Computer Speech and Language (1998) **12,** 75–98

Maximum likelihood linear transformations for HMM-based speech recognition

M. J. F. Gales

Cambridge University Engineering Department, Trumpington Street, Cambridge CB2 1PZ, England

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. Only model-based linear transforms are considered, since, for linear transforms, they subsume the appropriate feature–space transforms. 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. 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. Introduction

In recent years there has been a vast amount of work done on estimating and applying linear transformations to HMM-based recognizers (Cox & Bridle, 1989; Digalakis, Rtischev & Neumeyer, 1995; Leggetter & Woodland, 1995; Neumeyer, Sankar & Digalakis, 1995). Although not the only possible model adaptation scheme, for example maximum *a posteriori* adaptation (Gauvain & Lee, 1994) 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

Present address: IBM T. J. Watson Center, Yorktown Heights, NY 10598, USA. E-mail: mjfg@watson.ibm.com

0885–2308/98/02075 + 24 $25.00/0/la980043

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 maximizes the likelihood of that adaptation data given the current model set. The theory behind these ML trained transformations is well established (Sankar & Lee, 1996). However, the actual forms of the transform that have been applied to date are limited, due to the complexity of optimizing 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 recognizition system and describe how the transform parameters may be simply estimated.

Usually, linear transformations are described as being applied in either the *model–space* or *feature–space* (Sankar & Lee, 1995). This paper uses the same terminology; it is, however, 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 recognizer stage in any way.[1] A variety of linear feature–space transformations for adaptation and compensation for speech recognition have been proposed in the literature (Neumeyer & Weintraub, 1994; Leggetter, 1995; Neto *et al.*, 1995). ML training of strict linear feature–space transformations may be shown to be inappropriate for speech recognition (see Gales, 1997a). In contrast, model–space transformations, which act on the model parameters themselves, have been shown to reduce word error rates for speaker and environmental adaptation tasks. There are two main forms of model–space transformation.[2] First, there is the *unconstrained* case (e.g. Leggetter & Woodland, 1995; Gales & Woodland, 1996) where the transforms on the means and variances are unrelated to each other. Alternatively, for the *constrained* case (e.g. Digalakis *et al.*, 1995), 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 (Digalakis *et al.*, 1995; Leggetter & Woodland, 1995) and environmental compensation (Sankar & Lee, 1995; Gales & Woodland, 1996). Re-estimation formulae for both types of model–space transform are given in this paper. For the unconstrained transform the various forms of variance transform are described. These include a new and efficient full 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 transformation parameters.

There has also been much interest in using adaptation techniques in both training and testing (Anastasakos, McDonough, Schwartz & Makhoul, 1996; Lee & Rose, 1996). Here, instead of applying the test set adaptation transforms to a speaker–

[1] This disagrees with the "definition" in some papers (e.g. Sankar & Lee, 1996), 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 Digalakis *et al.* (1995). However, for non-linear transformations this definition does not permit a set of possibly useful transformations.

[2] 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 Digalakis *et al.* (1995) where it refers to the constraint that many Gaussian components share the same transform.

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) (Anastasakos *et al.*, 1996) 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.

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 behind constrained transformations is extended so that full, or block–diagonal, linear transformations may be trained in addition to the diagonal case described in Digalakis *et al.* (1995). 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 transform 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 expectation–maximization (EM) approach (Dempster, Laird & Rubin, 1977). The parameters of the transforms are found by optimizing the following equation (Sankar & Lee, 1996)

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

where $\hat{\boldsymbol{\mu}}^{(m)}$ and $\boldsymbol{\Sigma}^{(m)}$ 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) of the adapted model set $\hat{\mathcal{M}}$, M is the total number of Gaussian components associated with the particular transform, $(\,)^T$ represents matrix transpose, $|.|$ matrix determinant, and the posterior probability, $\gamma_m(\tau)$, determined by the original model set $\mathcal{M}$ is

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

where $q_m(\tau)$ indicates Gaussian component m at time τ. K is a constant dependent only on the transition probabilities, $K^{(m)}$ is the normalization constant associated with Gaussian component m, and $\mathbf{O}_T = \{\mathbf{o}(1), \ldots, \mathbf{o}(T)\}$ 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 linear transform of the mean and variance. They are, therefore, more flexible than the constrained case. The general linear transform of the mean, $\boldsymbol{\mu}$, is given by

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

where $\boldsymbol{\xi}$ is the extended mean vector, $[1\ \boldsymbol{\mu}^T]^T$, and $\mathbf{W}$ is the extended transform, $[\mathbf{b}^T\mathbf{A}^T]^T$. The variance parameters may be modified using either

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

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

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

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

2.1.1. Mean transform

The general transformation of the mean may be solved when the speaker–independent model set has full covariance matrices (Gales & Woodland, 1996). The following equation is solved to find $\mathbf{W}$

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

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 accmulating statistics for, and inverting, an $(n^2+n)\times(n^2+n)$ matrix. In Leggetter and Woodland (1995) 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} row of the transform, $\mathbf{w}_i$, is given by

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

where

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

and

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

Equation (10) requires the inverse of an $(n+1) \times (n+1)$ matrix to find each row of the transformation matrix.[3] If an approximate solution to the estimation of the mean constrained model–space trasformation 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 Equation (1) with respect to a particular element of the transformation matrix, w_{ij}, gives

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

Using the definition of $\mathbf{G}^{(i)}$ and $\mathbf{k}^{(i)}$ given in Equations (11) and (12), and equating to zero, this may be expressed as

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

At each iteration this is guaranteed to increase the likelihood.[4] As this is an indirect optimization solution, it is not possible to state the number of iterations required for a "good" solution; however, there is now no need to invert $\mathbf{G}^{(i)}$.

2.1.2. Variance transform

When the variance is to be transformed in addition to the means, the optimization is usually performed in two stages (Gales & Woodland, 1996). First, the mean transformation is found, given the current variance (and possible variance transform).

[3] 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. The cost of the diagonal case, MLLR, is $\mathcal{O}(n^4)$ operations.

[4] This is simple to show as

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

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

Second, the variance transform is found, given the current mean (and possible mean transform). The whole process may then be repeated. Thus, the following set of inequalities is set up:

$$\mathcal{L}(\mathbf{O}_T|\hat{\mathcal{M}}) \geqslant \mathcal{L}(\mathbf{O}_T|\check{\mathcal{M}}) \geqslant \mathcal{L}(\mathbf{O}_T|\mathcal{M}) \tag{15}$$

where the model set $\check{\mathcal{M}}$ has just the means updated to $\hat{\mu}^{(1)}, \ldots, \hat{\mu}^{(M)}$ and the model set $\hat{\mathcal{M}}$ has both the means and the variances $\hat{\mathbf{\Sigma}}^{(1)}, \ldots, \hat{\mathbf{\Sigma}}^{(M)}$ updated. When a simple diagonal variance transform is used, the same results are obtained using either of the types of variance transform in Equation (4) or (5). However, for the full variance transform case the transformations, and final covariance matrices obtained, are different.

In Neumeyer *et al*. (1995) the case of a bias on the mean with a simple scaling of the variance is described. An extension to the case where a general transform of the mean is applied is described in Gales and Woodland (1996). In the same paper the form of the variance transform is further extended to the case where non-diagonal transforms are used with the form described in Equation (4). For the variance transform of the form

$$\hat{\mathbf{\Sigma}} = \mathbf{L}\mathbf{H}\mathbf{L}^T \tag{16}$$

the ML estimate of $\mathbf{H}$ is shown to be

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

Unfortunately, the computational cost associated with recognition using the transform obtained from Equation (17) is high. In fact it is comparable to the full covariance case, though not necessarily with the memory requirements (Gales & Woodland, 1996), since the likelihood must be calculated as

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

and $\hat{\mathbf{\Sigma}}$ is now a full covariance matrix. This form of variance transform will be referred to as the *normalized–full* variance transform.

Alternatively, the variance transform described in Equation (5) may be used. The transformation is now of the form

$$\hat{\mathbf{\Sigma}} = \mathbf{H}\,\mathbf{\Sigma}\,\mathbf{H}^T. \tag{19}$$

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 the i^{th} row if the inverse of the transformation matrix $\mathbf{H}$, $(\mathbf{h}^{-1})_i$, is given by

$$(\mathbf{h}^{-1})_i = \mathbf{c}_i \mathbf{G}^{(i)-1} \sqrt{\left(\frac{\Sigma_{m=1}^{M} \Sigma_{\tau=1}^{T} \gamma_m(\tau)}{\mathbf{c}_i \mathbf{G}^{(i)-1} \mathbf{c}_i^T} \right)} \tag{20}$$

where

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \sum_{\tau=1}^{T} \gamma_m(\tau) (\mathbf{o}(\tau) - \hat{\boldsymbol{\mu}}^{(m)}) (\mathbf{o}(\tau) - \hat{\boldsymbol{\mu}}^{(m)})^T \tag{21}$$

and $\mathbf{c}_i$ is the i^{th} vector of the cofactors of $\mathbf{H}^{-1}$. The optimizaion described is thus an iterative one over rows, since each row is related to the other rows by the cofactors. However, the scheme is guaranteed to increase the likelihood at each iteration. The optimization has a similar form to the semi-tied full covariance optimization (Gales, 1997b), where an indirect method over the rows was previously presented. In contrast to the normalized–full variance transform, the log–likelihood calculation at run-time may be implemented efficiently when the original models have diagonal covariance matrices. Instead of adapting the diagonal variances of the models (which would result in full covariance matrices), the observations are adapted. The log–likelihood is calculated as

$$\begin{aligned} &\log(\mathscr{L}(\mathbf{o}(\tau); \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{A}, \mathbf{b}, \mathbf{H})) \\ &= \log(\mathscr{N}(\mathbf{H}^{-1}\mathbf{o}(\tau); \mathbf{H}^{-1}\hat{\boldsymbol{\mu}}, \boldsymbol{\Sigma})) - \frac{1}{2}\log(|\mathbf{H}|^2). \end{aligned} \tag{22}$$

Thus, by appropriately modifying the means the additional cost at recognition time is just a matrix–vector multiplication and a simple addition. This form of variance transformation will be referred to as the *efficient–full* variance transform.

The transform using a simple bias on the variance (Rose, Hofstetter & Reynolds, 1994; Sankar & Lee, 1996) 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 the variance tends to decrease, particularly with the cepstral parameters currently popular in speech recognition. This is true for both speech corrupted noise and when performing speaker adaptation.

2.2. Constrained model–space transformations

The constrained model-based transform was first described in Digalakis *et al*. (1995). Here the transformation applied to the variance must correspond to the transform applied to the means. Thus, the general form is

$$\hat{\boldsymbol{\mu}} = \mathbf{A}'\boldsymbol{\mu} - \mathbf{b}' \tag{23}$$

and

$$\hat{\boldsymbol{\Sigma}} = \mathbf{A}' \boldsymbol{\Sigma} \mathbf{A}'^T. \tag{24}$$

In Digalakis *et al.* (1995) the problem is solved for the diagonal transformation case. Here, a solution for the non-diagonal transformation case that 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 (23) and (24) in Equation (1) and rearranging

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = K - \frac{1}{2}\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)[K^{(m)} + \log(|\boldsymbol{\Sigma}^{(m)}|) - \log(|\mathbf{A}|^2) + \hat{\mathbf{o}}(\tau) - \boldsymbol{\mu}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}(\hat{\mathbf{o}}(\tau) - \boldsymbol{\mu}^{(m)})] \tag{25}$$

where

$$\hat{\mathbf{o}}(\tau) = \mathbf{A}'^{-1}\mathbf{o}(\tau) + \mathbf{A}'^{-1}\mathbf{b}' = \mathbf{A}\mathbf{o}(\tau) + \mathbf{b} = \mathbf{W}\boldsymbol{\zeta}(\tau) \tag{26}$$

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

$$\mathbf{w}_i = (\alpha\mathbf{p}_i + \mathbf{k}^{(i)})\mathbf{G}^{(i)-1} \tag{27}$$

where $\mathbf{p}_i$ is the extended cofactor row vector $[0\;c_{i1}\ldots c_{in}]$, $(c_{ij} = \mathrm{cof}(\mathbf{A}_{ij}))$,

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

$$\mathbf{k}^{(i)} = \sum_{m=1}^{M}\frac{1}{\sigma_i^{(m)2}}\mu_i^{(m)}\sum_{\tau=1}^{T}\gamma_m(\tau)\boldsymbol{\zeta}(\tau)^T \tag{29}$$

and α satisfies a simple quadratic expression given in Equation (B1.8). 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.2 an iterative solution over the rows, which does not require inverting $\mathbf{G}^{(i)}$ is also given. This allows an existing transform to be efficiently refined.

Equation (25) illustrates an 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 $\log(|\mathbf{A}|^2)$.[5] Thus, during recognition the log–likelihoods are calculated as

[5] This addition is the reason that it is not a feature–space transform. Of course, when using a single transformation it does not alter the performance, but it is necessary when multiple transformations are used.

$$\log(\mathscr{L}(\mathbf{o}(\tau); \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{A}, \mathbf{b})) = \log(\mathscr{N}(\mathbf{A}\mathbf{o}(\tau) + \mathbf{b}; \boldsymbol{\mu}, \boldsymbol{\Sigma})) + \frac{1}{2}\log(|\mathbf{A}|^2). \tag{30}$$

There is no need to adapt the original model parameters.

3. Implementation issues

3.1. Complexity vs. specificity

The trade-off between the complexity of the transformation (e.g. full, block–diagonal or diagonal) vs. 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 Gales (1996). Clustering Gaussian components according to how "close" they are in acoustic space was found to give good transformation classes. For schemes where a single recognition run is used to generate the transform, a simple minimum occupancy threshold was found to yield good results. The question of complexity vs. specificity was also examined in Neumeyer *et al.* (1995). The performance of a system adapted using unconstrained block–diagonal mean transformations was shown to be better than using many diagonal mean transformations, or a few full mean transformations. This may be contrasted with adapting the variances using a normalized–full variance transform, where the use of many diagonal transforms was found to be about the same performance as block–diagonal or full transforms (Gales, Pye & Woodland, 1996), at a considerably lower computational cost (as the transformation was implemented using a normalized–full variance transform). However, for the efficient–full variance transform performance gains can be obtained over the simple diagonal case.

3.2. Statistics required

An issue in the practical implementation of estimating the transform is the statistics required. For the unconstrained mean transformation case details of possible storage options are detailed in Gales and Woodland (1996). 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, have to be stored. For the constrained case with a full transformation matrix, if implemented directly, it is necessary to store $\mathcal{O}(n^2)$ parameters per Gaussian component, where n is the dimension of the feature vector. This can very rapidly become impractical due to the memory requirements as the number of Gaussian components increases. Alternatively, the optimization in Appendix B may be expressed in terms of $\mathbf{G}^{(i)}$, $\mathbf{k}^{(i)}$ and an occupation count at the transform level. It is thus only necessary to store $\mathcal{O}(n^3)$ counts per transform to estimate the transformation parameters. As there are typically far fewer transforms than Gaussian components this is an efficient way of storing the statistics. When using the efficient–full variance transform the same storage problems as the constrained transform case occur. Again, by storing the statistics at the transform level, it is possible to keep the memory requirements low.

3.3. 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.[6] 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.

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 (Gales and Woodland, 1996). If accumulated at the Gaussian component level it is only necessary to store the vector sum and occupancy for each component. This requires $\mathcal{O}(n)$ operations per Gaussian component that has a significant posterior probability per frame (i.e. $\gamma_m(\tau)$ is greater than some minimum threshold). In addition, prior to estimating the transform it is 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 for each. Alternatively, $\mathbf{G}^{(i)}$ and $\mathbf{k}^{(i)}$ may be directly accumulated. This has a cost of $\mathcal{O}(n^3)$ multiply accumulates for each Gaussian component with significant posterior probability every adaptation frame. This is computationally more expensive, but is performed continuously as each frame is observed. If an efficient–full variance transform is to be estimated then the statistics must be accumulated at a transform level for most reasonably large recognition systems. This requires $\mathcal{O}(n^3)$ multiply accumulates for each transform whose Gaussian components have a significant accumulated posterior probability for a particular frame. In addition, since the mean transform is not known when accumulating the statistics to estimate the variance transform, the change in the mean must be accounted for prior to estimating the transformation parameters.

When calculating the mean transform using the standard non-iterative 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.[7] 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 iterative 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 also 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 are additional costs. If the normalized–full variance

[6] In incremental adaptation, the adaptation data is made available as the system is used. The models must be repeatedly adapted as more data becomes available.

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

transform is used then, though cheap to calculate, there is a large run-time cost as a full covariance matrix likelihood must be calculated per Gaussian component. Alternatively, if the efficient–full variance transform is used, then at run-time the cost is a matrix–vector multiplication per transform per observation vector. In this case the cost of 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 3.2). $\mathcal{O}(n^3)$ multiply accumulates are required per transform with a significant posterior probability per frame.[8] The accumulation of the statistics is more expensive than the unconstrained case where the statistics are stored at the component level.

Using the optimization scheme described in Appendix B, the most expensive operation for each row when calculating the transformation matrix is the generation of the cofactors. Even a very naive implementation costs only $\mathcal{O}(n^3)$ 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 $\mathbf{G}^{(i)}$ for each dimension. This inversion only needs to be performed once, costing $\mathcal{O}(n^4)$ per transform. Unfortunately, the constrained case is an indirect optimization scheme. The total cost then becomes $(I+1)\mathcal{O}(n^4)$, where I is the total number of iterations.[9] In some situations, for example in incremental adaptation, the new transform estimate may be initialized with the previous estimate of the transform, rather than an identity matrix. This can reduce the required number of iterations. During recognition there is a cost of a matrix–vector multiplication for each transform for each observation, in addition to a simple addition per Gaussian component. Thus, for R transforms this is $\mathcal{O}(TRn^2)$ operations, where T is the total number of observations. However, it is now not necessary to adapt the model parameters.

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, when incremental adaptation is to be used, a constrained model space transformation may be good since there is no need to adapt the actual models themselves. However, there is the additional cost in accumulating the statistics to estimate the transformation parameters.

3.4. Numerical accuracy

For the general unconstrained mean transformation case, even with a diagonal covariance matrix, numerical accuracy problems may occur. In order to calculate a

[8] This makes use of the following rearranged version of Equation (28)

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

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

particular row of the transformation matrix, $\mathbf{w}_i$, it is necessary to invert $\mathbf{G}^{(i)}$ where (from Equation (11))

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

and $\boldsymbol{\xi}^{(m)}$ is the extended mean vector. It is simple to see that when $M<n$, $\mathbf{G}^{(i)}$ cannot have full rank. In addition, finite numerical accuracy may also result in the matrix not having 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 (Leggetter, 1995). Alternatively, the transformation may be constrained to be block–diagonal, reducing the effective dimensionality in the inversion.

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 $\mathbf{G}^{(i)}$, which has the form (from Equation (28))

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \sum_{\tau=1}^{T} \gamma_m(\tau) \boldsymbol{\zeta}(\tau) \boldsymbol{\zeta}(\tau)^T. \tag{32}$$

Now, the number of observations is required to be greater than the dimensionality of the observation vector (i.e. $T>n$). However, the same problems of numerical accuracy may occur, which may cause $\mathbf{G}^{(i)}$ to not have full rank. The same solutions as the unconstrained case may be used.

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 normalization (VTN) (Lee & Rose, 1996) and speaker adaptive training (SAT) (Anastasakos *et al.*, 1996). The gains obtained using VTN have been shown to be essentially additive to the gains obtained using SAT (Pye & Woodland, 1997). This paper does not consider the use of VTN as it is only concerned with maximum likelihood trained linear transformations, though VTN would similarly be expected to improve results quoted here. The standard SAT uses an unconstrained model–space transformation of the means (MLLR). This section considers the use of a constrained model–space transformation for this task.

In standard SAT the new mean and variance are given by Anastasakos *et al.* (1996)

$$\hat{\boldsymbol{\mu}}^{(m)} = \left(\sum_{s=1}^{S} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau) \mathbf{A}^{(s)T} \boldsymbol{\Sigma}^{(m)-1} \mathbf{A}^{(s)} \right)^{-1} \sum_{s=1}^{S} \sum_{\tau=1}^{T^{(s)}} \gamma_m(\tau) \mathbf{A}^{(s)T} \boldsymbol{\Sigma}^{(m)-1} (\mathbf{o}(\tau) - \mathbf{b}^{(s)}) \tag{33}$$

and

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \frac{\sum_{s=1}^{S}\sum_{\tau=1}^{T^{(s)}}\gamma_m(\tau)(\mathbf{o}(\tau)-\hat{\boldsymbol{\mu}}^{(sm)})(\mathbf{o}(\tau)-\hat{\boldsymbol{\mu}}^{(sm)})^T}{\sum_{s=1}^{S}\sum_{\tau=1}^{T^{(s)}}\gamma_m(\tau)} \tag{34}$$

where

$$\hat{\boldsymbol{\mu}}^{(sm)} = \mathbf{A}^{(s)}\hat{\boldsymbol{\mu}}^{(m)} + \mathbf{b}^{(s)} \tag{35}$$

and $\{\mathbf{A}^{(s)}, \mathbf{b}^{(s)}\}$ is the transformation associated with speaker s.[10] Unfortunately, when implementing these re-estimation formulae there are severe computational and memory overheads (Matsoukas, Schwartz, Jin & Nguyen, 1997; Pye & Woodland, 1997). In order to update the means as described in Equation (33) it is necessary to store a full (or block–diagonal) matrix for each Gaussian component, or store statistics for each Gaussian component for every speaker. This rapidly becomes impractical as the number of Gaussian components used in the system increases, or the number of speakers in the training data becomes large. Furthermore, it is not possible to perform a simple update of the model means and variances in a single step.

A simple extension to standard SAT is to incorporate an efficient–full variance transform into the training process. The auxiliary function for this modified case becomes

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

where $\mathbf{H}^{(s)}$ is the inverse of the efficient–full variance transform associated with speaker s. This yields an optimization very similar to the standard SAT case, thus

$$\begin{aligned}\hat{\boldsymbol{\mu}}^{(m)} = &\left(\sum_{s=1}^{S}\sum_{\tau=1}^{T^{(s)}}\gamma_m(\tau)\mathbf{A}'^{(s)T}\boldsymbol{\Sigma}^{(m)-1}\,\mathbf{A}'^{(s)}\right)^{-1} \\ &\sum_{s=1}^{S}\sum_{\tau=1}^{T^{(s)}}\gamma_m(\tau)\mathbf{A}'^{(s)T}\boldsymbol{\Sigma}^{(m)-1}\,\mathbf{H}^{(s)}(\mathbf{o}(\tau) - \mathbf{b}^{(s)})\end{aligned} \tag{37}$$

and

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

where

$$\mathbf{A}'^{(s)} = \mathbf{H}^{(s)}\mathbf{A}^{(s)} \tag{39}$$

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

and

$$\hat{\boldsymbol{\mu}}^{(sm)} = \mathbf{A}'^{(s)}\hat{\boldsymbol{\mu}}^{(m)} + \mathbf{H}^{(s)}\mathbf{b}^{(s)}. \tag{40}$$

Although this allows an additional degree of flexibility in the SAT process, this does not overcome the previously mentioned problems with SAT.

The problems of standard SAT 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.[11] The training of the speaker–dependent constrained transforms is performed as described in Section 2.2. The updating of the means and variances involves optimizing the following auxiliary function

$$\begin{aligned}\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = K - \frac{1}{2}\sum_{s=1}^{S}\sum_{m=1}^{M}\sum_{\tau=1}^{T^{(s)}}\gamma_m(\tau)[K^{(m)} + \log(|\hat{\boldsymbol{\Sigma}}^{(m)}|) \\ -2\log(|\mathbf{A}^{(s)}|) + (\mathbf{A}^{(s)}\mathbf{o}(\tau) + \mathbf{b}^{(s)} - \hat{\boldsymbol{\mu}}^{(m)})^T\hat{\boldsymbol{\Sigma}}^{(m)-1}(\mathbf{A}^{(s)}\mathbf{o}(\tau) + \mathbf{b}^{(s)} - \hat{\boldsymbol{\mu}}^{(m)})].\end{aligned} \tag{41}$$

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

$$\hat{\boldsymbol{\mu}}^{(m)} = \frac{\sum_{s=1}^{S}\sum_{\tau=1}^{T^{(s)}}\gamma_m(\tau)\hat{\mathbf{o}}^{(s)}(\tau)}{\sum_{s=1}^{S}\sum_{\tau=1}^{T^{(s)}}\gamma_m(\tau)} \tag{42}$$

and

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \frac{\sum_{s=1}^{S}\sum_{\tau=1}^{T^{(s)}}\gamma_m(\tau)(\hat{\mathbf{o}}^{(s)}(\tau) - \hat{\boldsymbol{\mu}}^{(m)})(\hat{\mathbf{o}}^{(s)}(\tau) - \hat{\boldsymbol{\mu}}^{(m)})^T}{\sum_{s=1}^{S}\sum_{\tau=1}^{T^{(s)}}\gamma_m(\tau)} \tag{43}$$

where

$$\hat{\mathbf{o}}^{(s)}(\tau) = \mathbf{A}^{(s)}\mathbf{o}(\tau) + \mathbf{b}^{(s)}. \tag{44}$$

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 an ML fashion. The aim is to compare some possible constrained and unconstrained transformations for speaker adaptation, environmental adaptation, and speaker adaptive training.

[11] The presentation given here considers linear model–space transformations. If the alternative feature–space transformation definition, given in Sankar and Lee (1996), is used instead of the strict form presented here, the same re-estimation formulae will result for all the possible non-linear 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 (Woodland, Odell, Valtchev & Young, 1995). The speech was parameterized into 12 MFCCs, C_1 to C_{12}, along with normalized log-energy and the first and second differentials of these parameters. This yielded a 39-dimensional feature vector. Cepstral mean normalization was then applied to this vector. The acoustic training data consisted of 36 493 sentences from the SI-284 WSJ0 and WSJ1 sets, and the LIMSI 1993 WSJ lexicon and phone set were used. This gave an average of about 125 sentences per speaker for training. The standard HTK system was trained using decision-tree-based state clustering (Young, Odell & Woodland, 1994) to define 6399 speech states. For the H1 task a 65k word list and dictionary was used with the trigram language model described in Woodland *et al*. (1995). For the S5 task a 5k vocabulary with trigram language model was used. All decoding used a dynamic-network decoder (Odell, Valtchev & Woodland, 1994), 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.

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[12] environment. The S5 task is an unknown microphone task with a 5k word vocabulary. For the secondary channel experiments, S5, a PLP version of the standard MFCC models were built using single-pass retraining (Gales, 1995) on the secondary channel training data. This was to ensure that a reasonable initial model set was used in the adaptation process.

For both the static and incremental adaptation experiments the assignment of Gaussian components to transforms was performed using a regression class tree. The regression classes were determined by grouping Gaussian components in acoustic space as described in Leggetter and Woodland (1995).

5.2. Constrained vs. 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 (Leggetter & Woodland, 1995) with the minimum occupancy thresholds empirically derived from similar tasks for both the diagonal and block–diagonal transformation cases. As expected, the minimum occupancy thresholds for the block–diagonal transform was significantly higher than for the diagonal case. Thus, in all experiments the number of diagonal transforms was far greater than the number of block–diagonal transforms. The block–diagonal transform had separate blocks for the static, delta, and delta–delta parameters.

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

TABLE I. Incremental adaptation results on H1 development and evaluation data and S5 evaluation data

Transform set	Form	Error rate (%)		
		H1 development	H1 evaluation	S5 evaluation
—	—	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 II. Baseline static unsupervised adaptation results

Transform set	Number of transforms	Error rate (%)	
		H1 development	H1 evaluation
Constrained	1	9·07	7·97
	2	8·64	7·73
Unconstrained	1	8·49	8·30
	2	8·39	8·21

Table I shows the performance of the diagonal and block–diagonal constrained model–space transforms and unconstrained mean transform runs in an incremental adaptation mode. Comparing the two forms of transformation it is hard to obtain a consistent picture. On the H1 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 8·04 error rate, and on the S5 task 6·93%.[13] What can be observed from Table I 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 adaptation 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.

Table II shows the baseline performance of the standard speaker–independent model

[13] The more complex variance transforms were not considered for this incremental task as the increased minimum occupancy required to robustly estimate the transform parameters considerably increases the number of sentences that are recognized before any transforms may be robustly estimated. This reduces the effectiveness of these transforms for incremental adaptation tasks with small amounts of data per speaker.

TABLE III. Speaker adaptive models static unsupervised adaptation results

Transform set	Speaker adapt. iteration	Number of transforms	Error rate (%)	
			H1 development	H1 evaluation
Constrained	1	1	8·42	7·44
		2	8·23	7·22
Constrained	2	1	8·26	7·26
		2	8·00	7·09

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, whereas 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 constrained mean adaptation (standard MLLR) is considered in Table II. Variance adaptation further improved performance. Using two diagonal variance transforms the error rate on the evaluation data dropped from 8·21 to 8·04%. Using the efficient–full variance transform (implemented using a block–diagonal transform similar to the means), this dropped to 7·70%. This result may be constrasted with the normalized–full variance transform where no gain in performance was observed using block–diagonal transform over the diagonal case (Gales *et al*., 1996). Even with variance adaptation the performance of the two schemes, constrained and unconstrained, is comparable.

The SAT routine used in these experiments was as follows:

(1) start with the speaker–independent model set and an identity matrix transformation;
(2) estimate a speaker–dependent model–space transform given the 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) go to step 2 until convergence criterion satisfied.

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,[14] so an additional pass using this transform with the same transcription to obtain the alignments was used to generate transforms used for recognition.

Table III shows the results on the H1 task. On the first iteration of speaker adaptive training gains of 5 and 7% respectively for the development and evaluation data using two transforms were obtained over applying a constrained transform to the standard

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

speaker–independent models. By using an additional iteration of speaker adaptive training these gains were increased to between 7 and 8%. This is comparable with gains obtained using unconstrained model–space transforoms in the SAT (Anastasakos *et al.*, 1996; Pye & Woodland, 1997), despite only using a single transform during training.

6. Conclusions

This paper has examined the use of maximum likelihood trained linear transformations applied to an HMM-based speech recognition system. Only model–space transformations are examined, since they subsume the appropriate feature–space transformations. The various forms of model–space linear transformations were investigated. They may be split into two groups: (i) unconstrained, where the mean and variance transform are unrelated to one another; and (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 was extended beyond the simple diagonal case to the full or block–diagonal case. The performance of these unconstrained and constrained model–space transforms were then compared for both speaker adaptation and environmental adaptation. Little difference was seen between constrained and unconstrained transforms. However, in both cases the use of block–diagonal transforms outperformed the diagonal transform case.

The use of the 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 speaker adaptive training, 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.

Mark Gales was 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 optimizations presented in the Appendices.

References

Anastasakos, T., McDonough, J., Schwartz, R. & Makhoul, J. (1996). A compact model for speaker-adaptive training. In *Proceedings of the ICSLP*, Philadelphia, pp. 1137–1140.

Cox, S. J. & Bridle, J. S. (1989). Unsupervised speaker adaptation by probabilistic spectrum fitting. In *Proceedings of the ICASSP*, Glasgow, pp. 294–297.

Dempster, A. P., Laird, N. M. & 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. & Neumeyer, L. G. (1995). Speaker adaptation using constrained estimation of Gaussian mixtures. *IEEE Transactions on Speech and Audio Processing* **3**, 357–366.

Gales, M. J. F. (1995). *Model-Based Techniques for Noise Robust Speech Recognition.* Ph.D. Thesis, Cambridge University.

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

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

Gales, M. J. F. (1997b). Semi-tied full covariance matrices for hidden Markov models. Technical Report

CUED/F-INFENG/TR287, Cambridge University. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.
Gales, M. J. F. & Woodland, P. C. (1996). Mean and variance adaptation within the MLLR framework. *Computer Speech and Language* **10**, 249–264.
Gales, M. J. F., Pye, D. & Woodland, P. C. (1996). Variance compensation within the MLLR framework for robust speech recognition and speaker adaptation. In *Proceedings of the ICSLP*, Philadelphia, pp. 1832–1835.
Gauvain, J. L. & Lee, C. H. (1994). Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing* **2**, 291–298.
Lee, L. & Rose, R. C. (1996). Speaker normalisation using efficient frequency warping procedures. In *Proceedings of the ICASSP*, volume 1, Atlanta, pp. 353–356.
Leggetter, C. J. (1995). *Improved Acoustic Modelling for HMMs using Linear Transformations.* Ph.D. Thesis, Cambridge University.
Leggetter, C. J. & Woodland, P. C. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language* **9**, 171–186.
Matsoukas, S., Schwartz, R., Jin, H. & Nguyen, L. (1997). Practical implementations of speaker-adaptive training. In *Proceedings of the DARPA Speech Recognition Workshop*, Chantilly.
Neto, J., Almeida, L., Hochberg, M. M., Martins, C., Nunes, L., Renals, S. J. & Robinson, A. J. (1995). Unsupervised speaker-adaptation for hybrid HMM–MLP continuous speech recognition system. In *Proceedings of Eurospeech*, Berlin, pp. 187–190.
Neumeyer, L. & Weintraub, M. (1994). Probabilistic optimum filtering for robust speech recognition. In *Proceedings of the ICASSP*, volume 1, Adelaide, pp. 417–420.
Neumeyer, L. R., Sankar, A. & Digalakis, V. V. (1995). A comparative study of speaker adaptation techniques. In *Proceedings of Eurospeech*, Berlin, pp. 1127–1130.
Odell, J. J., Valtchev, V., Woodland, P. C. & Young, S. J. (1994). A one pass decoder design for large vocabulary recognition. In *Proceedings of the ARPA Workshop on Human Language Technology*, Princeton, pp. 405–410.
Pye, D. & Woodland, P. C. (1997). Experiments in speaker normalisation and adaptation for large vocabulary speech recognition. In *Proceedings of the ICASSP*, Munich, pp. 1047–1050.
Rose, R. C., Hofstetter, E. M. & Reynolds, D. A. (1994). Integrated models of signal and background with application to speaker identification in noise. *IEEE Transactions on Speech and Audio Processing* **2**, 245–257.
Sankar, A. & Lee, C. H. (1995). Robust speech recognition based on stochastic matching. In *Proceedings of the ICASSP*, Detroit, pp. 121–124.
Sankar, A. & Lee, C. H. (1996). A maximum likelihood approach to stochastic matching for robust speech recognition. *IEEE Transactions on Speech and Audio Processing* **4**, 190–202.
Woodland, P. C., Odell, J. J., Valtchev, V. & Young, S. J. (1995). The development of the 1994 HTK large vocabulary speech recognition system. In *Proceedings of the ARPA Worshop on Spoken Language Systems Technology*, Austin, pp. 104–109.
Young, S. J., Odell, J. J. & Woodland, P. C. (1994). Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the ARPA Workshop on Human Language Technology*, Princeton, pp. 307–312.

(*Received May 1997 and accepted for publication January 1998*)

Appendix A: Unconstrained variance optimization

This section considers the optimization of the variance transform, **H**, when using a linear unconstrained model–space transform where the transformed variance has the form

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \mathbf{H}\,\boldsymbol{\Sigma}^{(m)}\,\mathbf{H}^{T}. \tag{A.1}$$

For the optimization presented here it is assumed that the original covariance matrices are diagonal and that the mean transform has already been found. Instead of estimating **H** the inverse is found. Letting

$$\mathbf{A} = \mathbf{H}^{-1} \tag{A.2}$$

the objective is to maximize the following expression with respect to A

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = K - \frac{1}{2}\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)[K^{(m)} + \log(|\boldsymbol{\Sigma}^{(m)}|) - \log(|\mathbf{A}|^2) + (\mathbf{A}\hat{\mathbf{o}}^{(m)}(\tau))^T \boldsymbol{\Sigma}^{(m)-1}(\mathbf{A}\hat{\mathbf{o}}^{(m)}(\tau)))] \tag{A.3}$$

where

$$\hat{\mathbf{o}}^{(m)}(\tau) = \mathbf{o}(\tau) - \hat{\boldsymbol{\mu}}^{(m)} \tag{A.4}$$

and $\hat{\boldsymbol{\mu}}^{(m)}$ is the estimate of the mean of Gaussian component m given the current mean transform. Using the fact that the original covariance matrices are diagonal, Equation (A.3) may be rewritten as (ignoring terms independent of $\mathbf{A}$)

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \beta \log(\mathbf{c}_i\mathbf{a}_i^T) - \frac{1}{2}\sum_{j=1}^{n}(\mathbf{a}_j\mathbf{G}^{(j)}\mathbf{a}_j^T) \tag{A.5}$$

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

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M}\frac{1}{\sigma_i^{(m)2}}\sum_{\tau=1}^{T}\gamma_m(\tau)(\hat{\mathbf{o}}^{(m)}(\tau))(\hat{\mathbf{o}}^{(m)}(\tau))^T \tag{A.6}$$

and

$$\beta = \sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau). \tag{A.7}$$

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

$$\beta\frac{\mathbf{c}_i}{\mathbf{c}_i\mathbf{a}_i^T} = \mathbf{a}_i\mathbf{G}^{(i)}. \tag{A.8}$$

Rearranging yields

$$\beta\mathbf{c}_i\mathbf{G}^{(i)-1} = \mathbf{c}_i\mathbf{a}_i^T\mathbf{a}_i. \tag{A.9}$$

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

$$\beta\mathbf{c}_i\mathbf{G}^{(i)-1} = \alpha^2\mathbf{c}_i\mathbf{G}^{(i)-1}\mathbf{c}_i^T\mathbf{c}_i\mathbf{G}^{(i)-1}. \tag{A.10}$$

This expression must be solved for α. This has the solution

$$\alpha = \pm \sqrt{\left(\frac{\beta}{\mathbf{c}_i \mathbf{G}^{(i)-1} \mathbf{c}_i^T} \right)}. \tag{A.11}$$

Only the positive root is considered,[15] hence the final solution for row i is

$$\mathbf{a}_i = \mathbf{c}_i \mathbf{G}^{(i)-1} \sqrt{\left(\frac{\beta}{\mathbf{c}_i \mathbf{G}^{(i)-1} \mathbf{c}_i^T} \right)}. \tag{A.12}$$

The optimization is thus an iterative one, where each row of **A** is optimized 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 optimization has the same form as the semi-tied full covariance matrix optimization (Gales, 1997*b*) where an indirect method over the rows was presented.

Appendix B: Constrained model–space optimization

The objective is to maximize the following expression with respect to **A** and **b**

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

Let **W** be the extended transformation matrix, $[\mathbf{b}^T \mathbf{A}^T]^T$, and $\zeta(\tau)$ be the extended observation vector, $[1\ \mathbf{o}(\tau)^T]^T$, thus

$$\hat{\mathbf{o}}(\tau) = \mathbf{A}\mathbf{o}(\tau) + \mathbf{b} = \mathbf{W}\boldsymbol{\zeta}(\tau). \tag{B.2}$$

Using the fact that only diagonal covariance matrices are being considered, it is possible to rewrite Equation (B.1) as (ignoring all terms independent of **W**)

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \beta \log(\mathbf{p}_i \mathbf{w}_i^T) - \frac{1}{2} \sum_{j=1}^{n} (\mathbf{w}_j \mathbf{G}^{(j)} \mathbf{w}_j^T - 2\mathbf{w}_j \mathbf{k}^{(j)T}) \tag{B.3}$$

where $\mathbf{p}_i$ is the extended cofactor row vector $[0\ c_{i1} \ldots c_{in}]$, (again $c_{ij} = \text{cof}(\mathbf{A}_{ij})$),

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \sum_{\tau=1}^{T} \gamma_m(\tau) \boldsymbol{\zeta}(\tau) \boldsymbol{\zeta}(\tau)^T \tag{B.4}$$

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

$$\mathbf{k}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)2}} \mu_i^{(m)} \sum_{\tau=1}^{T} \gamma_m(\tau) \boldsymbol{\zeta}(\tau)^T \tag{B.5}$$

and

$$\beta = \sum_{m=1}^{M} \sum_{\tau=1}^{T} \gamma_m(\tau). \tag{B.6}$$

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

$$\frac{\partial \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})}{\partial \mathbf{w}_i} = \beta \frac{\mathbf{p}_i}{\mathbf{p}_i \mathbf{w}_i^T} - \mathbf{w}_i \mathbf{G}^{(i)} + \mathbf{k}^{(i)}. \tag{B.7}$$

The optimization will be performed on a row by row basis, noting that after optimization of each row the cofactor vector $\mathbf{c}_i$ must be updated for the new row i to be optimized. Two approaches will be considered, one direct for each row, the other indirect.

Appendix B.1: Direct method over rows

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

$$\beta \frac{\mathbf{p}_i}{\mathbf{p}_i \mathbf{w}_i^T} = \mathbf{w}_i \mathbf{G}^{(i)} - \mathbf{k}^{(i)}. \tag{B1.1}$$

Rearranging yields

$$\mathbf{p}_i \mathbf{w}_i^T \mathbf{k}^{(i)} \mathbf{G}^{(i)-1} + \beta \mathbf{p}_i \mathbf{G}^{(i)-1} = \mathbf{p}_i \mathbf{w}_i^T \mathbf{w}_i. \tag{B1.2}$$

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

$$\mathbf{w}_i = \alpha(\mathbf{p}_i \mathbf{G}^{(i)-1} + \lambda \mathbf{k}^{(i)} \mathbf{G}^{(i)-1}). \tag{B1.3}$$

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

$$\alpha \mathbf{p}_i \mathbf{G}^{(i)-1} (\mathbf{p}_i^T + \lambda \mathbf{k}^{(i)T}) \mathbf{k}^{(i)} + \beta \mathbf{p}_i = \alpha^2 \mathbf{p}_i \mathbf{G}^{(i)-1} (\mathbf{p}_i^T + \lambda \mathbf{k}^{(i)T})(\mathbf{p}_i + \lambda \mathbf{k}^{(i)}). \tag{B1.4}$$

This may be rearranged to

$$(\beta - \alpha^2 \mathbf{p}_i \mathbf{G}^{(i)-1} (\mathbf{p}_i^T + \lambda \mathbf{k}^{(i)T})) \mathbf{p}_i = \alpha(\lambda\alpha - 1) \mathbf{p}_i \mathbf{G}^{(i)-1} (\mathbf{p}_i^T + \lambda \mathbf{k}^{(i)T}) \mathbf{k}^{(i)}. \tag{B1.5}$$

For this equality to always hold, it is necessary that

$$\lambda\alpha = 1 \tag{B1.6}$$

and

$$\beta = \alpha^2 \mathbf{p}_i \mathbf{G}^{(i)-1} (\mathbf{p}_i^T + \lambda \mathbf{k}^{(i)T}). \tag{B1.7}$$

Rearranging this and substituting in Equation (B1.6) yields

$$\alpha^2 \mathbf{p}_i \mathbf{G}^{(i)-1} \mathbf{p}_i^T + \alpha \mathbf{p}_i \mathbf{G}^{(i)-1} \mathbf{k}^{(i)T} - \beta = 0. \tag{B1.8}$$

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

$$\mathbf{w}_i = (\alpha \mathbf{p}_i + \mathbf{k}^{(i)}) \mathbf{G}^{(i)-1} \tag{B1.9}$$

into Equation (B.3) and ignoring all the terms independent of α yields

$$\mathcal{Q}^{(i)}(\mathcal{M}, \hat{\mathcal{M}}) = \beta \log(|\alpha \varepsilon_1 + \varepsilon_2|) - \frac{1}{2} \alpha^2 \varepsilon_1 \tag{B1.10}$$

where

$$\varepsilon_1 = \mathbf{p}_i \mathbf{G}^{(i)-1} \mathbf{p}_i^T \tag{B1.11}$$

and

$$\varepsilon_2 = \mathbf{p}_i \mathbf{G}^{(i)-1} \mathbf{k}^{(i)T} \tag{B1.12}$$

and using the two maximum values of α,

$$\mathcal{Q}^{(i)}(\mathcal{M}, \hat{\mathcal{M}}) = \beta \log\left(\left| \frac{\varepsilon_2 \pm \sqrt{(\varepsilon_2^2 + 4\varepsilon_1 \beta)}}{2} \right| \right) - \frac{\varepsilon_1}{2} \left(\frac{-\varepsilon_2 \pm \sqrt{(\varepsilon_2^2 + 4\varepsilon_1 \beta)}}{2\varepsilon_1} \right)^2. \tag{B1.13}$$

As it is not possible to ensure that $\varepsilon_2 > 0$, the value of α is selected that maximizes $\mathcal{Q}^{(i)}(\mathcal{M}, \hat{\mathcal{M}})$.

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

Appendix B.2: Indirect method over rows

Using the optimization 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}.

$$\frac{\partial \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})}{\partial w_{ij}} = \beta \frac{\mathbf{p}_{ij}}{\mathbf{p}_i \mathbf{w}_i^T} - \mathbf{w}_i \mathbf{g}_j^{(i)T} + k_j^{(i)}. \tag{B2.1}$$

Equating this expression to one and rearranging into the form

$$\varepsilon_1 w_{ij}^2 - \varepsilon_2 w_{ij} - \varepsilon_3 = 0 \tag{B2.2}$$

where

$$\varepsilon_1 = p_{ij} g_{jj}^{(i)}$$

$$\varepsilon_2 = p_{ij}\left(k_j^{(i)} - \sum_{l \neq j} w_{il} \mathbf{g}_{lj}^{(i)}\right) - |\mathbf{A}|^{(j)} g_{jj}^{(i)}$$

$$\varepsilon_3 = \beta p_{ij} + |\mathbf{A}|^{(j)}\left(k_j^{(i)} - \sum_{l \neq j} w_{il} g_{lj}^{(i)}\right)$$

and

$$|\mathbf{A}|^{(j)} = \sum_{l \neq j} w_{il} p_{il}. \tag{B2.3}$$

Solving this is a standard problem, thus

$$w_{ij} = \frac{\varepsilon_2 \pm \sqrt{\varepsilon_2^2 + 4\varepsilon_1 \varepsilon_3}}{2\varepsilon_1}. \tag{B2.4}$$

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.