MAXIMUM LIKELIHOOD LINEAR TRANSFORMATIONS FOR HMM-BASED SPEECH RECOGNITION

M.J.F. Gales[1]

May 1997 (revised January 1998)

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

Email: mjfg@watson.ibm.com

Abstract

This paper examines the application of linear transformations for speaker and environmental adaptation in an HMM-based speech recognition system. In particular, transformations that are trained in a maximum likelihood sense on adaptation data are investigated. Other than in the form of a simple bias, strict linear feature-space transformations are inappropriate in this case. Hence, only model-based linear transforms are considered. The paper compares the two possible forms of model-based transforms: (i) unconstrained, where any combination of mean and variance transform may be used, and (ii) constrained, which requires the variance transform to have the same form as the mean transform (sometimes referred to as feature-space transforms). Re-estimation formulae for all appropriate cases of transform are given. This includes a new and efficient "full" variance transform and the extension of the constrained model-space transform from the simple diagonal case to the full or block-diagonal case. The constrained and unconstrained transforms are evaluated in terms of computational cost, recognition time efficiency, and use for speaker adaptive training. The recognition performance of the two model-space transforms on a large vocabulary speech recognition task using incremental adaptation is investigated. In addition, initial experiments using the constrained model-space transform for speaker adaptive training are detailed.

[1]The author is now at the IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

1 Introduction

In recent years there has been a vast amount of work done on estimating and applying linear transformations to HMM-based recognisers [2, 4, 13, 17]. Though not the only possible model adaptation scheme, for example maximum a-posteriori adaptation [10] may be used, linear transforms have been shown to be a powerful tool for both speaker and environmental adaptation. Irrespective of the form of transformation, the ability to adapt model sets with large numbers of parameters with little adaptation is essential. Linear transformations achieve this by assuming that they capture general relationships between the original model set and the current speaker or new acoustic environment. Hence, many model parameters may be adapted using the same transform, even when those parameters have not been observed in the adaptation data. These transformations may be estimated in many ways, but for the purpose of this paper only maximum likelihood (ML) estimation will be considered. Here, the transformation is trained on a particular set of adaptation data, such that it maximises the likelihood of that adaptation data given the current model-set. The theory behind these ML trained transforms is well established [22]. However the actual forms of the transform that have been applied to date are limited, due to the complexity of optimising the transformation parameters. The aim of this paper is to present the various forms of maximum likelihood linear transformations that may be applied to an HMM-based speech recognition system and how they may be simply estimated.

Usually linear transformations are described as being applied in either the *model-space* or *feature-space* [21]. This paper uses the same terminology, but applied in a very strict sense. Thus a feature-space transform is required to only act on the features, it is not allowed to alter the recogniser stage in any way[2]. A variety of linear feature-space transformations for adaptation and compensation for speech recognition have been proposed in the literature [12, 15, 16]. ML training of linear feature-space transformations may be shown to be, not surprisingly, inappropriate for speech recognition (see [7]). In contrast, model-space transformations, which act on the model parameters themselves, have been shown to be useful. There are two main forms of model-space transformation[3]. First, there is the *unconstrained* case (e.g. [13, 9]) where the transforms on the means and variances are unrelated to each other. Alternatively, for the *constrained* case (e.g. [4]), the mean transformation and variance transformation are required to have the same form, other than the bias. Both forms of transform may be used for speaker adaptation [13, 4] and environmental compensation [21, 9].

Re-estimation formulae for both forms of model-space transform are given in this paper. For the unconstrained transform the various cases of variance transform are described. These include a new and efficient variance transform. Extension of the constrained model-space transform from the simple diagonal case to the full or block-diagonal case is also presented. These transforms are then compared in terms of efficiency at run-time and in training the transform.

There has also been much interest in using adaptation techniques in both training and testing [1, 11]. Here, instead of applying the test set adaptation transforms to a speaker-independent model-set they are applied to a model set trained using that adaptation scheme. Thus the model-set used in adaptation should model just the *intra-speaker* variability rather than both the *intra* and *inter-speaker* variability. Speaker adaptive training (SAT) [1] is one such scheme. Standard SAT uses an unconstrained model-space transform of the mean in both training and testing. The use of constrained model-space transforms for SAT is presented here. It yields simple re-estimation formulae, overcoming some of the problems associated with traditional SAT training.

The next section describes the two possible linear model-space transformations. For the unconstrained model-space transform an efficient new variance transform is described. The theory

[2]This disagrees with the "definition" in some papers (e.g. [22]), where the linear "feature-space" transform used is a constrained model-space transformation described in section 2.2. The descriptions of the transforms given is more consistent with that of [4]. However for non-linear transformations this definition does not permit a set of possibly useful transformations.

[3]Here the terms *constrained* and *unconstrained* refer to the form of variance transform and are not related to the use of constrained as used in [4] where it refers to the constraint that many Gaussian components share the same transform.

behind constrained transformations is extended so that full, or block-diagonal, linear transformations may be trained in addition to the diagonal case described in [4]. Various implementation issues involving linear transformations are then detailed including speed and applicability for speaker adaptive training. Finally, experiments on a large vocabulary task are described and conclusions drawn.

2 Linear Model-Space Transformations

As previously described there are two forms of model-space linear transformation. First an unconstrained transformation may be used where the mean transformation and the variance transform are independent of one another. Alternatively a constrained transform may be used, where the transformation of the variance must correspond to that applied to the mean. Both these transforms are described in detail below.

In all cases the parameters of the linear transform are found using an EM approach [3]. The parameters of the transforms are found by optimising the following equation

$$\begin{aligned}&\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) = \\ &\quad K - \frac{1}{2}\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[K^{(m)} + \log(|\hat{\mathbf{\Sigma}}^{(m)}|) + (\mathbf{o}(\tau)-\hat{\mu}^{(m)})^T\hat{\mathbf{\Sigma}}^{(m)-1}(\mathbf{o}(\tau)-\hat{\mu}^{(m)})\right]\end{aligned} \tag{1}$$

where $\hat{\mu}^{(m)}$ and $\hat{\mathbf{\Sigma}}^{(m)}$ are the transformed mean and variance for Gaussian component m (the superscript $^{(m)}$ will be used to indicate the Gaussian component for the model parameters), M is the total number of Gaussian components associated with the particular transform, and the posterior probability, $\gamma_m(\tau)$ is

$$\gamma_m(\tau) = p(q_m(\tau)|\mathcal{M},\mathbf{O}_T) \tag{2}$$

$q_m(\tau)$ indicates Gaussian component m at time τ. K is a constant dependent only on the transition probabilities, $K^{(m)}$ is the normalisation constant associated with Gaussian component m, and $\mathbf{O}_T = \{\mathbf{o}(1),\dots,\mathbf{o}(T)\}$ is the adaptation data on which the transform is to be trained.

2.1 Unconstrained Model-Space Transformations

Unconstrained linear model-space transformations allow any transform of the mean and variance. They are therefore more flexible than the constrained case. The general linear transform of the mean, μ, is given by

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

ξ is the extended mean vector, $\begin{bmatrix} 1 & \mu^T \end{bmatrix}^T$, and $\mathbf{W}$ is the extended transform, $\begin{bmatrix} \mathbf{b}^T & \mathbf{A}^T \end{bmatrix}^T$. The variance transform may be modified either using

$$\hat{\mathbf{\Sigma}} = \mathbf{L}\mathbf{H}\mathbf{L}^T \tag{4}$$

where $\mathbf{L}$ is the Choleski factor of the original covariance matrix $\mathbf{\Sigma}$, or

$$\hat{\mathbf{\Sigma}} = \mathbf{H}\mathbf{\Sigma}\mathbf{H}^T \tag{5}$$

In both cases $\mathbf{H}$ is the transformation matrix to be obtained. Solutions for various specific cases of these general transforms can be obtained.

2.1.1 Mean transform

The general transformation of the mean may be solved [9]. The following equation is used

$$\text{vec}(\mathbf{Z}) = \left(\sum_{m=1}^{M} \text{kron}(\mathbf{V}^{(m)}, \mathbf{D}^{(m)})\right) \text{vec}(\mathbf{W}) \tag{6}$$

where vec(.) converts a matrix to a vector ordered in terms of the rows, kron(.) is the Kronecker product,

$$\mathbf{V}^{(m)} = \sum_{\tau=1}^{T} \gamma_m(\tau)\boldsymbol{\Sigma}^{(m)-1} \tag{7}$$

$$\mathbf{Z} = \sum_{m=1}^{M}\sum_{\tau=1}^{T} \gamma_m(\tau)\boldsymbol{\Sigma}^{(m)-1}\mathbf{o}(\tau)\xi^{(m)T} \tag{8}$$

and

$$\mathbf{D}^{(m)} = \xi^{(m)}\xi^{(m)T} \tag{9}$$

Solving this expression is computationally expensive as it involves inverting an $(n^2+n)\times(n^2+n)$ matrix. In [13] the case of the general linear transformation of the means is solved for the diagonal covariance case. This is known as maximum likelihood linear regression (MLLR). It is shown that the i^{th} of the inverse of the transform is given by

$$\hat{\mathbf{w}}_i^T = \mathbf{G}^{(i)-1}\mathbf{k}^{(i)T} \tag{10}$$

where

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}}\xi^{(m)}\xi^{(m)T}\sum_{\tau=1}^{T}\gamma_m(\tau) \tag{11}$$

and

$$\mathbf{k}^{(i)} = \sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\frac{1}{sigma_i^{(m)2}}o_i(\tau)\xi^{(m)T} \tag{12}$$

Equation 10 requires the inverse of an $(n+1)\times(n+1)$ matrix[4]. If an approximate solution to the estimation of the mean constrained model-space transformation is available, then it is possible to iteratively refine this solution rather than starting from scratch. Considering only the diagonal covariance matrix case and differentiating with respect to w_{ij} gives

$$\frac{\partial\mathcal{Q}(\mathcal{M}\hat{\mathcal{M}})}{\partial w_{ij}} = \sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\frac{1}{\sigma_i^{(m)2}}\left(o_i(\tau) - \mathbf{w}_i\xi^{(m)}\right)\xi_j^{(m)T} \tag{13}$$

Using the definition of $\mathbf{G}^{(i)}$ as given in equation 11 and equating to zero, this may be expressed as

$$w_{ij} = \frac{z_{ij} - \sum_{k\neq j} w_{ik}g_{ik}^{(i)}}{g_{ij}^{(i)}} \tag{14}$$

[4]The cost of diagonal covariance transforms may be compared with the cost of full covariance cases. For the full case, using standard inversion routines, the inversion takes $\mathcal{O}(n^6)$ operations. This may contrasted with the cost of the diagonal case, MLLR, which is $\mathcal{O}(n^4)$ operations.

where $\mathbf{Z}$ is defined in equation 8. At each iteration this is guaranteed to increase the likelihood[5]. As this is an indirect optimisation solution, it is not possible to state the number of iterations required for a "good" solution, however there is now no need for inverting $\mathbf{G}^{(i)}$.

2.1.2 Variance transform

When the variance is to be transformed in addition to the means, the optimisation is performed in two stages [9]. First the mean transformation is found, given the current variance (and variance transform). Second the variance transform is found given the current mean (and mean transform). The whole process may then be repeated. Thus the following set of inequalities are set up.

$$\mathcal{L}(\mathbf{O}_T|\check{\mathcal{M}}) \geq \mathcal{L}(\mathbf{O}_T|\hat{\mathcal{M}}) \geq \mathcal{L}(\mathbf{O}_T|\mathcal{M}) \tag{16}$$

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.

In [17] the case of a bias on the mean with a simple scaling of the variance is described. In [9] this is extended to the case where a general transform of the mean is applied. It should be noted that for the simple diagonal variance transform case, the same results are obtained using either equation 4 or 5. In [9] the form of the variance transform is also extended to the case where non-diagonal transforms are used in equation 4. It is shown that

$$\mathbf{H} = \frac{\sum\limits_{m=1}^{M} \left\{ \mathbf{L}^{(m)T} \left[\sum\limits_{\tau=1}^{T} \gamma_m(\tau)(\mathbf{o}(\tau) - \hat{\mu}^{(m)})(\mathbf{o}(\tau) - \hat{\mu}^{(m)}) \right] \mathbf{L}^{(m)} \right\}}{\sum\limits_{m=1}^{M} \sum\limits_{\tau=1}^{T} \gamma_m(\tau)} \tag{17}$$

Unfortunately, the computational cost associated with recognition using the transform obtained from 17 is high. In fact it is comparable to the full covariance case, though not necessarily with the memory requirements [9], since the likelihood must be calculated as

$$\mathcal{L}\left(\mathbf{o}(\tau); \mu, \mathbf{\Sigma}, \mathbf{A}, \mathbf{b}, \mathbf{H}\right) = \mathcal{N}\left(\mathbf{o}(\tau); \hat{\mu}, \hat{\mathbf{\Sigma}}\right) \tag{18}$$

and $\hat{\mathbf{\Sigma}}$ is now a full covariance matrix.

Alternatively the variance transform described in equation 5 may be used. In appendix A an iterative solution for the non-diagonal variance transform case is given, assuming that the original covariance matrices were diagonal. It is shown that

$$\left(\mathbf{h}^{-1}\right)_i = \mathbf{c}_i \mathbf{G}^{(i)-1} \sqrt{\left(\frac{\sum_{m=1}^{M} \sum_{\tau=1}^{T} \gamma_m(\tau)}{\mathbf{c}_i \mathbf{G}^{(i)-1} \mathbf{c}_i^T} \right)} \tag{19}$$

where

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \sum_{\tau=1}^{T} \gamma_m(\tau) \left(\mathbf{o}(\tau) - \hat{\mu}^{(m)} \right) \left(\mathbf{o}(\tau) - \hat{\mu}^{(m)} \right)^T \tag{20}$$

and $\mathbf{c}_i$ is the i^{th} row vector of the cofactors of $\mathbf{H}^{-1}$. The optimisation described is thus an iterative one over rows, since each row is related to the other rows by the cofactors. It is guaranteed to

[5] This is simple to show as

$$\frac{\partial^2 \mathcal{Q}(\mathcal{M}\hat{\mathcal{M}})}{\partial w_{ij}^2} = (-) \sum_{m=1}^{M} \sum_{\tau=1}^{T} \gamma_m(\tau) \frac{1}{\sigma_i^{(m)2}} \xi_j^{(m)2} \tag{15}$$

this indicating a maximum. Note there is also the constraint that there are no numerical accuracy problems.

increase the likelihood at each iteration. The optimisation has the same form as the semi-tied full-covariance optimisation [8], where an indirect method over the rows was previously presented. The advantage of the indirect method was that it did not involve the inversion of $\mathbf{G}^{(i)}$. In contrast to the variance transform in equation 4, the log-likelihood calculation at run time may be implemented efficiently when the original models have diagonal covariance matrices as

$$\log\left(\mathcal{L}\left(\mathbf{o}(\tau);\mu,\boldsymbol{\Sigma},\mathbf{A},\mathbf{b},\mathbf{H}\right)\right)=\log\left(\mathcal{N}\left(\mathbf{H}^{-1}\mathbf{o}(\tau);\mathbf{H}^{-1}\hat{\mu},\boldsymbol{\Sigma}\right)\right)-\log\left(|\mathbf{H}|\right) \tag{21}$$

Thus by appropriately modifying the means the additional cost at recognition time is just a matrix-vector multiplication and a simple addition.

The transform using a simple bias on the variance [20, 22] is not considered here, as for many situations it can give an inappropriate transformation. For cases where the variance bias is not constrained to be positive any unobserved Gaussian component may end up with negative variances unless some variance flooring is used. Unfortunately constraining the variance bias to be positive is a major restriction as in many cases, particularly with cepstral parameters currently popular in speech recognition, the variance tends to decrease. This is true for both speech corrupted by additive noise and when performing speaker adaptation.

2.2 Constrained Model-Space Transformations

The constrained model-based transform was first described in [4]. Here the transformation applied to the variance must correspond to the transform applied to the means. Thus the general form is

$$\hat{\mu}=\mathbf{A}'\mu-\mathbf{b}' \tag{22}$$

and

$$\hat{\boldsymbol{\Sigma}}=\mathbf{A}'\boldsymbol{\Sigma}\mathbf{A}'^{T} \tag{23}$$

In [4] the problem is solved for the diagonal transformation case. Here, a solution for the full transformation case which is guaranteed to increase the likelihood of the adaptation data is given. It is assumed for this work that the original models to be adapted have diagonal covariance matrices.

Substituting equations 22 and 23 in equation 1 and re-arranging

$$\begin{aligned}\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}})=K-\frac{1}{2}\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\Big[&K^{(m)}\\&+\log(|\boldsymbol{\Sigma}^{(m)}|)-\log(|\mathbf{A}|^2)+(\hat{\mathbf{o}}(\tau)-\mu^{(m)})^T\boldsymbol{\Sigma}^{(m)-1}(\hat{\mathbf{o}}(\tau)-\mu^{(m)})\Big]\end{aligned} \tag{24}$$

where

$$\hat{\mathbf{o}}(\tau)=\mathbf{A}'^{-1}\mathbf{o}(\tau)+\mathbf{A}'^{-1}\mathbf{b}'=\mathbf{A}\mathbf{o}(\tau)+\mathbf{b}=\mathbf{W}\zeta(\tau) \tag{25}$$

so $\mathbf{W}$ is again the extended transformation matrix, $\left[\begin{array}{cc}\mathbf{b}^T & \mathbf{A}^T\end{array}\right]^T$, $\zeta(\tau)$ is the extended observation vector, $\left[\begin{array}{cc}1 & \mathbf{o}(\tau)^T\end{array}\right]^T$, $\mathbf{A}'=\mathbf{A}^{-1}$ and $\mathbf{b}=\mathbf{A}\mathbf{b}'$. An iterative solution to this optimisation problem is described in appendix B. It is shown that the i^{th} row of the transform is given by

$$\mathbf{w}_i=\left(\alpha\mathbf{p}_i+\mathbf{k}^{(i)}\right)\mathbf{G}^{(i)-1} \tag{26}$$

where $\mathbf{p}_i$ is the extended cofactor row vector $\left[\begin{array}{cccc}0 & c_{i1} & \dots & c_{in}\end{array}\right]$, ($c_{ij}=\mathrm{cof}(\mathbf{A}_{ij})$),

$$\mathbf{G}^{(i)}=\sum_{m=1}^{M}\frac{1}{\sigma_i^{(m)2}}\sum_{\tau=1}^{T}\gamma_m(\tau)\zeta(\tau)\zeta(\tau)^T \tag{27}$$

$$\mathbf{k}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \mu_i^{(m)} \sum_{\tau=1}^{T} \gamma_m(\tau)\zeta(\tau)^T \tag{28}$$

and α satisfies a simple quadratic expression given in equation 65. Again this is an iterative solution over the rows since the rows of the transform are dependent on one another via the extended cofactor vector $\mathbf{p}_i$. In appendix B an iterative solution over the rows, which does not require inverting $\mathbf{G}^{(i)}$ is also given.

Equation 24 illustrates a possible advantage of the constrained model-space transformation compared to the unconstrained case. The constrained transform may be implemented as a transformation of the observed features and a simple addition of the term $\log(|\mathbf{A}|)$[6]. Thus during recognition the log-likelihoods are calculated as

$$\log\left(\mathcal{L}\left(\mathbf{o}(\tau); \mu, \mathbf{\Sigma}, \mathbf{A}, \mathbf{b}\right)\right) = \log\left(\mathcal{N}\left(\mathbf{A}\mathbf{o}(\tau) + \mathbf{b}; \mu, \mathbf{\Sigma}\right)\right) + \log\left(|\mathbf{A}|\right) \tag{29}$$

Furthermore there is no need to adapt the original model parameters, which may in some circumstances be computationally expensive.

3 Implementation Issues

3.1 Complexity versus specificity

The trade-off between the complexity of the transformation, e.g. full, block-diagonal or diagonal, versus the number of transformations that may be robustly estimated is an important one. Normally the complexity of the transformation is selected then an appropriate number of transforms generated. The question of what the appropriate number of transforms is for a particular set of adaptation data and how the Gaussian components should be grouped together is interesting and is discussed in [6]. The question of complexity versus specificity was also examined in [17], where using an unconstrained block-diagonal mean transformation was shown to be better than a diagonal transformation. This may be contrasted with variances where the use of a diagonal transform was found to be about the same as block-diagonal or full transforms [9], but at a considerably increased computational cost (as the transformation was implemented using equation 4).

3.2 Computational Cost

An important consideration in the choice of adaptation algorithm is the computational load, both in training the transform and during recognition. This is particularly important when training and applying the transforms in an *incremental* adaptation mode[7]. For this section only the unconstrained and constrained model-space transformations with diagonal covariance matrices for the original models will be considered. In both cases the cost of a full transformation matrix will be calculated[8].

There are two distinct computational overheads associated with generating the transforms. The first is the cost of accumulating the appropriate statistics. Second is the cost of estimating the transform having accumulated the appropriate statistics.

1. **Unconstrained model-space transformation**: There is a choice of how the statistics are to be accumulated for the unconstrained model-space transformation [9]. If accumulated at the Gaussian component level it is only necessary to store the vector sum and occupancy for each component. However, prior to estimating the transform it os necessary to generate $\mathbf{G}^{(i)}$ and $\mathbf{k}^{(i)}$ (defined in equations 11 and 12 respectively for each of the n dimensions. This is a function of the number of observed components, requiring $\mathcal{O}(n^3)$ multiply accumulates

[6]This addition is the reason that it is not a feature-space transform, though of course when using a single transformation it does not alter the performance, but is necessary when multiple transformations are used

[7]The adaptation data is made available as the system is used and the models repeatedly adapted.

[8]Both schemes scale in the same way when block-diagonal transforms are used.

for each. Alternatively, $\mathbf{G}^{(i)}$ and $\mathbf{k}^{(i)}$ may be directly accumulated. This is now a function of the number of adaptation frames and number of Gaussian components with significant posterior probabilitied per frame. This is more computationally expensive, but is performed continuously.

When calculating the transform using the direct method with diagonal covariance matrices, it is necessary to invert an $(n+1)$ by $(n+1)$ matrix for each of the dimensions of the transformation matrix. This inversion may be performed in $\mathcal{O}(n^3)$ operations[9]. Hence the total cost is approximately $\mathcal{O}(n^4)$ operations per transform. After the transformation has been estimated $\mathcal{O}(Mn^2)$ operations are required to transform the model means. At run-time there is no additional cost. Using the indirect method there is no need for the inversion. The cost of each iteration is cheap, however the number of iterations required depends on how good the initial estimate is. If a diagonal variance transform is to also be used, the cost of calculating the transform is minimal (equation 4 using only the leading diagonal), with a cost of applying $\mathcal{O}(Mn)$ operations to scale the variances. Again there is no recognition time cost. However, if a full variance transform is to be used there is some additional cost. If the transform as described in equation 4 is used then, though cheap to calculate, there is a large runtime cost as a full covariance matrix likelihood must be calculated per Gaussian component. Alternatively, if the form of equation 5 is used then at runtime the cost is a matrix-vector multiplication per transform per observation vector. In this case the cost estimating the transform is approximately the same as estimating a constrained model-space transform described below.

2. **Constrained model-space transformation**: For the constrained model-space transformation it is normally necessary to store the statistics at the transform level (see section 3.4). This thus requires $\mathcal{O}(n^3)$ multiply accumulates per Gaussian component with a significant posterior probablility per frame. As mentioned this makes the accumulation of the statistics more computationally expensive than the unconstrained case where the statistocs are stored at the component level.

   Using the optimisation scheme described in appendix B, the most expensive operation for each row is the generation of the cofactors. Even a very naive implementation costs only $\frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n^2$ operations per row of the transform. Thus the total cost is approximately $\mathcal{O}(n^4)$ per iteration. This has ignored the actual cost of inverting $\mathbf{G}^{(i)}$ which only needs to be performed once, costing $\mathcal{O}(n^4)$. Unfortunately the constrained case is an indirect optimisation scheme. The total cost then becomes $(I+1)\mathcal{O}(n^4)$, where I is the total number of iterations[10]. In reality of course when using incremental adaptation the new transform estimate will be initialised with the previous one, thus dramatically reducing the required number of iterations. Furthermore, it is not necessary to invert $\mathbf{G}^{(i)}$, as an indirect optimisation over each row may be used. During recognition there is a cost of a matrix-vector multiplication for each transform for each observations, in addition to a simple addition per Gaussian component. Thus, for R transforms this is $\mathcal{O}(TRn^2)$, where T is the total number of observations.

The final choice of the most appropriate transformation, solely considering speed not performance, depends on the application and the nature of the model-set being used. For static adaptation, for example on enrolment, the use of an unconstrained model transformation (with either none or diagonal variance transformation) is good as the adaptation is only performed once and there is no additional recognition time cost. In contrast where incremental adaptation is to be used, a constrained model space transformation is good as there is no need to adapt the actual models themselves.

[9] This may actually be done in $n^{\log_2(7)}$ operations.

[10] In practice by initialising the leading diagonal terms to their diagonal transform values (this is non-iterative) only a couple of iterations were required in the optimisation to obtain "good" transforms.

3.3 Numerical Accuracy

For the general unconstrained mean transformation case with a diagonal covariance matrix, numerical accuracy problems occur during the inversion of of the term $\mathbf{G}^{(i)}$ where

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \xi^{(m)} \xi^{(m)T} \sum_{\tau=1}^{T} \gamma_m(\tau) \tag{30}$$

and $\xi^{(m)}$ is the extended mean vector. It is simple to see that when $M < n$, $\mathbf{G}^{(i)}$ cannot have full rank. This problem can be easily handled by using singular value decomposition (SVD), where eigenvalues that are below the accuracy of the machine are set to zero [12].

A similar situation may occur for the constrained model-space transform, or when calculating the efficient full variance transform for the unconstrained case. Again the numerical accuracy problem manifests itself when inverting $\mathbf{G}^{(i)}$, though now this has the form

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \sum_{\tau=1}^{T} \gamma_m(\tau) \zeta(\tau) \zeta(\tau)^T \tag{31}$$

There are two solutions to this problem. The first is to use block-diagonal transformations, thus dramatically reducing the chance of non-full rank matrices. Alternatively SVD may again be used.

3.4 Statistics Required

An issue in the practical implementation of estimating the transform is the statistics required. For the unconstrained case details of possible storage options are detailed in [9]. If only a mean transform is to be used then either $\mathcal{O}(n^3)$ parameters at the transform level, or $\mathcal{O}(n)$ parameters at the Gaussian component level. For the constrained case, if implemented directly it is necessary to store $\mathcal{O}(n^2)$, where n is the dimension of the feature vector, parameters per Gaussian component. This can very rapidly become expensive in terms of memory as the number of Gaussian components increases. Alternatively, the optimisation in appendix B is expressed in terms of $\mathbf{G}^{(i)}$ along with $\mathbf{k}^{(i)}$ and an occupation count at the transform level. It is thus only necessary to store $\mathcal{O}(n^3)$ counts per transform and $\mathcal{O}(n)$ per Gaussian component to estimate the transform. As there are typically far fewer transforms than Gaussian components this is an efficient way of storing the statistics.

4 Speaker Adaptive Training

Recently there has been much interest in using adaptation techniques in both training and testing. When using these techniques, instead of applying the test set adaptation transforms to a speaker-independent model-set they are applied to a model set trained using that adaptation scheme. Two currently popular transforms used are vocal tract normalisation (VTN) [11] and speaker adaptive training (SAT) [1]. The gains obtained using VTN have been shown to be essentially additive to the gains obtained using SAT [19]. This paper does not consider the use of VTN as it is only concerned with linear transformations, though VTN would similarly be expected to improve results quoted here. The standard SAT training uses an unconstrained model-space transformation of the means (MLLR). This section considers the use of a constrained model-space transformation, instead of the standard unconstrained transformation, for this task.

In standard SAT the new mean and variance are given by [1]

$$\hat{\mu}^{(m)} = \left(\sum_{s=1}^{S} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau) \mathbf{A}^{(s)T} \mathbf{\Sigma}^{(m)-1} \mathbf{A}^{(s)} \right)^{-1} \sum_{s=1}^{S} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau) \mathbf{A}^{(s)T} \mathbf{\Sigma}^{(m)-1} \left(\mathbf{o}(\tau) - \mathbf{b}^{(s)} \right) \tag{32}$$

and

$$\hat{\mathbf{\Sigma}}^{(m)} = \frac{\sum_{s=1}^{S} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau)(\mathbf{o}(\tau) - \hat{\mu}^{(sm)})(\mathbf{o}(\tau) - \hat{\mu}^{(sm)})^T}{\sum_{s=1}^{S} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau)} \tag{33}$$

where

$$\hat{\mu}^{(sm)} = \mathbf{A}^{(s)}\hat{\mu}^{(m)} + \mathbf{b}^{(s)} \tag{34}$$

$$\tag{35}$$

and $\{\mathbf{A}^{(s)}, \mathbf{b}^{(s)}\}$ is the transformation associated with speaker s[11]. Unfortunately when implementing these re-estimation formulae there are severe computational and memory overheads [14, 19]. In order to update the means as described in equation 32 it is necessary to store a full, or block-diagonal, matrix for each Gaussian component. This rapidly becomes impractical as the number of Gaussian components used in the system increases. Furthermore it is not possible to perform a simple update of the model means and variances in the same pass.

These problems do not occur when the constrained model-space linear transformation is used in SAT. The re-estimation formulae become almost identical to the standard mean and variance re-estimation formulae[12]. The training of the speaker-dependent constrained transforms is performed as described in section 2.2. The updating of the means and variances involves optimising the following auxiliary function

$$\begin{aligned} \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = K - \frac{1}{2} \sum_{s=1}^{S} \sum_{m=1}^{M} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau) \Big[K^{(m)} \qquad\qquad (36) \\ + \log(|\hat{\mathbf{\Sigma}}^{(m)}|) - 2\log(|\mathbf{A}^{(s)}|) + (\mathbf{A}^{(s)}\mathbf{o}(\tau) + \mathbf{b}^{(s)} - \hat{\mu}^{(m)})^T \hat{\mathbf{\Sigma}}^{(m)-1} (\mathbf{A}^{(s)}\mathbf{o}(\tau) + \mathbf{b}^{(s)} - \hat{\mu}^{(m)}) \Big] \end{aligned}$$

By inspection this is very similar to the standard optimisation task, hence the estimates of the mean and variance will be given by

$$\hat{\mu}^{(m)} = \frac{\sum_{s=1}^{S} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau)\hat{\mathbf{o}}^{(s)}(\tau)}{\sum_{s=1}^{S} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau)} \tag{37}$$

and

$$\hat{\mathbf{\Sigma}}^{(m)} = \frac{\sum_{s=1}^{S} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau)(\hat{\mathbf{o}}^{(s)}(\tau) - \hat{\mu}^{(m)})(\hat{\mathbf{o}}^{(s)}(\tau) - \hat{\mu}^{(m)})^T}{\sum_{s=1}^{S} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau)} \tag{38}$$

where

$$\hat{\mathbf{o}}^{(s)}(\tau) = \mathbf{A}^{(s)}\mathbf{o}(\tau) + \mathbf{b}^{(s)} \tag{39}$$

Thus with the constrained model-space transform the use of speaker adaptive training is simple and requires minimum alteration to the standard code.

5 Results

The results presented in this section are not meant to show a complete comparison of all possible linear model-space transformations trained in a ML fashion. The aim is to compare some possible constrained and unconstrained transformations for speaker adaptation, environmental adaptation and speaker adaptive training.

[11]For simplicity of notation a single transform is assumed per speaker. The extension to multiple transformations is trivial.

[12]The presentation given here considers linear transformations. If the alternative feature-space transformation definition given in [22] is used instead of the strict form presented here, the same re-estimation formulae will result for all the possible feature-space transforms, since the Jacobian will only be a function of the observation not the model parameters.

5.1 Recognition System

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 [23]. 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. Cepstral mean normalisation was then applied to this 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 [24] to define 6399 speech states. For the H1 task a 65k word list and dictionary was used with the trigram language model described in [23]. For the S5 task a 5K vocabulary with trigram language model was used. All decoding used a dynamic-network decoder [18] which can either operate in a single-pass or rescore pre-computed word lattices. A 12 component mixture Gaussian distribution was then trained for each tied state, a total of about 6 million parameters.

For the secondary channel experiments, S5, a PLP version of the standard MFCC models were built using single-pass retraining [5] on the secondary channel training data. This was to ensure that a reasonable initial model set was used in the adaptation process.

All recognition tests were carried out on the 1994 ARPA Hub 1 and S5 evaluation data. The H1 task is an unlimited vocabulary task with approximately 15 sentences per speaker. The data was recorded in a clean[13] environment. The S5 task is an unknown microphone task with a 5k word vocabulary.

For both the static and incremental adaptation experiments the assignment of Gaussian components to transforms was performed using a regression class tree. The classes were determined by grouping Gaussian components in acoustic space as described in [13].

5.2 Constrained versus Unconstrained Transformations

The experiments carried out in this section were run using incremental adaptation. The choice of clustering for the transformations was generated using a regression class tree [13] with the minimum occupancy thresholds empirically derived from similar tasks for both the diagonal and block-diagonal, with separate blocks for the static, delta and delta-delta parameters, transformation cases.

Transform Set	Form	Error Rate (%)		
		H1 Dev	H1 Eval	S5 Eval
—	—	9.57	9.20	8.95
Constrained	Diagonal	8.47	8.48	7.99
	Block	8.14	7.75	7.62
Unconstrained	Diagonal	8.61	8.48	7.93
	Block	8.06	8.13	7.15

Table 1: Incremental adaptation results on H1 development and evaluation data

Table 1 shows the performance of the block-diagonal constrained model-space transform and an unconstrained mean transform run in an incremental adaptation mode. Comparing the two forms of transformation it is hard to obtain a consistent picture. On the evaluation data, the constrained case performs better, on the S5 task the unconstrained case performs better. For the unconstrained case, further slight reductions in word error rate may be obtained by compensating the variances, for example using a diagonal variance transform on the H1 evaluation task the performance was

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

8.04% error rate, and on the S5 task 6.93%[14]. What can be observed from table 1 is that the use of block diagonal transformations, though resulting in far fewer transformations, gave consistently better results than the diagonal transform in all cases.

5.3 Speaker Adaptive Training

All the experiments described in this section were carried out in an unsupervised static mode with the speaker-independent recognition transcriptions used for adaptation. This was not acceptable for the actual evaluation, but was felt to allow better contrasts as the same initial adaptation word transcription can be used for all schemes. In all cases a block-diagonal transform was used with separate blocks for the static, delta and delta-delta parameters. An average of about 125 sentences per speaker were available for the training data and 15 sentences of test data.

Transform Set	Number Transforms	Error Rate (%) H1 Dev	Error Rate (%) H1 Eval
Constrained	1	9.07	7.97
	2	8.64	7.73
Unconstrained	1	8.49	8.30
	2	8.39	8.21

Table 2: Baseline static unsupervised adaptation results

Table 2 shows the performance of the standard SI model set adapted using static unsupervised adaptation on the test data. Only mean adaptation was used for the unconstrained case. For unsupervised static adaptation it is again hard to assess whether a constrained transform is better or worse than an unconstrained one. The unconstrained transform performs better on the development data, the constrained transform performed better on the evaluation data. This again indicates that in terms of performance the two types of transform are comparable.

Only unconstrained mean adaptation (standard MLLR) is considered in table 2. Variance adaptation further improves performance. Using two diagonal variance transforms the error rate on the evaluation data dropped from 8.21% to 8.04%. Using the efficient variance full variance transform (implemented using a block diagonal transform similar to the means), equation 5, this dropped to 7.70%. Even with variance adaptation the performance of the two schemes, constrained and unconstrained, is comparable.

The SAT training routine used in these experiments was as follow:

1. Start with the speaker independent model set and an identity matrix transformation;
2. Estimate a speaker-dependent constrained transform given current model set;
3. Estimate new model set given current speaker-dependent transform using two iterations of Baum-Welch re-estimation (updating all the model parameters);
4. Goto step 2.

For the experiments presented here only a single speaker-dependent transform was used during training. During recognition two passes through the data using the speaker-independent transcription was performed with the SAT models. The first was used to obtain a single transform for the speaker with the SAT model. The alignments for this were felt not to be optimum[15], so an additional pass using this transform with the same transcription to obtain the alignments was used to generate transforms used for recognition.

[14] The more complex variance transforms were not considered for this incremental task as the increased minimum occupancy considerably increases the number of sentences that are recognised before any transforms may be robustly estimated.

[15] In practice this was found to only make a small difference.

Transform Set	Speaker Adapt Iteration	Number Transforms	Error Rate (%) H1 Dev	H1 Eval
Constrained	1	1	8.42	7.44
		2	8.23	7.22
Constrained	2	1	8.26	7.26
		2	8.00	7.09

Table 3: Speaker adaptive models static unsupervised adaptation results

Table 3 shows the results on the H1 task. On the first iteration of speaker adaptive training gains, over applying a constrained transform to the standard speaker-independent models, of 5% and 7% respectively for the development and evaluation data using two transforms were obtained. By using an additional iteration of speaker adaptive these gains were increased to 7% and 8%. This is comparable with gains obtained using unconstrained model-space transforms in the SAT training [1, 19], despite only using a single transform during training.

6 Conclusions

This paper has examined the use of ML trained linear transformations applied to an HMM-based speech recognition system. It has only considered model-space transformations, as it can be shown that a linear feature-space other than as simple bias trained in a ML fashion is not an appropriate transformation. The various forms of model-space linear transformations are investigated. They may be split into two groups: (i) unconstrained where the mean and variance transform are unrelated to one another; (ii) constrained where the variance transform has the same form as the mean transform. For the unconstrained model-space transform solutions to both the mean and variance transforms are derived, with a new efficient form of full variance transform being given. The range of possible constrained model-space transforms is extended beyond the simple diagonal case to the full or block-diagonal case. The performance of these unconstrained and constrained model-space transforms are then compared for both speaker adaptation and environmental adaptation. In both cases the use of block-diagonal transforms out-performed the diagonal transform case. However, it is not clear from the experiments performed whether one or other of the model-space transforms is better in terms of performance.

The use of this constrained transform for speaker adaptive training is also described. Simple re-estimation formulae for both the means and the variances, which avoid many of the problems associated with the use of the unconstrained transform for SAT, may be obtained for this case. Moreover these formulae may be implemented with little change to the standard training scheme. The gains obtained using the constrained transform were similar to the gains reported elsewhere for the unconstrained transform.

Acknowledgements

Mark Gales is funded as a Research Fellow at Emmanuel College, Cambridge. The notation used for the full covariance MLLR transform was suggested by Olivier Cappé of ENST. Sree Balakrishnan gave invaluable help with the optimisations presented in the appendices.

A Unconstrained Variance Optimisation

This section considers the optimisation of the variance transform, $\mathbf{H}$, when using an linear unconstrained model-space transform where the variance has the form

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \mathbf{H}\boldsymbol{\Sigma}^{(m)}\mathbf{H}^T \tag{40}$$

For the optimisation presented here it is assumed that the original covariance matrices are diagonal and that the mean transform has already been found. Instead of estimating $\mathbf{H}$ the inverse is found. Letting

$$\mathbf{A} = \mathbf{H}^{-1} \tag{41}$$

the objective is to maximise the following expression

$$\begin{aligned}\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) = K - \frac{1}{2}\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\Big[K^{(m)} \\ + \log(|\boldsymbol{\Sigma}^{(m)}|) - \log(|\mathbf{A}|^2) + \left(\mathbf{A}\hat{\mathbf{o}}^{(m)}(\tau)\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{A}\hat{\mathbf{o}}^{(m)}(\tau))\right)\Big]\end{aligned} \tag{42}$$

where

$$\hat{\mathbf{o}}^{(m)}(\tau) = \mathbf{o}(\tau) - \hat{\mu}^{(m)} \tag{43}$$

and $\hat{\mu}^{(m)}$ is the estimate of the mean of Gaussian component m given the current mean transform. Usinng the fact that the original covariances matrices are diagonal equation 42 may be written as

$$\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) = \beta\log(\mathbf{c}_i\mathbf{a}_i^T) - \frac{1}{2}\sum_{j=1}^{n}\left(\mathbf{a}_j\mathbf{G}^{(j)}\mathbf{a}_j^T\right) \tag{44}$$

where $\mathbf{a}_i$ is i^{th} row of $\mathbf{A}$, the $1\times n$ row vector $\mathbf{c}_i$ is defined $c_{ij} = \mathrm{cof}(\mathbf{A}_{ij})$ and $\mathbf{G}^{(i)}$ is defined as

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M}\frac{1}{\sigma_i^{(m)2}}\sum_{\tau=1}^{T}\gamma_m(\tau)\left(\hat{\mathbf{o}}^{(m)}(\tau)\right)\left(\hat{\mathbf{o}}^{(m)}(\tau)\right)^T \tag{45}$$

Differentiating with respect to $\mathbf{a}_i$ and equating to zero yields

$$\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\frac{\mathbf{c}_i}{\mathbf{c}_i\mathbf{a}_i^T} = \mathbf{a}_i\mathbf{G}^{(i)} \tag{46}$$

where the Rearranging yields

$$\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\mathbf{c}_i\mathbf{G}^{(i)-1} = \mathbf{c}_i\mathbf{a}_i^T\mathbf{a}_i \tag{47}$$

It is simple to see that $\mathbf{a}_i$ must be in the direction of $\mathbf{c}_i\mathbf{G}^{(i)-1}$. Letting $\mathbf{a}_i = \alpha\mathbf{c}_i\mathbf{G}^{(i)-1}$ gives

$$\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\mathbf{c}_i\mathbf{G}^{(i)-1} = \alpha^2\mathbf{c}_i\mathbf{G}^{(i)-1}\mathbf{c}_i^T\mathbf{c}_i\mathbf{G}^{(i)-1} \tag{48}$$

Therefore

$$\alpha = \pm\sqrt{\left(\frac{\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)}{\mathbf{c}_i\mathbf{G}^{(i)-1}\mathbf{c}_i^T}\right)} \tag{49}$$

Only the positive root is considered[16], hence the final solution for row i is

$$\mathbf{a}_i = \mathbf{c}_i \mathbf{G}^{(i)-1} \sqrt{\left(\frac{\sum_{m=1}^{M} \sum_{\tau=1}^{T} \gamma_m(\tau)}{\mathbf{c}_i \mathbf{G}^{(i)-1} \mathbf{c}_i^T} \right)} \tag{50}$$

The optimisation is thus an iterative one, where each row of $\mathbf{A}$ is optimised given the current value of all the other rows.

The solution presented here is a direct method over the rows and indirect over the columns. The optimisation has the same form as the semi-tied full-covariance matrix optimisation [8] where an indirect method over the rows was presented.

B Constrained Model-Space Optimisation

The objective is to maximise the following expression with respect to $\mathbf{A}$ and $\mathbf{b}$

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = K - \frac{1}{2} \sum_{m=1}^{M} \sum_{\tau=1}^{T} \gamma_m(\tau) \Big(K^{(m)} \tag{51}$$
$$+ \log(|\mathbf{\Sigma}^{(m)}|) - \log(|\mathbf{A}|^2) + (\mathbf{A}\mathbf{o}(\tau) + \mathbf{b} - \mu^{(m)})^T \mathbf{\Sigma}^{(m)-1} (\mathbf{A}\mathbf{o}(\tau) + \mathbf{b} - \mu^{(m)}) \Big)$$

Let $\mathbf{W}$ be the extended transformation matrix, $\begin{bmatrix} \mathbf{b}^T & \mathbf{A}^T \end{bmatrix}^T$, and $\zeta(\tau)$ be the extended observation vector, $\begin{bmatrix} 1 & \mathbf{o}(\tau)^T \end{bmatrix}^T$, thus

$$\hat{\mathbf{o}}(\tau) = \mathbf{A}\mathbf{o}(\tau) + \mathbf{b} = \mathbf{W}\zeta(\tau) \tag{52}$$

Using the fact that only diagonal covariance matrices are being considered, it is possible to rewrite equation 51 as (ignoring all terms independent of $\mathbf{W}$)

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \beta \log(\mathbf{p}_i \mathbf{w}_i^T) - \frac{1}{2} \sum_{j=1}^{n} \left(\mathbf{w}_j \mathbf{G}^{(j)} \mathbf{w}_j^T - 2\mathbf{w}_j \mathbf{k}^{(j)T} \right) \tag{53}$$

where $\mathbf{p}_i$ is the extended cofactor row vector $\begin{bmatrix} 0 & c_{i1} & \dots & c_{in} \end{bmatrix}$, (again $c_{ij} = \mathrm{cof}(\mathbf{A}_{ij})$),

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \sum_{\tau=1}^{T} \gamma_m(\tau) \zeta(\tau) \zeta(\tau)^T \tag{54}$$

$$\mathbf{k}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \mu_i^{(m)} \sum_{\tau=1}^{T} \gamma_m(\tau) \zeta(\tau)^T \tag{55}$$

and

$$\beta = \sum_{m=1}^{M} \sum_{\tau=1}^{T} \gamma_m(\tau) \tag{56}$$

Differentiating with respect to $\mathbf{w}_i$ yields

$$\frac{\partial \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})}{\partial \mathbf{w}_i} = \beta \frac{\mathbf{p}_i}{\mathbf{p}_i \mathbf{w}_i^T} - \mathbf{w}_i \mathbf{G}^{(i)} + \mathbf{k}^{(i)} \tag{57}$$

The optimisation is on a row by row basis, noting that after optimisation each row it is necessary to update the cofactor vector $\mathbf{c}_i$ for the new row i to be optimised.

[16] It makes no difference whether the positive or negative root is selected as they will yield the same likelihood.

B.1 Direct Method over Rows

Assuming that the determinant of $\mathbf{A}$ is non-zero and equating to zero for row i,

$$\beta \frac{\mathbf{p}_i}{\mathbf{p}_i \mathbf{w}_i^T} = \mathbf{w}_i \mathbf{G}^{(i)} - \mathbf{k}^{(i)} \tag{58}$$

Rearranging yields

$$\mathbf{p}_i \mathbf{w}_i^T \mathbf{k}^{(i)} \mathbf{G}^{(i)-1} + \beta \mathbf{p}_i \mathbf{G}^{(i)-1} = \mathbf{p}_i \mathbf{w}_i^T \mathbf{w}_i \tag{59}$$

Considering the direction of the row vector $\mathbf{w}_i$ it is simple to see that

$$\mathbf{w}_i = \alpha \left(\mathbf{p}_i \mathbf{G}^{(i)-1} + \lambda \mathbf{k}^{(i)} \mathbf{G}^{(i)-1} \right) \tag{60}$$

The task is now to find α and λ. Substituting this expression for $\mathbf{w}_i$ and post-multiplying by $\mathbf{G}^{(i)}$ yields

$$\alpha \mathbf{p}_i \mathbf{G}^{(i)-1} \left(\mathbf{p}_i^T + \lambda \mathbf{k}^{(i)T} \right) \mathbf{k}^{(i)} + \beta \mathbf{p}_i = \alpha^2 \mathbf{p}_i \mathbf{G}^{(i)-1} \left(\mathbf{p}_i^T + \lambda \mathbf{k}^{(i)T} \right) \left(\mathbf{p}_i + \lambda \mathbf{k}^{(i)} \right) \tag{61}$$

This may be re-arranged to

$$\left(\beta - \alpha^2 \mathbf{p}_i \mathbf{G}^{(i)-1} \left(\mathbf{p}_i^T + \lambda \mathbf{k}^{(i)T} \right) \right) \mathbf{p}_i = \alpha \left(\lambda \alpha - 1 \right) \mathbf{p}_i \mathbf{G}^{(i)-1} \left(\mathbf{p}_i^T + \lambda \mathbf{k}^{(i)T} \right) \mathbf{k}^{(i)} \tag{62}$$

For this equality to always hold, it is necessary that

$$\lambda \alpha = 1 \tag{63}$$

and

$$\beta = \alpha^2 \mathbf{p}_i \mathbf{G}^{(i)-1} \left(\mathbf{p}_i^T + \lambda \mathbf{k}^{(i)T} \right) \tag{64}$$

Rearranging this and substituting in equation 63 yields

$$\alpha^2 \mathbf{p}_i \mathbf{G}^{(i)-1} \mathbf{p}_i^T + \alpha \mathbf{p}_i \mathbf{G}^{(i)-1} \mathbf{k}^{(i)T} - \beta = 0 \tag{65}$$

This is a simple quadratic expression in α and may be solved in the usual way. There will again be two possible solutions, so there is the question of which root to select. It is simple to show that both roots are maxima. Substituting

$$\mathbf{w}_i = \left(\alpha \mathbf{p}_i + \mathbf{k}^{(i)} \right) \mathbf{G}^{(i)-1} \tag{66}$$

into equation 53 and ignoring all the terms independent of α yields

$$\mathcal{Q}^{(i)}(\mathcal{M}, \hat{\mathcal{M}}) = \beta \log(|\alpha \epsilon_1 + \epsilon_2|) - \frac{1}{2} \alpha^2 \epsilon_1 \tag{67}$$

where

$$\epsilon_1 = \mathbf{p}_i \mathbf{G}^{(i)-1} \mathbf{p}_i^T \tag{68}$$

and

$$\epsilon_2 = \mathbf{p}_i \mathbf{G}^{(i)-1} \mathbf{k}^{(i)T} \tag{69}$$

and using the two maximum values of α,

$$\mathcal{Q}^{(i)}(\mathcal{M}, \hat{\mathcal{M}}) = \beta \log \left(\left| \frac{\epsilon_2 \pm \sqrt{(\epsilon_2^2 + 4\epsilon_1 \beta)}}{2} \right| \right) - \frac{\epsilon_1}{2} \left(\frac{-\epsilon_2 \pm \sqrt{(\epsilon_2^2 + 4\epsilon_1 \beta)}}{2\epsilon_1} \right)^2 \tag{70}$$

As it is not possible to ensure that $\epsilon_2 > 0$, the value of α is selected that maximises $\mathcal{Q}^{(i)}(\mathcal{M}, \hat{\mathcal{M}})$.

The optimisation presented here is an iterative one, since it performs a row by row optimisation and each row is dependent on the other rows via its cofactors. The total number of iterations required will depend on the start point.

B.2 Indirect Method over Rows

Using the optimisation in the previous section requires the inverse of $\mathbf{G}^{(i)}$ to be calculated for all dimensions. If an initial solution which is felt to be "close" to the actual solution is known then an alternative solution is possible, which does not require this inversion.

Consider only element w_{ij}.

$$\frac{\partial \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})}{\partial w_{ij}} = \beta \frac{p_{ij}}{\mathbf{p}_i \mathbf{w}_i^T} - \mathbf{w}_i \mathbf{g}_j^{(i)T} + k_j^{(i)} \tag{71}$$

Equating this expression to one and re-arranging into the form

$$\epsilon_1 w_{ij}^2 - \epsilon_2 w_{ij} - \epsilon_3 = 0 \tag{72}$$

where

$$\begin{aligned}
\epsilon_1 &= p_{ij} g_{jj}^{(i)} \\
\epsilon_2 &= p_{ij} \left(k_j^{(i)} - \sum_{l \neq j} w_{il} g_{lj}^{(i)} \right) - |\mathbf{A}|^{(\bar{j})} g_{jj}^{(i)} \\
\epsilon_3 &= \beta p_{ij} + |\mathbf{A}|^{(\bar{j})} \left(k_j^{(i)} - \sum_{l \neq j} w_{il} g_{lj}^{(i)} \right)
\end{aligned}$$

and

$$|\mathbf{A}|^{(\bar{j})} = \sum_{l \neq j} w_{il} p_{il} \tag{73}$$

Solving this is a standard problem, thus

$$w_{ij} = \frac{\epsilon_2 \pm \sqrt{\epsilon_2^2 + 4\epsilon_1\epsilon_3}}{2\epsilon_1} \tag{74}$$

There are two solutions, so there is the question of which root is to be chosen. Similar arguments to the direct method are used to select the root.

References

[1] T Anastasakos, J McDonough, R Schwartz, and J Makhoul. A compact model for speaker-adaptive training. In *Proceedings ICSLP*, pages 1137–1140, 1996.

[2] S J Cox and J S Bridle. Unsupervised speaker adaptation by probabilistic spectrum fitting. In *Proceedings ICASSP*, pages 294–297, 1989.

[3] A P Dempster, N M Laird, and D B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

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

[5] M J F Gales. *Model-Based Techniques for Noise Robust Speech Recognition.* PhD thesis, Cambridge University, 1995.

[6] M J F Gales. The generation and use of regression class trees for MLLR adaptation. Technical Report CUED/F-INFENG/TR263, Cambridge University, 1996. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

[7] M J F Gales. Maximum likelihood linear transformations for HMM-based speech recognition. Technical Report CUED/F-INFENG/TR291, Cambridge University, 1997. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

[8] M J F Gales. Semi-tied full-covariance matrices for hidden Markov models. Technical Report CUED/F-INFENG/TR287, Cambridge University, 1997. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

[9] M J F Gales and P C Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, 10:249–264, 1996.

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

[11] L Lee and R C Rose. Speaker normalisation using efficient frequency warping procedures. In *Proceedings ICASSP*, volume 1, pages 353–356, 1996.

[12] C J Leggetter. *Improved Acoustic Modelling for HMMs using Linear Transformations.* PhD thesis, Cambridge University, 1995.

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

[14] S Matsoukas, R Schwartz, H Jin, and L Nguyen. Practical implementations of speaker-adaptive training. In *Proceedings DARPA Speech Recognition Workshop*, 1997.

[15] J Neto, L Almeida, M M Hochberg, C Martins, L Nunes, S J Renals, and A J Robinson. Unsupervised speaker-adaptation for hybrid HMM-MLP continuous speech recognition system. In *Proceedings Eurospeech*, pages 187–190, 1995.

[16] L Neumeyer and M Weintraub. Probabilistic optimum filtering for robust speech recognition. In *Proceedings ICASSP*, volume 1, pages 417–420, 1994.

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

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

[19] D Pye and P C Woodland. Experiments in speaker normalisation and adaptation for large vocabulary speech recognition. In *Proceedings ICASSP*, pages 1047–1050, 1997.

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

[21] A Sankar and C H Lee. Robust speech recognition based on stochastic matching. In *Proceedings ICASSP*, pages 121–124, 1995.

[22] A Sankar and C H Lee. A maximum likelihood approach to stochastic matching for robust speech recognition. *IEEE Transactions Speech and Audio Processing*, 4:190–202, 1996.

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

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