new_i} &= \omega w_{1i} + \sum_{j=1}^{n} w_{j+1,i}\mu_i \end{aligned} \tag{63}$$

There are two special cases of the regression matrix which can reduce the number of parameters to estimate, both have a bearing on computation and accuracy of the adaptation.

1. A full matrix without an offset term ($\omega = 0$)
2. A regression matrix with the scaling portion diagonal

These cases and their effects on the adaptation theory are briefly described below.

4.2.1 Full Matrix without Offset

The offset column of the regression matrix provides an intersection point between the old and the new mean values, allowing a simple shift from one to the other. By setting the offset term (ω) in the extended mean to zero, the effect of the offset can be ignored.

$$\hat{\mu} = \begin{bmatrix} 0 \\ \mu_1 \\ \vdots \\ \mu_n \end{bmatrix} \tag{64}$$

The same equations apply to the matrix estimation except that W is now effectively $n \times n$ and hence 1 column calculation is saved. If there is sufficient inter-dependency between the mean components, the effect of ignoring the offset column may be small.

4.2.2 Using a diagonal scaling matrix

If the scaling portion of the regression matrix is assumed to be diagonal, the computation can be vastly reduced. This makes the assumption that all the features are independent, and that the shift in each mean component can be calculated by a simple single variable linear regression

$$\mu_{new_i} = x + y\mu_i \tag{65}$$

Still considering the transform as a matrix, the diagonal matrix is rewritten as a vector $\hat{\mathbf{w}}_s$

$$\overline{W}_s = \begin{pmatrix} w_{1,1} & w_{1,2} & 0 & \dots & 0 \\ w_{2,1} & 0 & w_{2,3} & \dots & 0 \\ \vdots & & & & \vdots \\ w_{n,1} & \dots & \dots & 0 & w_{n,n} \end{pmatrix} \quad \hat{\mathbf{w}}_s = \begin{pmatrix} w_{1,1} \\ \vdots \\ w_{n,1} \\ w_{1,2} \\ \vdots \\ w_{n,n+1} \end{pmatrix} \tag{66}$$

and considering the quadratic form with the diagonal regression matrix:

$$\begin{aligned} h(\mathbf{o}_t, s) &= (\mathbf{o}_t - \overline{W}_s\hat{\mu}_s)' C_s^{-1} (\mathbf{o}_t - \overline{W}_s\hat{\mu}_s) \\ &= (\mathbf{o}_t - D_s\hat{\mathbf{w}}_s)' C_s^{-1} (\mathbf{o}_t - D_s\hat{\mathbf{w}}_s) \end{aligned} \tag{67}$$

where D_s is a $n \times 2n$ matrix made up of elements of the extended mean vector ($\hat{\mu}$)

$$D_s = \begin{pmatrix} \omega & 0 & \dots & \dots & 0 & \mu_1 & 0 & \dots & & 0 \\ 0 & \omega & 0 & \dots & \dots & 0 & \mu_2 & 0 & \dots & 0 \\ \vdots & & & & & & & & & \vdots \\ 0 & \dots & 0 & \omega & 0 & \dots & \dots & 0 & \mu_{n-1} & 0 \\ 0 & \dots & \dots & 0 & \omega & 0 & \dots & \dots & 0 & \mu_n \end{pmatrix} \tag{68}$$

Differentiating $h(\mathbf{o}_t, s)$ with respect to $\hat{\mathbf{w}}_s$

$$\frac{d}{d\hat{\mathbf{w}}_s} h(\mathbf{o}_t, s) = -2D_s' C_s^{-1} (\mathbf{o}_t - D_s\hat{\mathbf{w}}_s) \tag{69}$$

Substituting in equation (27)

$$\frac{d}{d\hat{\mathbf{w}}_s} Q_b(\lambda, b'_s) = \sum_{t=1}^{T} \gamma_s(t) D'_s C_s^{-1} (\mathbf{o}_t - D_s \hat{\mathbf{w}}_s) \tag{70}$$

This leads to an equation for the values of the regression matrix elements:

$$\hat{\mathbf{w}}_s = \left[\sum_{t=1}^{T} \gamma_s(t) D'_s C_s^{-1} D_s \right]^{-1} \left[\sum_{t=1}^{T} \gamma_s(t) D'_s C_s^{-1} \mathbf{o}_t \right] \tag{71}$$

With an offset term only one matrix inversion is required to calculate the regression matrix entries, and by ignoring the offset term ($\omega = 0$) the offset column can be ignored, and all matrices can be reduced to diagonal matrices making inversion trivial.

The extension to the tied regression matrix case is simply to extend the summations over the tied mixture components:

$$\hat{\mathbf{w}}_s = \left[\sum_{r=1}^{R} \sum_{t=1}^{T} \sum_{t=1}^{T} \gamma_s(t) D'_{s_r} C_{s_r}^{-1} D_{s_r} \right]^{-1} \left[\sum_{r=1}^{R} \sum_{t=1}^{T} \sum_{t=1}^{T} \gamma_{s_r}(t) D'_{s_r} C_{s_r}^{-1} \mathbf{o}_t \right] \tag{72}$$

The extension to mixture distributions and multiple observation sequences is similar to that presented in section 3.

5 Implementation Issues

The approach is straightforward to implement as it is very similar to standard re-estimation algorithms (e.g. Baum-Welch embedded re-estimation), and the tying of the regression matrices to state mixture components is very similar to that used in HTK [15].

5.1 Estimation of the Transforms

The transform can be calculated using a two step procedure, the first being the accumulation of statistics, and the second step using the statistics to calculate the transform:

For each speech frame in the adaptation data

For each mixture component in the adaptation data

{
Determine probability of frame belonging to mixture component
Determine which W matrix is associated with mixture component
Record probability/frame/mixture component in accumulator for W
}

For each regression matrix

Use accumulator contents to calculate W

The accumulation of data is requires a state/mixture component alignment of the adaptation data, which can be obtained from standard alignment strategies such as Viterbi or Baum-Welch forward backward algorithms. The above algorithm can be used iteratively to find a final value for the regression transforms.

5.2 Calculating Mixture Component Alignments

Assume that the observed speech vectors of the adaptation data are $\mathbf{o}_1 \ldots \mathbf{o}_T$, and that the data is labelled with a model transcription. This allows a forward-backward algorithm to be used to generate the mixture component alignment probabilities. $\alpha_i(t)$ represents the forward likelihood, the likelihood of occupying state i at time t having generated observations $\mathbf{o}_1 \ldots \mathbf{o}_t$, and $\beta_i(t)$ the backward likelihood, the likelihood of generating observations $\mathbf{o}_{t+1} \ldots \mathbf{o}_T$ given that state i is occupied at time t. The initial conditions for α at time $t = 0$ are:

$$\alpha_i(0) = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i \neq 1 \end{cases} \tag{73}$$

The α values are then calculated for each time frame $t \geq 1$

$$\alpha_1(t) = 0 \tag{74}$$

$$\alpha_j(t) = \left[\sum_{i=1}^{N} \alpha_i(t-1)a_{ij}\right] b_j(\mathbf{o}_t) \quad \text{if } j > 1 \tag{75}$$

For the backward pass the initial conditions for β at time $t = T$ are:

$$\beta_i(T) = a_{iN} \qquad \text{for } i = 1 \ldots (N-1) \tag{76}$$

The β values are calculated for each time frame $t < T$

$$\beta_i(t) = \sum_{j=1}^{N-1} a_{ij}\beta_j(t+1)b_j(\mathbf{o}_{t+1}) \qquad 1 \leq i < N \tag{77}$$

The total likelihood of the models generating the observation sequence ($\mathcal{F}(O|\lambda)$) can be found from either α or β:

$$\mathcal{F}(O|\lambda) = p(\mathbf{o}_1 \ldots \mathbf{o}_T|\lambda) = \alpha_N(T) = \beta_1(0) \tag{78}$$

Thus once α and β are calculated, the probability of occupying state i at time t ($p_i(t)$) can be calculated:

$$p_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\mathcal{F}(O|\lambda)} \tag{79}$$

and the probability of occupation of mixture component j of state i at time t is

$$\gamma_{ij}(t) = \frac{\left[\sum_{k=1}^{N} \alpha_k(t-1)a_{ki}\right] c_{ij} b_{ij}(\mathbf{o}_t)\beta_i(t)}{\mathcal{F}(O|\lambda)} \tag{80}$$

When an observation sequence contains data for several models (e.g. subword models) the appropriate models can be concatenated into a single model and the occupation probability calculation performed in the same manner.

Thus each frame can be assigned to each mixture component with the calculated probability, and these are the mixture component occupation probabilities which are used in the MLLR transform calculation. In actual implementation all calculations should be performed in the logarithmic domain to prevent computational underflow.

5.3 Tying Regression Matrices

The tied MLLR transform approach is ideal for situations where there are only small amounts of adaptation data available. A small number of regression matrices can be defined and used to estimate general mean transformations for all of the mixture components. Figure 1 shows the

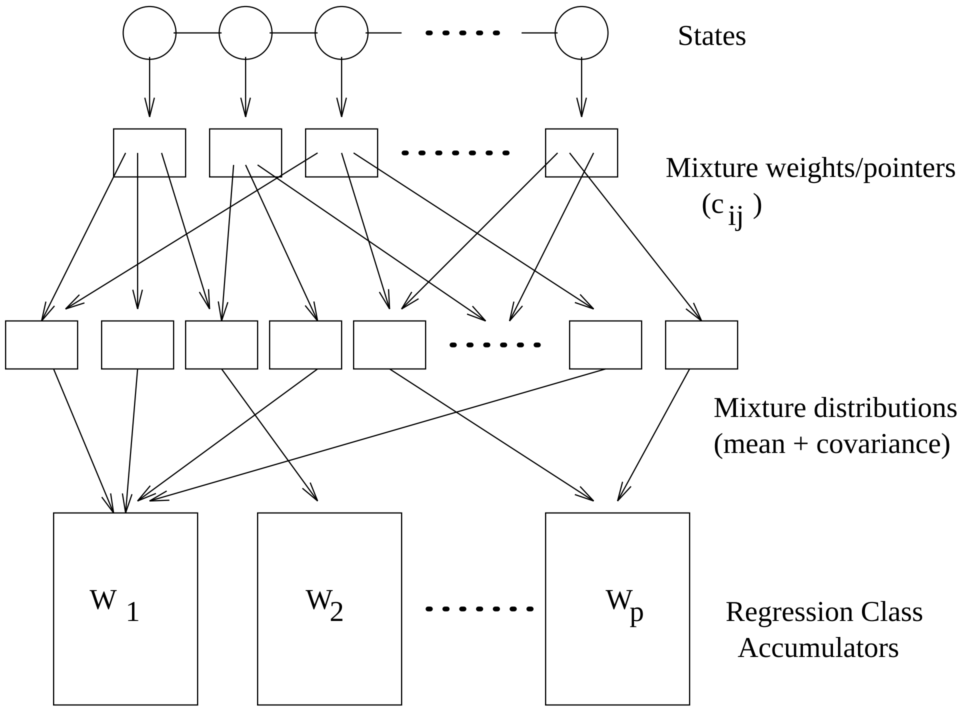

Figure 1: Schematic diagram of tying the regression matrices

schematic diagram of tying regression matrices among several mixture components. Each mixture component is associated with one regression matrix, and any observation vectors assigned to the mixture component are assigned to that regression matrix. Once the regression matrices have been estimated, each mixture component mean is adapted by applying the associated regression matrix.

5.4 Implementing MLLR Adaptation

After determining the mixture component occupancy probabilities the speech frames can be assigned to the appropriate regression classes. The frames are accumulated in a straightforward manner to compute the matrix Z (the left hand side of equation (30)). The accumulation of the individual mixture component occupancies allows the $G^{(i)}$ matrices (equation (36)) to be computed.

Although the method can be applied to dynamic adaptation (performing adaptation as new data is presented), the application to static adaptation, where all the adaptation data is seen before adaptation takes place, is more straightforward. A set of labelled speech data is aligned with the speaker independent model set and the required statistics accumulated. The regression matrices are then computed and all mixture components are updated. A standard recogniser can then be used to perform the recognition task, without regard to the fact that the system has been adapted.

5.5 Computation

The amount of computation involved in computing the regression transformations depends on the degree of tying of the regression matrices. Assuming the matrix Z and the individual mixture component occupation counts have been computed from the adaptation data statistics, the computation needed for adaptation can be assessed. For adapting M mixture components using R regression matrices (with an observation vector of length n) the computation required is as follows:

A single multiplication is required to compute each $v_{ii}^{(r)}$, and 1 multiplication for each $d_{jq}^{(r)}$. Thus each term in the summation for $g_{jk}^{(i)}$ requires 3 multiplications. Each mixture component to be adapted will be incorporated into a $G^{(i)}$ matrix at some point and there are $n \times (n+1)$ $g_{jk}^{(i)}$ elements, making the computation in calculating all the $G^{(i)}$ matrices $3n(n+1)M$ multiplications.

There is a separate $G^{(i)}$ matrix for each column of each regression matrix, making a total of nR $G^{(i)}$ matrices. Each of these is inverted, and then used in a matrix-vector multiplication (n^2 multiplications) to compute the final transform. The component means are then updated at a cost of n^2 multiplications each (total n^2M multiplications).

Thus the overall computational requirement for adaptation is $4n^2M + nM + n^2$ multiplications and nR matrix inversions.

The matrix inversions are computationally expensive, but by using a high degree of tying, the number of inversions needed can be kept small. The $G^{(i)}$ matrices may be ill conditioned due to the lack of data or bias towards an individual mixture component mean, so the matrix inversion needs to be able to invert ill-conditioned matrices. Thus it is suggested that a method such as singular value decomposition is used.

6 Evaluation of MLLR adaptation

6.1 Experimental Setup

The ARPA Resource Management RM1 database was used to evaluate MLLR speaker adaptation. The speech was coded into 25ms frames, with a frame advance of 10ms. Each frame was represented by a 39 component vector consisting of 12 MFCCs plus energy, and their first and second time derivatives.

A set of speaker independent models was trained on the speaker independent portion of the database (3990 utterances), using standard Baum-Welch maximum likelihood estimation. The set consisted of 2371 state clustered word-internal triphone models, containing a total of 1651 states (similar to the model sets in [15]). Each state had 2 mixture components for the output distribution. The 48 phone CMU set was used for labels, and an optional single state silence model was used for between word silences. A single pronunciation for each word was derived from the dictionary used by Lee [11]. This is the initial speaker independent (SI) model set which will be used for adaptation.

The speaker dependent portion of the database was used for all adaptation experiments. This consists of training and testing data from 12 different speakers (7 male and 5 female). A set of 40 files from the test set of each speaker was used for evaluation, and all adaptation data was drawn from the training set.

Adaptation was implemented by using a forward-backward alignment of the adaptation data to assign frames to regression matrices as previously described. All recognition tests were performed using a standard Viterbi recogniser with the standard RM word-pair grammar (perplexity 60).

Definition of Regression Classes

Each regression matrix has to be associated with a set of mixture components. This is achieved by defining a set of regression equivalence classes, which contain a list of the components associated with each regression matrix. Initially all mixture components are assigned to a single global class, and as more classes are required the global class is split according to broad phonetic definitions. Each triphone was mapped to the central phone to decide the broad phonetic category, and then assigned to a class. For example, the split for two classes used one class for vowels and one class for all other phones. The 47 classes division is the case where each central phone is assigned to its own separate class. This method of splitting ensured that all mixtures in all states of the same model were assigned to the same regression class. None of the components in the silence models were adapted.

6.2 Experiments

Experiments have been performed to investigate the merits of the approach. The results reported here look at how the adaptation performance is affected by:

1. The number of regression classes
2. The amount of adaptation data used
3. The use of diagonal or full regression matrices

All adaptation is performed in a static supervised manner with the labelled adaptation data. Only one iteration of adaptation is performed in all cases.

6.2.1 Number of Regression Classes

A series of experiments were performed using 40 adaptation files and different numbers of regression classes.

The results in Table 1 show that a substantial decrease in error rate is achieved on most speakers using a global regression matrix, with an average error reduction of 42%. Increasing the

Speaker	SI	Adapted - Number of Classes						
		1	2	4	8	10	12	47
bef0_3	8.5	7.9	7.6	7.6	6.7	6.0	7.3	10.6
cmr0_2	12.2	3.3	4.8	3.3	3.3	2.9	3.3	7.5
das1_2	5.6	1.7	2.0	1.7	2.0	1.6	2.8	5.6
dms0_4	8.1	4.2	5.1	4.8	2.7	2.7	3.0	7.8
dtb0_3	9.0	7.0	6.7	6.7	7.0	7.0	7.0	7.6
dtd0_5	9.7	7.6	7.9	7.9	7.0	7.8	8.2	10.0
ers0_7	8.7	8.4	7.5	6.3	6.9	7.8	9.0	8.7
hxs0_6	9.8	5.7	5.7	6.0	5.1	3.8	3.9	6.9
jws0_4	5.3	5.0	5.0	4.7	5.0	4.7	5.3	5.9
pgh0_1	4.7	4.7	4.7	4.4	3.0	3.0	4.1	5.0
rkm0_5	17.0	12.3	12.3	11.2	10.3	9.5	9.2	15.9
tab0_7	3.0	2.5	2.2	2.5	2.0	1.9	1.7	5.0
Average	8.4	5.8	5.9	5.6	5.1	4.9	5.4	8.0

Table 1: Supervised static adaptation on 40 files (% Word error on speaker test set)

number of regression classes gradually improves the adaptation performance, until a maximum is reached. Further divisions of classes then gradually degrade the performance. The optimal division in this case is 10 regression classes on average, although the optimal points of individual speakers varies somewhat. The results show that adapting the models to the speaker improves recognition performance over the SI system for all speakers.

6.2.2 Amount of Adaptation Data Needed

To assess how the amount of adaptation data present affects the performance of the adaptation, the number of utterances used in the adaptation process was varied. The number of regression classes was also varied to see how the optimal class splits changed with different data sets.

Number of Adaptation Utterances	Adapted - Number of Classes						
	2	4	6	8	10	12	47
10	6.4	6.1	8.0	8.4	10.9	16.9	82.9
20	6.0	5.8	6.3	6.1	7.6	8.2	54.8
40	5.9	5.6	5.5	5.1	4.9	5.4	8.0
60	5.9	5.6	5.4	4.9	5.0	4.9	7.1
80	5.9	5.4	5.4	5.1	4.9	4.8	5.8
100	5.8	5.5	5.3	5.0	4.7	4.5	4.5
200	5.8	5.5	5.1	4.8	4.6	4.4	3.9
600	5.8	5.5	5.3	4.9	4.6	4.5	3.7

Table 2: Effect of amount of adaptation data on supervised static adaptation
(% Word error on test set, averaged over all speakers)

The results in Table 2 show that as the amount of data increases the number of regression classes should be increased. If the number of regression classes is kept constant and more data is used for adaptation the regression classes appear to reach a point of saturation, where adding more adaptation data gives no further improvement. Using 2 regression classes, the saturation point is about 20 utterances, and adding more data does not significantly improve (or degrade) recognition performance.

The method performs poorly when there are too many classes and not enough data. In these cases the assignment of data to each class is insufficient, and the accumulated matrices to invert are very close to being singular (due to linear dependence). This results in a poor estimate of the regression matrix due to computational error in the matrix inversion and the system performs poorly.

6.2.3 Use of a Diagonal Regression Matrix

The use of a diagonal regression matrix assumes that all speech vector components change independently for a new speaker.

No. of Classes	Diag	Full
1	8.0	6.3
2	7.6	6.4
4	7.6	6.1
6	7.3	8.0
8	7.1	8.4
10	6.9	10.9
20	6.4	–
30	6.3	–
40	6.4	–
47	6.7	–

Table 3: Using diagonal and full regression matrices on 10 adaptation files
(% Word error on test set, averaged over all speakers)

The results of a comparison between using diagonal and full regression matrices (Table 3) show that using a full regression matrix is much more effective than using a diagonal matrix with the same class definitions, as long as there is sufficient data to estimate the full matrix.. Although an examination of full regression matrices showed the main diagonal to be quite dominant, it is clear that the off-diagonal terms relating the interdependencies between components is important. The amount of data needed to estimate a diagonal regression matrix is much smaller than that of a full matrix (diagonal matrix has $2n$ entries and a full regression has $n(n+1)$ entries), so more classes can be used with the same amount of data. The results show that increasing the number of classes does improve performance, but to get adaptation performance approaching that of the full regression matrix an equivalent number of regression terms need to be computed. This requires a similar amount of computation, and with small amounts of data only a small number of transforms can be robustly estimated, suggesting that using a small number of full matrices may be better than using many diagonal matrices.

7 Conclusions

The maximum likelihood linear regression (MLLR) technique of speaker adaptation for continuous density HMMs has been described. The method starts with a set of speaker independent models and modifies these by applying a set of linear transformations to the Gaussian mean vectors. These transformations are each shared between a fairly large number of Gaussians which allows all the mean vectors in the system to be adapted with only a small amount of adaptation data. The parameters of the transformation matrices are chosen so that the likelihood of the speaker specific data is maximised. The re-estimation formulae for this maximisation have been derived, and it is shown how the necessary computations can be integrated into the forward-backward algorithm.

The efficacy of the method has been evaluated using a mixture Gaussian tied-state triphone system with the ARPA Resource Management corpus. A number of experiments showed that both full and diagonal regression matrices could lead to improvements in the system, but that full matrices are rather more effective. Furthermore, it was shown that it is necessary to match the number of regression transformations in the system to the quantity of available adaptation data. It was found that by using an appropriate number of transformations the error rate of a recognition system using 2 component mixture HMMs could be reduced by 42% using 40 adaptation utterances in a static supervised adaptation mode. Future work will investigate unsupervised adaptation and dynamic adaptation using the MLLR technique.

References

[1] L.E. Baum. An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. *Inequalities*, Vol. 3, pp. 1–8, 1972.

[2] H.C. Choi and R.W. King. Speaker Adaptation through Spectral Transformation for HMM based Speech Recognition. *International Symposium on Speech Image Processing and Neural Networks*, Vol. 2, pp. 686–689, Hong Kong, April 1994.

[3] F. Class, A. Kaltenmeier, et al. Fast Speaker Adaptation for Speech Recognition Systems. *Proc. ICASSP*, Vol. 1, pp. 133–136, 1990.

[4] J-L. Gauvain and C-H. Lee. Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *IEEE Trans. Speech and Audio Processing*, Vol. 2, No. 2, pp. 291–298, 1994.

[5] Y. Hao and D. Fang. Speech Recognition Using Speaker Adaptation by System Parameter Transformation. *IEEE Trans. Speech and Audio Processing*, Vol. 2, No. 1, Part 1, pp. 63–68, January 1994.

[6] A.J. Hewett. Training and Speaker Adaptation in Template-based Speech Recognition. PhD thesis, Cambridge University, 1989.

[7] X.D. Huang and K.F. Lee. On Speaker-Independent, Speaker-Dependent and Speaker-Adaptive Speech Recognition. *Proc. ICASSP*, Vol. 2, pp. 877–880, Toronto, 1991.

[8] Q. Huo, C. Chan, and C-H. Lee. Segmental Quasi-Bayesian Learning of the Mixture Coefficients in SCHMM for Speech Recognition. *International Symposium on Speech Image Processing and Neural Networks*, Vol. 2, pp. 678–681, Hong Kong, April 1994.

[9] B-H. Juang. Maximum Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains. *A.T. & T. Technical Journal*, Vol. 64, No. 6, pp. 1235–1249, July-August 1985.

[10] C-H. Lee, C-H. Lin, and B-H. Juang. A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models. *IEEE Trans. Sig. Proc.*, Vol. 39, No. 4, pp. 806–814, April 1991.

[11] K-F. Lee. *Automatic Speech Recognition: The Development of the SPHINX System.* Kluwer Academic, 1989.

[12] L.A. Liporace. Maximum Likelihood Estimation for Multivariate Observations of Markov Sources. *IEEE Trans. Information Theory*, Vol. IT-28, No. 5, pp. 729–734, September 1982.

[13] S. Nakamura and K. Shikano. Speaker Adaptation Applied to HMM and Neural Networks. *Proc. ICASSP*, Vol. 1, pp. 89–92, 1989.

[14] P.C. Woodland, J.J. Odell, V. Valtchev, and S.J. Young. Large Vocabulary Continuous Speech Recognition Using HTK. *Proc. ICASSP*, Vol. 2, pp. 125–128, 1994.

[15] P.C. Woodland and S.J. Young. The HTK Tied-State Continuous Speech Recogniser. *Proc. EuroSpeech*, Vol. 3, pp. 2207–2210, Berlin, 1993.

[16] S.J. Young and P.C. Woodland. The Use of State Tying in Continuous Speech Recognition. *Proc. EuroSpeech*, Vol. 3, pp. 2203–2206, Berlin, 1993.

[17] S.J. Young, P.C. Woodland, and W.J. Byrne. *HTK - Hidden Markov Model Toolkit, Version 1.5.* Cambridge University Engineering Department and Entropic Research Laboratories Inc.