Mean and Variance Adaptation within the MLLR Framework

M.J.F. Gales & P.C. Woodland

April 1996

Revised August 23rd 1996

Cambridge University Engineering Department

Trumpington Street

Cambridge CB2 1PZ

England

Email: {mjfg,pcw}@eng.cam.ac.uk

Abstract

One of the key issues for adaptation algorithms is to modify a large number of parameters with only a small amount of adaptation data. Speaker adaptation techniques try to obtain near speaker dependent (SD) performance with only small amounts of speaker specific data, and are often based on initial speaker independent (SI) recognition systems. Some of these speaker adaptation techniques may also be applied to the task of adaptation to a new acoustic environment. In this case a SI recognition system trained in, typically, a clean acoustic environment is adapted to operate in a new, noise-corrupted, acoustic environment. This paper examines the Maximum Likelihood Linear Regression (MLLR) adaptation technique. MLLR estimates linear transformations for groups of models parameters to maximise the likelihood of the adaptation data. Previously, MLLR has been applied to the mean parameters in mixture Gaussian HMM systems. In this paper MLLR is extended to also update the Gaussian variances and re-estimation formulae are derived for these variance transforms. MLLR with variance compensation is evaluated on several large vocabulary recognition tasks. The use of mean and variance MLLR adaptation was found to give an additional 2% to 7% decrease in word error rate over mean-only MLLR adaptation.

1 Introduction

Current state-of-the-art speaker independent (SI) speech recognition systems are capable of achieving impressive performance in clean acoustic environments for speakers that are well represented in the training data. However for some speakers, performance can be relatively poor e.g. for non-native speakers using a system trained on speech from natives. Furthermore, the performance degrades, often dramatically, if there is some mismatch between the training and test data acoustic environments. For complex speech recognition systems a large amount of data is required to retrain the system for a particular speaker or for a new acoustic environment. Hence, it is very desirable to be able to improve the performance of an existing system while only using a small amount of speaker-specific or environment-specific adaptation data.

One of the key issues to be faced in adaptation is how to adapt a large number of parameters with only a small amount of data. Some environmental adaptation techniques require no speech data in the new acoustic environment, only noise samples, to adapt the model parameters (Gales, 1996; Varga and Moore, 1990). However these schemes make assumptions about the form of the acoustic environment. Techniques that only update distributions for which observations occur in the adaptation data, such as those using maximum a-posteriori (MAP) estimation (Gauvain and Lee, 1994; Lee et al., 1990), require a relatively large amount of adaptation data to be effective. An alternative approach is to estimate a set of transformations that can be applied to the model parameters. If these transformations can capture general relationships between the original model set and the current speaker or new acoustic environment, they can be effective in adapting all the HMM distributions. One such transformation approach is maximum likelihood linear regression (MLLR) (Leggetter and Woodland, 1994; Leggetter and Woodland, 1995b;

Leggetter and Woodland, 1995a) which estimates a set of linear transformations for the mean parameters of a mixture Gaussian HMM system to maximise the likelihood of the adaptation data. It should be noted that while MLLR was initially developed for speaker adaptation, since it reduces the mismatch between a set of models and adaptation data, it can also be used to perform environmental compensation by reducing a mismatch due to channel or additive noise effects[1].

Adaptation techniques may operate in a number of *modes*. If the true transcription of the adaptation data is known then it is termed supervised adaptation, whereas if the adaptation data is unlabelled the adaptation is unsupervised. Situations in which all the adaptation data is available in one block (e.g. from a system enrolment session) and the the system adapted once before use is termed static adaptation. Alternatively the data may become available as the system is used and the system adapted incrementally. The MLLR techniques described in this paper are applicable to all these adaptation modes.

The original MLLR scheme only updated the Gaussian mean parameters. However to more closely model the data in either speaker adaptation or acoustic environment compensation the Gaussian variances should also be modified. Speaker independent models capture both inter- and intra-speaker variability. When such a system has been adapted to a particular speaker only the intra-speaker variabilty should be modelled, and hence the model variances should, in general, be reduced towards those typical of a speaker dependent system. Furthermore, the data variance alters in different acoustic environments. For example the variance of clean speech cepstra tend to be greater than data which contains additive noise (Gales and Young, 1995a).

This paper extends the basic MLLR approach to be able to compensate the variances

[1]This is only strictly true for a stationary noise environment.

of the models in addition to the means. The estimation of variance transformation is again performed in a maximum likelihood (ML) fashion. The technique allows both full and diagonal covariance matrices to be compensated with little additional memory or computational load. The transforms used to adapt the variances may also be either full or diagonal.

The paper starts by examining several transformation approaches for adaptation based on maximising the likelihood of the adaptation data. It then describes the standard MLLR adaptation of the means and the extension of the technique to adapting the variances. MLLR adaptation is then evaluated on a series of test sets from the 1994 ARPA CSR evaluation.

2 Linear Transformation Techniques for Adaptation

A number of different types of linear transformation have been proposed for adaptation of model parameters. These transformations are estimated to reduce the mismatch between the adaptation data and the models using either a least squares criterion (e.g. (Jaschul, 1982; Hewett, 1989))[2] or a ML criterion, as used in MLLR. Furthermore there are a number of possibilities for choosing the form of the transformation and the HMM parameters to which it applies. As in any HMM training problem, it is essential to ensure that the transformation parameters are robustly estimated given the available adaptation data. One approach to ensure robust estimation is to vary the number of transformations depending on the available data so that if insufficient data is available, more Gaussians will share the same transformation.

[2]The least squares criterion is a constrained case of the maximum likelihood criterion (Leggetter and Woodland, 1995b).

A number of different transformation types, based on the maximum likelihood criterion, have been examined in the literature. The simplest approach is to apply a bias to the means (Kenny et al., 1990), or to use diagonal transformation matrices with a bias (e.g. (Digalakis et al., 1995)). This approach can also be used to transform the variances (Digalakis et al., 1995) provided that the means and the variances are transformed using the same diagonal transform. The new model mean, $\hat{\mu}$, and new variance $\hat{\mathbf{\Sigma}}$ ($\mathbf{\Sigma}$ is a diagonal matrix with elements σ^2_{ii}) are given by

$$\hat{\mu}_i = a_{ii}\mu_i + b_i \tag{1}$$

$$\hat{\sigma}^2_{ii} = a^2_{ii}\sigma^2_{ii} \tag{2}$$

where μ is the SI mean, $\mathbf{\Sigma}$ the SI variance, $\mathbf{A}$ is the diagonal transformation matrix with elements a_{ii} and $\mathbf{b}$ is the bias vector.

MLLR (Leggetter and Woodland, 1995b) removes the restriction of a diagonal transformation for the means. Using a similar notation to the above

$$\hat{\mu} = \mathbf{A}\mu + \mathbf{b} \tag{3}$$

$$\hat{\mathbf{\Sigma}} = \mathbf{\Sigma} \tag{4}$$

where $\mathbf{A}$ is now a full transformation matrix. This method has been found to outperform the use of diagonal transformations (Leggetter and Woodland, 1995b; Neumeyer et al., 1995). However the variances are not modified by this method.

To reduce the number of transformation matrix parameters required to be learnt and hence reduce the adaptation data required per transform, a block diagonal matrix may be

used in place of the full transformation matrix (Leggetter, 1995; Neumeyer et al., 1995).

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_s & \mathbf{0}^n & \mathbf{0}^n \\ \mathbf{0}^n & \mathbf{A}_\Delta & \mathbf{0}^n \\ \mathbf{0}^n & \mathbf{0}^n & \mathbf{A}_{\Delta^2} \end{pmatrix} \tag{5}$$

An example of a block diagonal transform is shown in equation 5. Here, the transforms for the static, delta and delta-delta parameters are $\mathbf{A}_s$, $\mathbf{A}_\Delta$ and $\mathbf{A}_{\Delta^2}$ respectively.

An alternative technique which modifies the means and variances is the Stochastic Additive Transform (SAT) (Rose et al., 1994), in which

$$\hat{\mu}_i = \mu_i + \mu_{b_i} \tag{6}$$

$$\hat{\sigma}^2_{ii} = \sigma^2_{ii} + \sigma^2_{b_{ii}} \tag{7}$$

where the additive bias $\mathbf{b}$ has a mean, μ_b, and variance, $\mathbf{\Sigma}_b$, associated with it. Although this transform allows the variances to be modified, the resultant "variances" may not be positive for unobserved distributions, as the bias variance will only be based on the distributions for which there are observations. This is not a major problem as a variance floor may be applied. A slightly modified version of this transformation was also examined (Neumeyer et al., 1995) where

$$\hat{\mu}_i = \mu_i + \mu_{b_i} \tag{8}$$

$$\hat{\sigma}^2_{ii} = a_{ii}\sigma^2_{ii} \tag{9}$$

This modifies the variances in a well-motivated fashion, guaranteeing that the resultant variances are positive. However, the transform was derived for the case where there is only an additive bias for the mean and a diagonal scaling of a diagonal covariance matrix.

In this work the type of transformations given in equations 8 and 9 are derived for more general mean and variance transformations. The transformation of both the mean

and variance may be full, block or diagonal.

3 Maximum Likelihood Linear Regression

The aim of MLLR is to obtain a set of transformation matrices for the model parameters that maximises the likelihood of the adaptation data. MLLR has been applied to a range of speaker adaptation tasks (Leggetter, 1995), in both supervised, unsupervised, static and incremental modes. This section gives the basic theory for MLLR adaptation of the means (Leggetter and Woodland, 1995b) and shows how the linear regression transformation matrices and biases are trained. In addition, the memory requirements to store the statistics used to determine the transformations are discussed. The derivations and notation used in this section follow those in (Leggetter, 1995).

3.1 Estimation of the Mean Transformation

A new estimate of the mean, $\hat{\mu}_m$, is found by

$$\hat{\mu}_m = \hat{\mathbf{W}}_m \xi_m \tag{10}$$

where $\hat{\mathbf{W}}_m$ is the $n \times (n+1)$ transformation matrix (n is the dimensionality of the data) and ξ_m is the extended mean vector

$$\xi_m = \begin{bmatrix} 1 & \mu_1 & \dots & \mu_n \end{bmatrix}^T \tag{11}$$

It is simple to see that

$$\hat{\mathbf{W}}_m = \begin{bmatrix} \hat{\mathbf{b}}_m & \hat{\mathbf{A}}_m \end{bmatrix} \tag{12}$$

The aim is to find the transformation $\hat{\mathbf{W}}_m$ that maximises the likelihood of the adaptation data.

In order to solve this maximisation problem an *Expectation-Maximisation* (EM) technique (Dempster et al., 1977) is used. The standard auxiliary function $\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})$ is adopted,

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \tag{13}$$
$$K_1 - \frac{1}{2}\mathcal{L}(\mathbf{O}_T|\mathcal{M}) \sum_{m=1}^{M}\sum_{\tau=1}^{T} L_m(\tau)\left[K_m + \log(|\boldsymbol{\Sigma}_m|) + (\mathbf{o}(\tau) - \hat{\mu}_m)^T\boldsymbol{\Sigma}_m^{-1}(\mathbf{o}(\tau) - \hat{\mu}_m)\right]$$

where K_1 is a constant dependent only on the transition probabilities, K_m is the normalisation constant associated with Gaussian m, $\mathbf{O}_T = \{\mathbf{o}(1), \ldots, \mathbf{o}(T)\}$ is the adaptation data and

$$L_m(\tau) = p(q_m(\tau)|\mathcal{M}, \mathbf{O}_T) \tag{14}$$

where $q_m(\tau)$ indicates Gaussian m at time τ. Increasing the value of this auxiliary function is guaranteed to increase the likelihood of the adaptation data. To enable robust transformations to be trained the transformation matrices are tied across a number of Gaussians (a transformation per Gaussian is equivalent to conventional re-training of the means). For this work the Gaussians were grouped using a regression class tree (Leggetter and Woodland, 1995a). The tree contains all the Gaussians in the system and statistics are gathered at the leaves (which may each contain a number of Gaussians and define the *base classes*). The set of Gaussians that share a transform are referred to as a *regression class*. The most specific transform that can be robustly estimated is then generated for all the Gaussians in the system. The techniques described here are also applicable to other methods of assigning Gaussians to regression classes.

Given that a particular transformation $\hat{\mathbf{W}}_m$ is to be tied across R Gaussians, $\{m_1, \ldots, m_R\}$, $\hat{\mathbf{W}}_m$ may be found by solving

$$\sum_{\tau=1}^{T}\sum_{r=1}^{R} L_{m_r}(\tau)\boldsymbol{\Sigma}_{m_r}^{-1}\mathbf{o}(\tau)\xi_{m_r}^T = \sum_{\tau=1}^{T}\sum_{r=1}^{R} L_{m_r}(\tau)\boldsymbol{\Sigma}_{m_r}^{-1}\hat{\mathbf{W}}_m\xi_{m_r}\xi_{m_r}^T \tag{15}$$

For the full covariance matrix case the solution is computationally very expensive[3], however, for the diagonal covariance matrix case a closed-form solution is computationally feasible (Leggetter and Woodland, 1995b).

The left-hand side of equation 15 is independent of the transformation matrix and will be referred to as $\mathbf{Z}$, where

$$\mathbf{Z} = \sum_{r=1}^{R} \sum_{\tau=1}^{T} L_{m_r}(\tau) \boldsymbol{\Sigma}_{m_r}^{-1} \mathbf{o}(\tau) \xi_{m_r}^{T} \tag{16}$$

A new variable $\mathbf{G}^{(i)}$ is defined with elements

$$g_{jq}^{(i)} = \sum_{r=1}^{R} v_{ii}^{(r)} d_{jq}^{(r)} \tag{17}$$

where

$$\mathbf{V}^{(r)} = \sum_{\tau=1}^{T} L_{m_r}(\tau) \boldsymbol{\Sigma}_{m_r}^{-1} \tag{18}$$

and

$$\mathbf{D}^{(r)} = \xi_{m_r} \xi_{m_r}^{T} \tag{19}$$

$\hat{\mathbf{W}}_m$ is calculated using

$$\hat{\mathbf{w}}_i^{T} = \mathbf{G}^{(i)-1} \mathbf{z}_i^{T} \tag{20}$$

where $\hat{\mathbf{w}}_i$ is the i^{th} vector of $\hat{\mathbf{W}}_m$ and $\mathbf{z}_i$ is the i^{th} vector of $\mathbf{Z}$.

[3]The closed-form solution requires solving $n \times (n+1)$ simultaneous equations of the form

$$z_{kl} = \sum_{p=1}^{n} \sum_{q=1}^{n+1} \hat{w}_{pq} \left(\sum_{r=1}^{R} v_{kp}^{(r)} d_{ql}^{(r)} \right)$$

for $k = 1 \ldots n, l = 1 \ldots (n+1)$, where $\mathbf{Z}$, $\mathbf{V}^{(r)}$ and $\mathbf{D}^{(r)}$ are defined in equations 16, 18 and 19 respectively.

3.2 Statistics Required for the Mean Transformation

It is interesting to examine the statistics that must be gathered in order to compute the mean transformation matrices. These statistics may be stored at either the Gaussian level or at the regression class level. The most memory efficient technique is dependent on the ratio of the regression classes to the number of Gaussians.

1. **Gaussian Level**. This requires $\sum_{\tau=1}^{T} L_m(\tau)\mathbf{o}(\tau)$ and $\sum_{\tau=1}^{T} L_m(\tau)$ to be stored at a cost of $(n+1)$ floats[4] per Gaussian. It is then possible to generate the right-hand-side of equation 20 directly.

2. **Regression Class Level**. The statistics, $\mathbf{G}^{(i)}$ and $\mathbf{Z}$ may be stored at the regression-class level. This has a memory requirement of $\mathcal{O}(n^3)$ for each regression class (Gales and Woodland, 1996). This assumes that the regression classes have been pre-defined. When regression classes are defined dynamically (Leggetter and Woodland, 1995a), $\mathbf{G}^{(i)}$ and $\mathbf{Z}$ for the chosen regression class may be obtained from child classes. In this case, $\mathbf{G}^{(i)}$ and $\mathbf{Z}$ must be stored at the base class level.

3.3 Multiple Iterations of MLLR

As MLLR is dependent on the frame/state component alignment, $L_m(\tau)$ in equation 13, performance can sometimes be improved using multiple iterations of MLLR (Leggetter and Woodland, 1995a). Additional implementation issues arise when multiple iterations are used, particularly if the use of the regression class tree changes. For a given regression class it is unimportant how many times the means have been transformed, the final transformation will always yield the same value irrespective of whether the original or the latest model set is transformed, provided the frame/state component alignment is the

[4] A float is used as the unit of storage.

same (Gales and Woodland, 1996). However if the regression class tree is used dynamically, the situation may occur where due to changes in the alignments there is insufficient data to generate a transformation for a specific class. It is now not possible to compensate for the transform previously applied to that class. This problem may be overcome by always adapting the original model parameters[5].

4 MLLR Adaptation of the Variances

This section describes the basis of a transformation of the Gaussian variances in the MLLR framework. The means and variances are adapted in two separate stages. Initially new means are found and then, given these new means, the variances are updated.

4.1 Estimation of the Variance Transformation

The HMMs are modified in two steps such that

$$\mathcal{L}(\mathbf{O}_T|\check{\mathcal{M}}) \geq \mathcal{L}(\mathbf{O}_T|\hat{\mathcal{M}}) \geq \mathcal{L}(\mathbf{O}_T|\mathcal{M}) \tag{21}$$

where the models $\hat{\mathcal{M}}$ have just the means updated to $\hat{\mu}_1, \ldots, \hat{\mu}_M$ and the models $\check{\mathcal{M}}$ have both the means and the variances $\hat{\mathbf{\Sigma}}_1, \ldots, \hat{\mathbf{\Sigma}}_M$ updated.

The Gaussian covariance matrices are updated by

$$\hat{\mathbf{\Sigma}}_m = \mathbf{B}_m^T \hat{\mathbf{H}}_m \mathbf{B}_m \tag{22}$$

where $\hat{\mathbf{H}}_m$ is the linear transformation to be estimated and $\mathbf{B}_m$ is the inverse of the Choleski factor of $\mathbf{\Sigma}_m^{-1}$, so

$$\mathbf{\Sigma}_m^{-1} = \mathbf{C}_m \mathbf{C}_m^T \tag{23}$$

[5] Although the original parameters are transformed, the frame/state alignments are found using the latest model set, which requires both model sets to be kept in memory.

and

$$\mathbf{B}_m = \mathbf{C}_m^{-1} \tag{24}$$

The standard auxiliary function is again employed

$$\mathcal{Q}(\mathcal{M}, \check{\mathcal{M}}) = \tag{25}$$
$$K_1 - \frac{1}{2}\mathcal{L}(\mathbf{O}_T|\mathcal{M}) \sum_{m=1}^{M}\sum_{\tau=1}^{T} L_m(\tau)\left[K_m + \log(|\hat{\mathbf{\Sigma}}_m|) + (\mathbf{o}(\tau) - \hat{\mu}_m)^T\hat{\mathbf{\Sigma}}_m^{-1}(\mathbf{o}(\tau) - \hat{\mu}_m)\right]$$

It is hard to directly optimise this expression for both the mean transformation matrix and the variance transform. However it is sufficient to ensure that

$$\mathcal{Q}(\mathcal{M}, \check{\mathcal{M}}) \geq \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) \tag{26}$$

to satisfy equation 21.

Rewriting equation 25 using equations 23 and 22 leads to

$$\mathcal{Q}(\mathcal{M}, \check{\mathcal{M}}) = K_1 - \frac{1}{2}\mathcal{L}(\mathbf{O}_T|\mathcal{M}) \sum_{m=1}^{M}\sum_{\tau=1}^{T} L_m(\tau)\left[K_m + \log(|\mathbf{\Sigma}_m|) + \log(|\hat{\mathbf{H}}_m|)\right. \tag{27}$$
$$\left. + (\mathbf{C}_m^T\mathbf{o}(\tau) - \mathbf{C}_m^T\hat{\mu}_m)^T\hat{\mathbf{H}}_m^{-1}(\mathbf{C}_m^T\mathbf{o}(\tau) - \mathbf{C}_m^T\hat{\mu}_m)\right]$$

The maximisation of equation 27 has a simple closed form solution and leads to a re-estimation formula for $\hat{\mathbf{H}}_m$ analogous to the standard ML estimate of the covariance matrix

$$\begin{aligned}\hat{\mathbf{H}}_m &= \frac{\sum_{\tau=1}^{T} L_m(\tau)(\mathbf{C}_m^T\mathbf{o}(\tau) - \mathbf{C}_m^T\hat{\mu}_m)(\mathbf{C}_m^T\mathbf{o}(\tau) - \mathbf{C}_m^T\hat{\mu}_m)^T}{\sum_{\tau=1}^{T} L_m(\tau)} \\ &= \frac{\mathbf{C}_m^T\left[\sum_{\tau=1}^{T} L_m(\tau)(\mathbf{o}(\tau) - \hat{\mu}_m)(\mathbf{o}(\tau) - \hat{\mu}_m)^T\right]\mathbf{C}_m}{\sum_{\tau=1}^{T} L_m(\tau)}\end{aligned} \tag{28}$$

Up to this point the tying of the variance transformation matrices has been ignored. However, if the transform is to be shared over a number of Gaussians, $\{m_1, \ldots, m_R\}$, then

the re-estimation formula becomes

$$\hat{\mathbf{H}}_m = \frac{\sum_{r=1}^{R} \left\{ \mathbf{C}_{m_r}^T \left[\sum_{\tau=1}^{T} L_{m_r}(\tau)(\mathbf{o}(\tau) - \hat{\mu}_{m_r})(\mathbf{o}(\tau) - \hat{\mu}_{m_r})^T \right] \mathbf{C}_{m_r} \right\}}{\sum_{r=1}^{R} \sum_{\tau=1}^{T} L_{m_r}(\tau)} \tag{29}$$

It is preferable to obtain all the transformation statistics for both the mean and variance transforms in a single pass. Since $\hat{\mu}_m$ is not known when the statistics are being accumulated, it is necessary to rearrange equation 29 as

$$\hat{\mathbf{H}}_m = \frac{\sum_{r=1}^{R} \left\{ \mathbf{C}_{m_r}^T \left[\sum_{\tau=1}^{T} L_{m_r}(\tau)\mathbf{o}(\tau)\mathbf{o}(\tau)^T - \hat{\mu}_{m_r}\overline{\mathbf{o}}_{m_r}^T - \overline{\mathbf{o}}_{m_r}\hat{\mu}_{m_r}^T + \hat{\mu}_{m_r}\hat{\mu}_{m_r}^T \sum_{\tau=1}^{T} L_{m_r}(\tau) \right] \mathbf{C}_{m_r} \right\}}{\sum_{m=1}^{R} \sum_{\tau=1}^{T} L_{m_r}(\tau)} \tag{30}$$

where

$$\overline{\mathbf{o}}_{m_r} = \sum_{\tau=1}^{T} L_{m_r}(\tau)\mathbf{o}(\tau) \tag{31}$$

The estimate of $\hat{\mathbf{H}}_m$ given in equation 30 results in a full covariance matrix, yielding a full covariance matrix for the new estimate of the covariance, $\hat{\mathbf{\Sigma}}_{m_r}$, even if the original covariance matrices were diagonal. This would dramatically increase the memory requirements for the model set if the original covariance matrices were diagonal. However, due to the derivation of these matrices, it is not necessary to store a full symmetric covariance matrix for each individual Gaussian. The original diagonal covariances may be left unchanged and the likelihood calculated as

$$\begin{aligned} \mathcal{L}(\mathbf{o}(\tau)|\mathcal{M}_{m_r}) = K_{m_r} - \frac{1}{2} \Big(& \log(|\mathbf{\Sigma}_{m_r}|) + \log(|\hat{\mathbf{H}}_m|) \\ & + (\mathbf{C}_{m_r}^T \mathbf{o}(\tau) - \mathbf{C}_{m_r}^T \hat{\mu}_{m_r})^T \hat{\mathbf{H}}_m^{-1} (\mathbf{C}_{m_r}^T \mathbf{o}(\tau) - \mathbf{C}_{m_r}^T \hat{\mu}_{m_r}) \Big) \end{aligned} \tag{32}$$

Alternatively $\hat{\mathbf{H}}_m$ may be forced to be a diagonal transformation by setting the off-diagonal elements to zero which results in $\hat{\mathbf{\Sigma}}_m$ being a diagonal covariance matrix. This is still guaranteed to increase the likelihood of the adaptation data. If these diagonal

transformations are used then the number of transformation parameters is small, n, compared to the number used with a full transform for the means, $(n+1) \times n$. This diagonal transformation is used for all the results presented in this paper.

4.2 Statistics Required to Estimate Variance Transformations

The statistics for calculating the variance transformation may, again, be stored at either the Gaussian level or at the regression class level.

1. **Gaussian Level**. In addition to the statistics required to estimate the mean transformation it is necessary to store $\sum_{\tau=1}^{T} L_m(\tau)\mathbf{o}(\tau)\mathbf{o}(\tau)^T$ to calculate equation 30. If a full covariance transformation, $\hat{\mathbf{H}}_m$, is to be calculated, then a full symmetric matrix must be stored for each Gaussian. For many systems this is impractical, as it has a memory requirement of $\mathcal{O}(n^2)$ per Gaussian.

2. **Regression Class Level** Alternatively the statistics may be stored at the regression class level. There are two options available. The statistics to estimate both the mean and variance transformation matrices may be accumulated within a single pass, or two passes, of the adaptation data.

   (a) **Single-Pass**. To accumulate the statistics to estimate the transformation matrices for both means and variances in a single pass is complex as the new estimate of the mean, $\hat{\mu}_{m_r}$, is not known when the transformation statistics are required to be collected. Hence, it is necessary to modify the elements in equation 30 so that they are independent of the mean transformation, $\hat{\mathbf{W}}_m$. This has a memory requirement of $\mathcal{O}(n^3)$ per regression class with the additional constraint that the original covariance matrices are diagonal (Gales and Woodland, 1996).

(b) **Two-Pass**. Statistics to calculate the transformation matrix for the means are obtained in the first pass. In the second pass, the mean transformation is known, so it is only necessary to store

$$\mathbf{Z}^{(R_c)} = \sum_{r=1}^{R_c} \mathbf{C}_{m_r}^T \left[\sum_{\tau=1}^{T} L_{m_r}(\tau)(\mathbf{o}(\tau) - \hat{\mu}_{m_r})(\mathbf{o}(\tau) - \hat{\mu}_{m_r})^T \right] \mathbf{C}_{m_r} \tag{33}$$

and the regression class occupancy to calculate equation 29. This has a memory requirement of $\mathcal{O}(n^2)$ per regression class, but is more computationally expensive as a second pass through the adaptation data is required.

3. **Regression Class and Gaussian Level**. A combination of the above two storage strategies may also be used. The mean transformation statistics are stored at the Gaussian level, a cost of $\mathcal{O}(n)$ per Gaussian. At the regression class level the value of

$$\mathbf{Z}^{(R_c)} = \sum_{r=1}^{R_c} \mathbf{C}_{m_r}^T \left[\sum_{\tau=1}^{T} L_{m_r}(\tau)\mathbf{o}(\tau)\mathbf{o}(\tau)^T \right] \mathbf{C}_{m_r} \tag{34}$$

which has a cost of $\mathcal{O}(n^2)$ per regression class.

The choice of which statistics are accumulated is dependent on the number of Gaussians compared to the number of regression classes and the allowable computational load for the adaptation.

One of the drawbacks of the variance transformation described is that it does not simultaneously optimise the mean and the variance transformations. The ML estimate of the mean transformation matrix, equation 15, is a function of the current estimate of the covariance matrix. Thus as the variances change so the ML estimate of $\hat{\mathbf{W}}_m$ will alter. It is therefore possible to use an iterative scheme to alternately optimise the mean transformation and then the covariance transformation provided $\hat{\mathbf{H}}_m$ is constrained to be

diagonal. However, in practice, it has been found that no significant gains were obtained using this iterative scheme. Therefore all the experiments described in this paper a non-iterative scheme for estimating the mean and variances transforms was used.

5 Experiments and Results

To evaluate the variance adaptation scheme the ARPA 1994 CSRNAB development and evaluation data was used (Pallett et al., 1995). A variety of tasks were examined, with data recorded in both clean[6] and noise corrupted environments. In order to compare results with those quoted for the 1994 ARPA evaluation it was necessary to use incremental unsupervised adaptation for all tasks. The tasks examined are listed below.

1. **Spoke 4: Incremental Speaker Adaptation**. This is a 5k word recognition task recorded in a clean environment with around one hundred sentences from each of 4 speakers. The aim of the task was to perform unsupervised incremental adaptation with a relatively large amount of adaptation data per speaker.

2. **Hub 1: Unlimited Vocabulary NAB News Baseline**. This is an unlimited vocabulary task with approximately 15 sentences from each of 20 speakers. The data was recorded in a clean environment.

3. **Spoke 5: Microphone Independence**. This task uses data recorded with unknown microphones. It is a 5k word recognition task with about 10 sentences from each of 20 speakers.

[6]Here the term "clean" refers to the training and test conditions being from the same microphone type with a high signal-to-noise ratio.

4. **Spoke 10: Noisy Channel**. The S10 spoke is a 5k word task with car noise artificially added onto clean speech. Three noise levels are given, however only the noisiest *Level 3* condition was evaluated in this paper. There were approximately 10 sentences from each of 10 speakers.

5.1 System Description

The baseline system used for the recognition task was a gender-independent cross-word-triphone mixture-Gaussian tied-state HMM system. This was the same as the "HMM-1" model set used in the HTK 1994 ARPA evaluation system (Woodland et al., 1995). The speech was parameterised into 12 MFCCs, C_1 to C_{12}, along with normalised log-energy and the first and second differentials of these parameters. This yielded a 39-dimensional feature vector. The acoustic training data consisted of 36493 sentences from the SI-284 WSJ0 and WSJ1 sets, and the LIMSI 1993 WSJ lexicon and phone set were used. The standard HTK system was trained using decision-tree-based state clustering (Young et al., 1994) to define 6399 speech states. A 12 component mixture Gaussian distribution was then trained for each tied state, a total of about 6 million parameters. For all the spoke tasks, S4, S5 and S10, the standard MIT Lincoln Labs 5k word trigram language model was used. For the H1 task a 65k word list and dictionary was used with the trigram language model described in (Woodland et al., 1995). All decoding used a dynamic-network decoder (Odell et al., 1994) which can either operate in a single-pass or rescore pre-computed word lattices.

For all the noise corrupted tasks, S5 and S10, the model set parameters were initially modified using parallel model combination (PMC) (Gales and Young, 1995a). PMC was used to modify the models so that they were approximately matched to the new acoustic

environment. MLLR was then applied to fine-tune the models (Woodland et al., 1996). For computational efficiency the PMC Log-Add approximation (Gales and Young, 1995a) with simple convolutional noise estimation (Gales and Young, 1995b) was used to modify the means of the models. In order to apply PMC to the models it was necessary to replace normalised log-energy by C_0 and linear regression differentials by simple differences in the front end analysis. Furthermore, cepstral mean normalisation (CMN) was not used for the PMC models and hence it was necessary to first compensate for the different global signal levels of the WSJ0 and WSJ1 databases by applying an offset to the C_0 feature vector coefficient of the WSJ0 data such that it had the same average value as the WSJ1 database. To generate the PMC models, the standard HTK model set was initially estimated as described above. The model set was then updated using single-pass retraining (Gales, 1996) to be based on the PMC parameter set. An additional pass of standard Baum-Welch re-estimation was then performed.

So that meaningful comparison can be made with other published results, all the experiments described here were implemented using unsupervised incremental adaptation. Furthermore, none of the systems were optimised for the test data. Appropriate values of the grammar scale factors and insertion penalties were determined from the standard clean speaker independent system, or in the case of the PMC model sets, by optimising them on the ARPA 1994 CSRNAB S0 development data (Gales, 1996).

For all tasks full transformation matrices for the means and diagonal transforms for the variances were used. The transformation sequence for the variances was the same as that for the means and the minimum class occupancy counts were also set to be the same for both the mean and variance transformation matrices. The regression class trees used throughout this work were based on clean speech and did not transform the silence models.

The regression class tree was defined by clustering components in acoustic space (Leggetter and Woodland, 1995a). The model parameters were updated after every two sentences of adaptation data.

5.2 Results

The first task examined was the Spoke 4 incremental speaker adaptation task. The effectiveness of the variance adaptation was examined by analysing the change in auxiliary function as adaptation proceeds and then recognition results on test data are given.

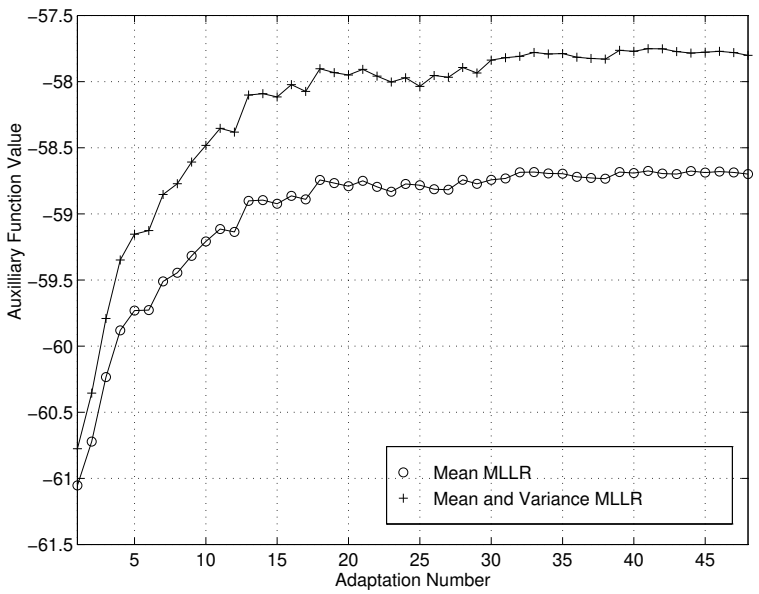

Figure 1: Auxiliary function value obtained with and without adapting the variances for speaker 4tb on the S4 task

Figure 1 shows how the auxiliary function value (see equation 13) of the adaptation data varies with the number of adaptation updates for a MLLR mean adapted model set and a MLLR mean and variance adapted system for speaker *4tb* from the S4 task[7]. From

[7]Since this is an incremental task the alignments for the mean adapted and the mean-and-variance adapted systems will be different after the first model update.

the graph it can be seen that the use of variance adaption showed a distinct improvement in auxiliary function over the standard mean adaptation case. This means that the likelihood of the adaptation data given the mean-and-variance adapted models was greater than that given the mean adapted models. Note, since the system was run in an incremental adaptation mode the likelihoods did not increase monotonically as new adaptation data was added at each update. Although figure 1 shows that the variance adaptation increased the likelihood of the adaptation data, this does not necessarily indicate a reduction in word error rate on the test data.

Transform	Speaker				Average
	4tb	4tc	4td	4te	
None*	5.6	6.3	14.2	4.8	7.7
Mean	5.1	5.8	12.1	4.0	6.7
Mean + Var	4.7	5.5	12.1	4.1	6.6

Table 1: Incremental adaptation results on the S4 evaluation data, * from (Leggetter, 1995)

Table 1 shows the word error rate for the S4 task[8]. The use of mean MLLR adaptation reduced the error rate by 13%. A further decrease of 2% was obtained by also adapting the variances.

MLLR has been shown to improve the recognition performance using only a small amount of adaptation data (Leggetter and Woodland, 1995a). MLLR mean and variance

[8]The results given here differ from those submitted for the CSRNAB 1994 ARPA S4 evaluation as recognition was performed using lattices generated by the unadapted system and the adaptation was performed every other sentence as opposed to every sentence for the evaluation.

Transform Set	Error Rate (%)	
	H1 Dev	H1 Eval
None*	9.5	9.2
Mean	8.0	8.3
Mean + Var	7.9	8.1

Table 2: Incremental adaptation results on H1 development and evaluation data, * from (Woodland et al., 1995)

adaption was therefore examined on the H1 task where relatively few adaptation sentences were available per speaker. The results in table 2 show that on average mean adaptation gave a 13% reduction in error and variance adaptation a further 2% decrease in error rate.

In addition to using MLLR to adapt a model set to a particular speaker, it may also be used to compensate for environmental mismatches. MLLR was therefore applied to two noise corrupted tasks: the S5 and S10 tasks. For both these tasks PMC was used prior to MLLR in order to give initial models. This was found to be important as using clean initial models gave poor alignments for adaptation. In addition, the effects of noise are non-linear so for high noise conditions a large number of linear transforms may be required to adapt the clean models.

On these noise corrupted tasks the improvements gained using mean and variance adaptation over mean adaptation were larger than those obtained in clean conditions (see table 3). On the S5 task mean adaptation reduced the error rate by 17% and variance adaptation yielded an additional 7% reduction. The results on the S5 and S10 tasks compare favourably with the official results (Pallett et al., 1995), where the best S5 performance was 9.7% and the best S10 performance was 12.2%. The best performance

Transform Set	Error Rate (%)	
	S5 Eval	S10 Eval
None	10.3	10.7*
Mean	8.6	9.3
Mean + Var	8.0	8.9

Table 3: Incremental adaptation results on S5 and S10 evaluation data, * from (Gales, 1996)

obtained using just PMC was 10.1% (Gales, 1996), where both the means and variances were compensated.

On all tasks considered the variance compensation gave additional gains of between 2% to 7% over mean-only compensation while mean compensation yielded larger gains of between 13% and 17%. This is not surprising as the number of additional parameters used for variance compensation was small compared to the mean compensation. Furthermore it is commonly accepted that compensating the means will have the greatest effect on performance.

6 Conclusions

A new technique for adapting both the means and the variances of a set of continuous density HMMs within the MLLR framework has been described. The variance transformation may yield a full or diagonal transform, even if the original covariance matrices were diagonal. In either case, it is guaranteed to increase the likelihood of the adaptation data. The computational load of calculating the actual transformation matrix is small.

The technique was evaluated on a variety of large vocabulary recognition tasks. On all tasks variance adaption was found to improve the performance, with gains ranging from 2% to 7%. Though these improvements are small compared to those obtained adapting the means, the results are consistently better at little additional cost in computation. The number of additional parameters introduced is very small. For the full MLLR mean transform a total of $(n+1) \times n$ parameters are used per transform while for the diagonal transformation matrix for the variance only n parameters are used per transform.

For all the experiments described the variance adaptation the same threshold was used for the number of frames in a regression class in order for that class to be adapted despite the small number of parameters used for the variance transformation. Additional improvements may be possible by reducing the number of frames required at a regression class for the variance adaptation. However as the variance adaptation is based on second order statistics, care must be taken to ensure that the transform is robust. In addition the variance transformation matrices were constrained to be diagonal. As this is not a necessary constraint, further gains in performance may be obtained using a full variance transformation matrix.

Acknowledgements

The MLLR variance adaptation code was based on code originally written by Chris Leggetter. Mark Gales is a Research Fellow at Emmanuel College, Cambridge.

References

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–

38.

Digalakis, V. V., Rtischev, D., and Neumeyer, L. G. (1995). Speaker adaptation using constrained estimation of Gaussian mixtures. *IEEE Transactions Speech and Audio Processing*, 3:357–366.

Gales, M. J. F. (1996). *Model-Based Techniques for Noise Robust Speech Recognition*. PhD thesis, Cambridge University.

Gales, M. J. F. and Woodland, P. C. (1996). Variance compensation within the MLLR framework. Technical Report CUED/F-INFENG/TR242, Cambridge University. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

Gales, M. J. F. and Young, S. J. (1995a). A fast and flexible implementation of parallel model combination. In *Proceedings ICASSP*, pages 133–136.

Gales, M. J. F. and Young, S. J. (1995b). Robust speech recognition in additive and convolutional noise using parallel model combination. *Computer Speech and Language*, 9:289–307.

Gauvain, J. L. and Lee, C. H. (1994). Maximum a-posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions Speech and Audio Processing*, 2:291–298.

Hewett, A. J. (1989). *Training and Speaker Adaptation in Template-Based Speech Recognition*. PhD thesis, Cambridge University.

Jaschul, J. (1982). Speaker adaptation by a linear transformation with optimised parameters. In *Proceedings ICASSP*, pages 1657–1670.

Kenny, P., Lenning, M., and Mermelstein, P. (1990). Speaker adaptation in a large-vocabulary Gaussian HMM recogniser. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 12:917–920.

Lee, C. H., Lin, C. H., and Juang, B. H. (1990). A study of speaker adaptation of continuous density HMM parameters. In *Proceedings ICASSP*, pages 145–148.

Leggetter, C. J. (1995). *Improved Acoustic Modelling for HMMs using Linear Transformations*. PhD thesis, Cambridge University.

Leggetter, C. J. and Woodland, P. C. (1994). Speaker adaptation of continuous density HMMs using linear regression. In *Proceedings ICSLP*, pages 451–454.

Leggetter, C. J. and Woodland, P. C. (1995a). Flexible speaker adaptation for large vocabulary speech recognition. In *Proceedings Eurospeech*, pages 1155–1158.

Leggetter, C. J. and Woodland, P. C. (1995b). Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, 9:171–186.

Neumeyer, L. R., Sankar, A., and Digalakis, V. V. (1995). A comparative study of speaker adaptation techniques. In *Proceedings Eurospeech*, pages 1127–1130.

Odell, J. J., Valtchev, V., Woodland, P. C., and Young, S. J. (1994). A one pass decoder design for large vocabulary recognition. In *Proceedings ARPA Workshop on Human Language Technology*, pages 405–410.

Pallett, D. S., Fiscus, J. G., Fisher, W. M., Garofolo, J. S., Lund, B. A., Martin, A., and Przybocki, M. A. (1995). 1994 benchmark tests for the ARPA spoken language

program. In *Proceedings ARPA Workshop on Spoken Language Systems Technology*, pages 5–36.

Rose, R. C., Hofstetter, E. M., and Reynolds, D. A. (1994). Integrated models of signal and background with application to speaker identification in noise. *IEEE Transactions Speech and Audio Processing*, 2:245–257.

Varga, A. P. and Moore, R. K. (1990). Hidden Markov model decomposition of speech and noise. In *Proceedings ICASSP*, pages 845–848.

Woodland, P. C., Gales, M. J. F., and Pye, D. (1996). Improving environmental robustness in large vocabulary speech recognition. In *Proceedings ICASSP*, pages 65–68.

Woodland, P. C., Odell, J. J., Valtchev, V., and Young, S. J. (1995). The development of the 1994 HTK large vocabulary speech recognition system. In *Proceedings ARPA Workshop on Spoken Language Systems Technology*, pages 104–109.

Young, S. J., Odell, J. J., and Woodland, P. C. (1994). Tree-based state tying for high accuracy acoustic modelling. In *Proceedings ARPA Workshop on Human Language Technology*, pages 307–312.