SEMI-TIED FULL-COVARIANCE MATRICES FOR HIDDEN MARKOV MODELS

M.J.F. Gales

CUED/F-INFENG/TR 287

April 1997

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

Email: mjfg@eng.cam.ac.uk

Abstract

There is normally a simple choice made in the form of the covariance matrix to be used with HMMs. Either a diagonal covariance matrix is used, with the underlying assumption that elements of the feature vector are independent, or a full or block-diagonal matrix is used, where all or some of the correlations are explicitly modelled. Unfortunately when using full or block-diagonal covariance matrices there tends to be a dramatic increase in the number of parameters per Gaussian component, limiting the number of components which may be robustly estimated. This paper introduces a new form of covariance matrix which allows a few "full" covariance matrices to be shared over many distributions, whilst each distribution maintains its own "diagonal" covariance matrix. In contrast to other schemes which have hypothesised a similar form, this technique fits within the standard maximum-likelihood criterion used for training HMMs. The new form of covariance matrix is evaluated on a large-vocabulary speech-recognition task. In initial experiments the performance of the standard system was achieved using approximately half the number of parameters. Moreover, a 10% reduction in word error rate compared to a standard system can be achieved with less than a 1% increase in the number of parameters and little increase in recognition time.

1 Introduction

There is normally a simple choice made in the form of the covariance matrix to be used with hidden Markov models (HMMs) [17]. Either a diagonal covariance matrix is used, with the underlying assumption that each element of the feature vector is independent, or a full or block-diagonal matrix is used, where all or some of the correlations are explicitly modelled. Unfortunately when using full or block-diagonal covariance matrices there tends to be a dramatic increase in the number of parameters per Gaussian component, limiting the number of components which may be robustly estimated. To overcome this problem multiple diagonal-covariance Gaussian distributions may be used [13, 10]. In addition to being able to model non-Gaussian distributions they can model correlations. However, it is preferable to decorrelate the feature as far as possible, as components must be used to model correlations rather than the possible non-Gaussian nature of the density function associated with a particular state.

There have been many attempts to overcome this problem. They may be basically split into two areas, feature-space and model-space schemes. In feature-space schemes, the front-end processing is modified to try and ensure that all elements of the feature vector are independent. These include schemes such as linear-discriminant analysis and the Karhunen-Loève transform [4]. In speech recognition the use of the discrete cosine transform is common for this reason [2]. However, it is hard to find a single transform which decorrelates all elements of the feature vector for all states. Model-space approaches allow many decorrelating transforms to be used. A different transform is selected depending on which component the observation was hypothesised to be generated from. In the limit a transform may be used for each component, which is equivalent to a full-covariance matrix system. The scheme which is most closely related to the one which will be described here is the state-specific rotation [14], which normally uses a separate transform for each state, but may be applied at any level of clustering.

The model-space transform introduced in this paper is a natural extension of the state-specific rotation. Instead of estimating the transform independently of the specific components associated with it, the transform is estimated in a maximum-likelihood (ML) fashion given the current model parameters. Recently an extension to linear discriminant analysis based on ML has been proposed [11]. Though addressing a different problem, this results in a similar optimisation task to the one described here, but limited to having a single, global, transform. In contrast to the scheme presented here, where an iterative scheme which is guaranteed to increase the likelihood is proposed, numerical techniques or steepest descent are used in estimating the transform. With a simple modification the optimisation scheme presented here may also be used to obtain the linear discriminant transform for the diagonal-covariance-matrix case. The two approaches can be combined so that each transformation selects a particular feature sub-space, rather than just a linear transformation of the feature space.

The next section describes the state-specific transform. Semi-tied full-covariance matrices are then introduced and re-estimation formulae, which are guaranteed to increase the likelihood of the training data, are detailed. Various implementation issues, such as the memory requirements, how the component to transform clustering may be performed and numerical accuracy issues are discussed. The use of standard linear model-space adaptation schemes in conjunction with the semi-tied full-covariance matrices is then described. The new technique is evaluated on a large-vocabulary speech recognition task.

2 State-Specific Rotations

In HMM-based systems there is a basic choice in the form of covariance matrix to be used. It may either be diagonal, block-diagonal, or full. The full covariance matrix case has the advantage over the diagonal case in that it models inter feature-vector element correlation. However this is at the cost of a greatly increased number of parameters, $n(n+3)/2$ compared to $2n$ per component. Due to this massive increase in the number of parameters, diagonal covariance matrices are commonly used in large-vocabulary speech recognition. The hope is that by using multiple components any strong correlations may be implicitly modelled.

One scheme proposed for handling this problem is to use state-specific rotations [14]. Here a full covariance matrix is calculated for each state in the system. This is then decomposed into its eigenvectors and eigenvalues. All data from that state is then decorrelated using the eigenvectors and multiple diagonal covariance matrix components may then be trained. Thus the covariance matrix associated with each state, s , is decomposed as

$$\mathbf{\Sigma}_{\text{full}}^{(s)} = \mathbf{U}^{(s)}\mathbf{\Lambda}^{(s)}\mathbf{U}^{(s)T} \tag{1}$$

where

$$\mathbf{\Sigma}_{\text{full}}^{(s)} = \frac{\sum_{\tau=1}^{T}\gamma_s(\tau)(\mathbf{o}(\tau)-\mu^{(s)})(\mathbf{o}(\tau)-\mu^{(s)})^T}{\sum_{\tau=1}^{T}\gamma_s(\tau)} \tag{2}$$

$\mathbf{\Lambda}^{(s)}$ is the diagonal matrix of the eigenvalues, $\mu^{(s)}$ is the state mean and

$$\gamma_s(\tau) = p(q_s(\tau)|\mathcal{M}, \mathbf{O}_T) \tag{3}$$

where $q_s(\tau)$ indicates state (or component) s at time τ and $\mathbf{O}_T$ is the complete set of training data. When training, instead of using the standard observation vector, $\mathbf{o}(\tau)$, a state specific observation vector, $\mathbf{o}^{(s)}(\tau)$, is used where

$$\mathbf{o}^{(s)}(\tau) = \mathbf{U}^{(s)T}\mathbf{o}(\tau) \tag{4}$$

Each component, m, associated with that particular state, s, is then trained using

$$\mu^{(sm)} = \frac{\sum_{\tau=1}^{T}\gamma_m(\tau)\mathbf{o}^{(s)}(\tau)}{\sum_{\tau=1}^{T}\gamma_m(\tau)} \tag{5}$$

(note that $\mu^{(sm)} = \mathbf{U}^{(s)T}\mu^{(m)}$) and

$$\mathbf{\Sigma}_{\text{diag}}^{(m)} = \text{diag}\left(\frac{\sum_{\tau=1}^{T}\gamma_m(\tau)\left(\mathbf{o}^{(s)}(\tau)-\mu^{(sm)}\right)\left(\mathbf{o}^{(s)}(\tau)-\mu^{(sm)}\right)^T}{\sum_{\tau=1}^{T}\gamma_m(\tau)}\right) \tag{6}$$

where diag(.) just extracts the leading diagonal. The covariance matrix associated with each component is

$$\mathbf{\Sigma}^{(m)} = \mathbf{U}^{(s)}\mathbf{\Sigma}_{\text{diag}}^{(m)}\mathbf{U}^{(s)T} \tag{7}$$

During recognition and training the likelihood used for component m of state s is

$$\mathcal{L}\left(\mathbf{o}(\tau);\mu^{(m)},\mathbf{\Sigma}^{(m)},\mathbf{U}^{(s)}\right) = \mathcal{N}\left(\mathbf{o}^{(s)}(\tau);\mu^{(sm)},\mathbf{\Sigma}_{\text{diag}}^{(m)}\right) \tag{8}$$

Computationally this is relatively efficient as it is only necessary to perform one rotation per state, in contrast to standard full covariance matrices which are the equivalent of one rotation per component.

Although this does partially handle the problem of correlation in feature vectors it does not fit within the standard ML estimation approach for training HMMs. The transforms are not related to the multiple-component models being used for recognition. One simple extension is to use the average within-component covariance per state, as opposed to the global state covariance. Thus the same transform is used except that equation 2 is replaced by

$$\mathbf{\Sigma}_{\text{full}}^{(s)} = \frac{\sum_{m=1}^{M^{(s)}}\sum_{\tau=1}^{T}\gamma_m(\tau)(\mathbf{o}(\tau)-\mu^{(m)})(\mathbf{o}(\tau)-\mu^{(m)})^T}{\sum_{m=1}^{M^{(s)}}\sum_{\tau=1}^{T}\gamma_m(\tau)} \tag{9}$$

where $\mu^{(m)}$ is the current estimate of the component mean. This still does not yield a transform that is guaranteed to increase the likelihood (it uses the same sort of approximation as least-squares linear regression [8]), but does relate the transform to the current model set.

A further modification can be used to achieve a transform that is guaranteed to increase the likelihood. This transform has a similar form to the variance transform described in [6]. The component-specific variance may be written as

$$\mathbf{\Sigma}^{(m)} = \mathbf{L}_{\text{diag}}^{(m)} \mathbf{\Sigma}_{\text{full}}^{(s)\prime} \mathbf{L}_{\text{diag}}^{(m)T} \tag{10}$$

where

$$\mathbf{\Sigma}_{\text{full}}^{(s)\prime} = \frac{\sum_{m=1}^{M^{(s)}} \mathbf{L}_{\text{diag}}^{(m)-1} \left\{ \sum_{\tau=1}^{T} \gamma_m(\tau) (\mathbf{o}(\tau) - \mu^{(m)})(\mathbf{o}(\tau) - \mu^{(m)})^T \right\} \mathbf{L}_{\text{diag}}^{(m)-1}}{\sum_{m=1}^{M^{(s)}} \sum_{\tau=1}^{T} \gamma_m(\tau)} \tag{11}$$

and the diagonal matrix $\mathbf{L}_{\text{diag}}^{(m)T}$ is defined as

$$\mathbf{\Sigma}_{\text{diag}}^{(m)} = \mathbf{L}_{\text{diag}}^{(m)T} \mathbf{L}_{\text{diag}}^{(m)T} \tag{12}$$

This allows a full-covariance element to be shared over many components. Unfortunately there is a significant increase in the computational load during recognition [6].

This paper introduces a natural extension to the state-specific rotation approach. The transforms are trained in a ML sense, whilst maintaining the low recognition-time cost of the state-specific rotation.

3 Semi-Tied Full-Covariance Matrices

Semi-tied full covariance matrices extend the concept of state-specific rotations, so that the transformation are trained in a ML fashion given the current model set, rather than being obtained from a rotation derived independently of the current model set.

Consider the covariance matrix

$$\hat{\mathbf{\Sigma}}^{(m)} = \mathbf{A}^{(r)\prime} \mathbf{\Sigma}_{\text{diag}}^{(m)} \mathbf{A}^{(r)\prime T} \tag{13}$$

where $\mathbf{\Sigma}_{\text{diag}}^{(m)}$ is again a diagonal covariance matrix and $\mathbf{A}^{(r)\prime}$ is the full or block diagonal transform of the covariance matrix. This is very similar to covariance matrix form described in equation 1, other than the implicit constraint in the state-specific rotation, resulting from the eigenvector decomposition, that the transform is orthonormal. $\mathbf{A}^{(r)\prime}$ may be tied over a set of components, say all those associated with the same state of a particular context-independent phone, the, and $\mathbf{\Sigma}_{\text{diag}}^{(m)}$ is component specific. Rather than optimising $\mathbf{A}^{(r)\prime}$, the inverse, $\mathbf{A}^{(r)}$, is optimised, thus

$$\mathbf{A}^{(r)} = \mathbf{A}^{(r)\prime -1} \tag{14}$$

It is very complex to optimise this directly so an expectation-maximisation approach is adopted [3]. To train the transform the following auxiliary function is used[1]

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = K - \tag{15}$$
$$\frac{1}{2} \sum_{m=1}^{M} \sum_{\tau=1}^{T} \gamma_m(\tau) \left[K^{(m)} + \log(|\mathbf{\Sigma}_{\text{diag}}^{(m)}|) - 2\log(|\mathbf{A}|) + \left(\mathbf{A}\hat{\mathbf{o}}^{(m)}(\tau) \right)^T \mathbf{\Sigma}_{\text{diag}}^{(m)-1} \left(\mathbf{A}\hat{\mathbf{o}}^{(m)}(\tau) \right) \right]$$

where K is a constant independent of the model mean and variance, $K^{(m)}$ is the standard normalisation factor associated with component m and

$$\hat{\mathbf{o}}^{(m)}(\tau) = \mathbf{o}(\tau) - \mu^{(m)} \tag{16}$$

[1] The superscript $^{(r)}$ has been dropped for ease of notation. Hence M here represents the total number of components associated with the transformation class r.

Differentiating with respect to $\mathbf{A}$ yields

$$(-)\frac{\partial \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})}{\partial \mathbf{A}} = \sum_{m=1}^{M} \sum_{\tau=1}^{T} \gamma_m(\tau) \left[-\mathbf{A}^{T-1} + \boldsymbol{\Sigma}_{\text{diag}}^{(m)-1} \mathbf{A} \hat{\mathbf{o}}^{(m)}(\tau) \hat{\mathbf{o}}^{(m)}(\tau)^T \right] \tag{17}$$

Unfortunately this does not yield a simple closed-form solution to the problem, as equation 17 is non-linear. However it is possible to describe a simple iterative scheme which is guaranteed at each iteration to increase the likelihood of the adaptation data. Details of the optimisation are given in appendix A. For estimating the component specific model parameters equations similar to 5 and 6 are used.

During recognition the likelihood is based on[2]

$$\mathcal{L}\left(\mathbf{o}(\tau); \mu^{(m)}, \boldsymbol{\Sigma}^{(m)}, \mathbf{A}^{(r)}\right) = \mathcal{N}\left(\mathbf{o}^{(r)}(\tau); \mathbf{A}^{(r)}\mu^{(m)}, \boldsymbol{\Sigma}_{\text{diag}}^{(m)}\right) + \log\left(|\mathbf{A}^{(r)}|\right) \tag{18}$$

and

$$\mathbf{o}^{(r)}(\tau) = \mathbf{A}^{(r)}\mathbf{o}(\tau) \tag{19}$$

Thus by storing $\mathbf{A}^{(r)}\mu^{(m)}$ instead of $\mu^{(m)}$ the cost of calculating the likelihoods associated with semi-tied full covariance matrices is that of one matrix vector multiplication per-transform class and an addition[3].

It is worth emphasising the difference between semi-tied full-covariance matrices and state-based rotations. The most important difference is that the semi-tied full-covariance matrices are trained in a ML sense from the training data given the current model set. It would only be possible to train a state-based rotation in a ML sense when all the values of $\boldsymbol{\Sigma}_{\text{diag}}^{(m)}$ associated with a particular transform are the same[4]. Typically this constraint is not satisfied. As the number of transforms decreases, so the differences between the component specific variances associated with a particular transform becomes larger, and the difference between the state-specific rotation and ML estimated rotation becomes increasingly large. Furthermore, there are no constraints on the form of the matrix transform in semi-tied full-covariance matrices. In the state-specific rotations the transforms are constrained to be orthonormal, as they are derived from the eigenvectors of the state covariance matrix.

4 Implementation Issues

4.1 Statistics required

An important issue in the practical implementation of estimating the transform is the statistics required. If implemented directly it is necessary to store $\mathcal{O}(n^2)$, where n is the dimension of the feature vector, parameters per component. This can very rapidly become expensive in terms of memory as the number of components increases. Alternatively, it is possible to re-express all the expressions in appendix A in terms of

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

where i indicates one of the n dimensions. Assuming that a full transformation matrix is to be estimated then it is only necessary to store $\mathcal{O}(n^3)$ parameters per transform. If block-diagonal

[2]The last term, $\log(|\mathbf{A}|)$, should strictly be written as $\frac{1}{2}\log\left(|\mathbf{A}|^2\right)$, thus allowing the determinant of $\mathbf{A}$ to go negative.

[3]If only one transformation class is used then $\log(|\mathbf{A}|)$ does not discriminate between the models so may be ignored. It is also possible to eliminate the requirements for the determinant term by simply scaling the elements of the diagonal covariance matrix [11].

[4]This effectively states that all component variances associated with the same state are tied.

transforms are to be used, for example as used in the experiments here where a separate transform is used for each of the static, delta and delta-delta parameters, the storage requirements is reduced as $\frac{1}{b^2}\mathcal{O}(n^3)$ where b is the number of blocks. The actual choice of where to store the statistics depends on the number and nature of the transforms and the number of components.

In addition to just updating the transform part of the covariance matrix, it is possible to store the standard statistics to update the mean. This allows the means to be updated at the same time as the semi-tied full-covariance matrix transform. If the statistics are stored at the component level, rather than at the transform level, it is also possible to update the component-specific diagonal-elements of the covariance matrix. If all the parameters are updated simultaneously according to the current alignment then there is no unique solution in common with the ML linear discriminant analysis training [11]. One way around this problem, which is also applicable to the ML trained linear discriminant analysis, is to iteratively optimise the mean and transform, $\mathbf{A}$, then the component specific diagonal covariance matrix. Each iteration is guaranteed to increase the likelihood of the training data. This may be performed using the scheme described in appendix A. However, this is not considered in this work due to the memory requirements that would result from dealing with large model sets.

4.2 Number of iterations required

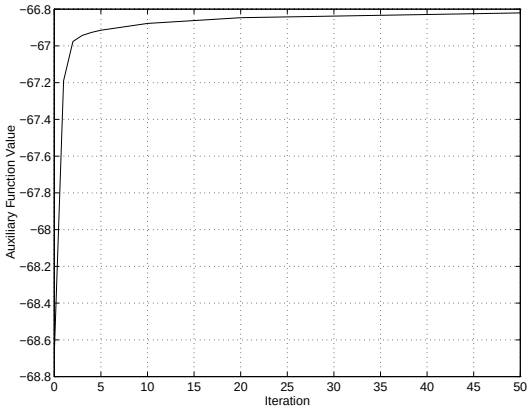

Figure 1: Convergence rate for estimating the transformation matrix of the semi-tied full-covariance matrix.

One issue is how long the estimation task takes. For the semi-tied full-covariance matrix the estimation process is an iterative one. Figure 1 shows a typical change in auxiliary function value against iteration number. In many applications, particularly large-vocabulary speech-recognition tasks, the majority of the training time is spent obtaining the statistics from the training data, rather than estimating the model parameters given those statistics. Thus the actual parameter estimation time is not crucial. For all the cases considered here the estimation of the transformation matrix was run to convergence.

4.3 Transformation parameter tying

A variety of techniques may be used for clustering components, for example decision tree tying [1]. Unfortunately, it is harder to decide how to group the components when using semi-tied full-covariance matrices. The simplest approach is to tie all states of the same monophone together. The clustering may also be determined by generating a full covariance matrix single component system and performing agglomerative clustering.

An alternative scheme that has previously been used for generating regression class trees is based on locally maximising the likelihood [5]. Here a modified version of K-means clustering is

used. If there are currently R transforms then for each state[5], s, the transform associated with that state, $\hat{r}^{(s)}$, is determined by

$$\hat{r}^{(s)} = \arg\max_{r \in R} \left\{ \sum_{m=1}^{M^{(s)}} \sum_{\tau=1}^{T} \gamma_m(\tau) \left[2\log(|\mathbf{A}^{(r)}|) - \left(\mathbf{A}^{(r)} \hat{\mathbf{o}}^{(m)}(\tau) \right)^T \boldsymbol{\Sigma}^{(m)-1} \left(\mathbf{A}^{(r)} \hat{\mathbf{o}}^{(m)}(\tau) \right) \right] \right\} \quad (21)$$

This is guaranteed to generate a local maximum of assignment. One problem that has been observed with this form of optimisation is the dependency of the clustering on the start position.

For the experiments presented in this paper a single transform is used for each phone class and no reassignment of component or state to transform undertaken.

4.4 Numerical accuracy

When calculating the transform there is a danger of the statistics stored not having full rank. This may be due to numerical inaccuracies or a limited amount of training data. Defining $\mathbf{G}^{(i)}$ as in equation 20, the final term of the calculation of the probability may be shown to contain the term

$$\sum_{m=1}^{M} \sum_{\tau=1}^{T} \gamma_m(\tau) \left(\mathbf{A}^{(r)} \hat{\mathbf{o}}^{(m)}(\tau) \right)^T \boldsymbol{\Sigma}_{\text{diag}}^{(m)-1} \left(\mathbf{A}^{(r)} \hat{\mathbf{o}}^{(m)}(\tau) \right) = \sum_{i=1}^{n} \mathbf{a}_i^{(r)} \mathbf{G}^{(i)} \mathbf{a}_i^{(r)T} \quad (22)$$

It is simple to see that when $\mathbf{G}^{(i)}$ does not have full rank then the elements of $\mathbf{a}_i$, row i of $\mathbf{A}$, in the null space may be set arbitrarily large (this will increase the likelihood as it increases $|\mathbf{A}|$, but does not alter the value of equation 22). There are two solutions to this problem, similar to the calculation of the transforms in maximum likelihood linear regression (MLLR) [12]. The first is to use block diagonal transformations, thus dramatically reducing the chance of non-full rank matrices. Alternatively SVD may again be used. $\mathbf{G}^{(i)}$ may be rewritten as

$$\mathbf{G}^{(i)} = \mathbf{U}^{(i)} \boldsymbol{\Lambda}^{(i)} \mathbf{U}^{(i)T} \quad (23)$$

and

$$\hat{\mathbf{a}}_i^T = \mathbf{U}^{(i)T} \mathbf{a}_i^T \quad (24)$$

By optimising $\hat{\mathbf{a}}_i$ in only those dimensions where there are no numerical accuracy problems, the standard re-estimation formulae for the constrained case may be used. Further details of this optimisation are given in appendix B.

4.5 Non-diagonal component-specific matrices

There are situations where the elements of the component-specific matrix are non-diagonal. The most obvious situation would be where stochastic segment models [16] were to be used. Here the covariance matrix may be split into two terms, a within-frame correlation and a between-frame correlation. Thus the within-frame correlation may be modelled by a component-specific full-covariance matrix and the between-frame elements on say a phone-specific semi-tied full-covariance matrix. The optimisation in appendix A assumes that the component-specific matrix is diagonal. Standard gradient descent schemes could be used, though this requires storage of statistics at the component level and may be computationally expensive. Alternatively, this problem may be overcome by using a modified version of *normalised domain* MLLR [7]. Here the within-frame full covariance matrices, $\boldsymbol{\Sigma}_w^{(m)}$, are decomposed as

$$\boldsymbol{\Sigma}_w^{(m)} = \mathbf{U}^{(m)} \boldsymbol{\Sigma}_{\text{diag}}^{(m)} \mathbf{U}^{(m)T} \quad (25)$$

The semi-tied full-covariance matrices are then obtained in this normalised domain, where only $\boldsymbol{\Sigma}_{\text{diag}}^{(m)}$ needs to be considered. This approach may also be used for adaptation and is detailed in the next section.

[5]This assumes that all components associated with a particular state will use the same transformation.

5 Speaker and Environmental Adaptation

There is the question of how the model-based linear transformation schemes, currently popular in speech recognition, may be applied to situations where semi-tied full covariance matrices are used. MLLR is not usually applied to models with full-covariance matrices as it is computationally too expensive [6]. The same problem applies when semi-tied full covariance are used. However, instead of applying standard adaptation schemes, a modified version of normalised domain MLLR [7] may be used. The original normalised domain MLLR mapped all the covariance matrices to the identity matrix and calculated the transform in this new simplified domain (this is the same computational cost in estimating the transforms as least squares linear regression [8]). The transform was then mapped back into the original domain. This generates a transform which is guaranteed to increase the likelihood of the adaptation data. However, the scheme was found to be sensitive in certain cases of unobserved components. Instead of converting the covariance matrices to the identity matrix, the matrix may be decomposed into its eigenvalues and eigenvectors[6]

$$\boldsymbol{\Sigma}^{(m)} = \mathbf{U}^{(m)}\boldsymbol{\Sigma}^{(m)}_{\text{diag}}\mathbf{U}^{(m)T} \tag{27}$$

By expressing the matrix in this format the optimisation for MLLR becomes

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = K - \tag{28}$$
$$\frac{1}{2}\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[K^{(m)} + \log(|\boldsymbol{\Sigma}^{(m)}|) + (\mathbf{o}(\tau) - \mu^{(m)})^T\mathbf{U}^{(m)}\boldsymbol{\Sigma}^{(m)-1}_{\text{diag}}\mathbf{U}^{(m)T}(\mathbf{o}(\tau) - \mu^{(m)})\right]$$

This may be optimised using the standard MLLR equations [12] and the final transform is

$$\hat{\mu}^{(m)} = \mathbf{U}^{(m)}\mathbf{A}\mathbf{U}^{(m)T}\mu^{(m)} + \mathbf{U}^{(m)}\mathbf{b} \tag{29}$$

where $\mathbf{A}$ and $\mathbf{b}$ are the results of optimising equation 28. This allows models with component-specific full-covariance matrices to be compensated in approximately the same cost at standard MLLR, whilst avoiding the problems of normalised-domain MLLR described in [7].

This adaptation may be contrasted with the least squares linear regression (LSLR) implemented in [9] when the decorrelating rotation described in [14] was used. Using LSLR it is not possible to guarantee that the likelihood of the adaptation data will increase.

6 Results

An initial investigation of the use of semi-tied full-covariance matrices was carried out on a large-vocabulary speaker-independent continuous-speech recognition task. All recognition experiments were carried out on the 1994 ARPA Hub 1 data. The H1 task is an unlimited vocabulary task with approximately 15 sentences per speaker. The data was recorded in a clean[7] environment. No speaker adaptation was performed.

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 [18]. 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.

[6] An alternative decomposition is to use

$$\boldsymbol{\Sigma}^{(m)} = \mathbf{A}^{(r)\prime}\boldsymbol{\Sigma}^{(m)}_{\text{diag}}\mathbf{A}^{(r)\prime T} \tag{26}$$

This is computationally cheaper as the SVD does not need to be performed for each component (though of course $\mathbf{A}^{(r)}$ must be inverted.

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

This yielded a 39-dimensional feature vector, to which cepstral mean normalisation was applied. 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 [20] to define 6399 speech states. For the H1 task a 65k word list and dictionary was used with the trigram language model described in [18]. All decoding used a dynamic-network decoder [15].

When generating multiple component systems *mixing-up* was used [19]. The performance was investigated at various stages of this process. It should be emphasised that the grammar scale factor and insertion penalties were not optimised at any stage for the particular number of components (or for the use semi-tied covariance matrices). For the particular implementation of semi-tied full-covariance matrices considered here all states of all context-dependent phones associated with the same monophone were tied. Furthermore, a simple block-diagonal transformation was used. This resulted in very few additional parameters, 23322, in a system of up to 6 million parameters. The process of building the semi-tied full-covariance matrices was to firstly mix-up to the new number of components. Two iterations of Baum-Welch re-estimation were performed. The new transform was then estimated and finally an additional two iterations of Baum-Welch re-estimation run. This process was repeated as necessary.

Number Speech Components	Semi-Tied Covariance	Distribution Parameters	Error Rate (%)	
			H1 Dev	H1 Eval
1	—	501018	13.93	15.54
	Block	(+23322)	12.27	13.70
2	—	1012938	12.01	13.04
	Block	(+23322)	11.06	11.81
4	—	2023980	10.56	11.43
	Block	(+23322)	9.86	9.65
6	—	3036918	10.08	10.91
	Block	(+23322)	9.17	9.30
8	—	4049224	9.67	9.97
	Block	(+23322)	8.88	8.61
10	—	5061530	9.42	9.51
	Block	(+23322)	8.46	8.38
12	—	6073836	9.57	9.20
	Block	(+23322)	8.62	8.12

Table 1: Semi-tied covariance matrices results on H1 development and evaluation data

The first thing to notice about table 1 is that despite the very small increase in the number of parameters, even for the single component case only an additional 5%, the effects on the recognition performance is quite dramatic. For the single component case on the evaluation data the block transform reduced the error rate by 12%. For all cases the semi-tied full covariance matrix case gives a gain over the standard covariance matrix; the performance of the standard 12-component system was achieved using only 6 components. In the 12-component case, a 12% reduction in word error rate was achieved, which is comparable with the performance achieved *with* incremental speaker adaptation for the standard system [6].

The performance figures in table 1 may be compared to the state-based rotation scheme [14]. For the implementation examined here separate transforms were calculated for each state of each monophone using equation 2. On the H1 evaluation task this system had a word error rate of 14.25% for the single component system and 12.85% on the two component system. Though this shows an slight improvements over the standard system, the gain is considerably less than that achieved with semi-tied full-covariance matrices described here. Furthermore the gains over the standard became negligible as the number of components increased. It should be emphasised that

the "state-specific" rotations estimated were far fewer than those in [14], which possibly explains the poor performance. However, there are still more transforms than the semi-tied full-covariance matrix case.

The number of Baum-Welch re-estimations after the transform has been learnt was set very low: only two. This is not expected to seriously effect the mixing-up process, but for a particular number of components it may give poorer recognition performance than is actually possible. Thus purely for recognition purposes an additional two iterations of Baum-Welch re-estimation were performed[8]. For the six component system this gave a slight increase in performance, 9.03% word error rate on the development data and 9.28% on the evaluation data.

An alternative method of using semi-tied full-covariance matrices is to take a fully trained system and then train the transformation part of the covariance matrix. After obtaining the transform additional iterations of Baum-Welch may then be applied. Using this approach on the 12 component system and performing two additional iterations of Baum-Welch gave 9.00% on the development task and 8.59% on the evaluation task. These again show improvements compared to the standard system, though not as large as incorporating training the transforms into the mixing up process. This indicates how the transforms may be incorporated when "mixing-up" is not used in the standard training procedure. Of course further improvements could be obtained by generating new components, typically using K-means clustering [9], given the current set of transforms.

7 Conclusions

This paper has introduced a new form of covariance matrix, the semi-tied full-covariance matrix. Using this new form of matrix, it is possible to choose a compromise between the large number of parameters of the full-covariance matrix and the poor modelling ability of the diagonal case. Maximum likelihood re-estimation formulae are derived, which are guaranteed to increase the likelihood of the training data. How this new form of covariance matrix may be used with standard model-based adaptation schemes is also described. The new models were tested on a large-vocabulary speech recognition task where a reduction in word error rate of 10% over the standard system was achieved with little increase in the number of parameters or computational cost.

Future work will involve experiments using the proposed linear adaptation scheme with semi-tied full-covariance matrices and use of other transformation groupings.

Acknowledgements

Mark Gales is funded as a Research Fellow at Emmanuel College, Cambridge.

A Semi-Tied Full-Covariance Optimisation

The objective is to find the values of $\mathbf{A}$ that maximises the following expression

$$\begin{aligned}\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) = K_1 - \frac{1}{2}\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\Big[K^{(m)} \\ + \log(|\boldsymbol{\Sigma}^{(m)}|) - 2\log(|\mathbf{A}|) + \left(\mathbf{A}\hat{\mathbf{o}}^{(m)}(\tau)\right)^T \boldsymbol{\Sigma}_{\text{diag}}^{(m)-1}\left(\mathbf{A}\hat{\mathbf{o}}^{(m)}(\tau)\right)\Big]\end{aligned} \tag{30}$$

Differentiating with respect to $\mathbf{A}$ yields

$$(-)\frac{\partial\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}})}{\partial\mathbf{A}} = \sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[-\mathbf{A}^{T-1} + \boldsymbol{\Sigma}_{\text{diag}}^{(m)-1}\left[\mathbf{A}\hat{\mathbf{o}}^{(m)}(\tau)\right]\hat{\mathbf{o}}^{(m)T}\right] \tag{31}$$

[8]The reason for only performing the additional iterations for recognition is to make the standard and semi-tied systems as comparable as possible.

Noting that

$$(a^{T-1})_{ij} = \frac{\text{cof}(\mathbf{A}_{ij})}{\sum_{k=1}^{n} a_{ik}\text{cof}(\mathbf{A}_{ik})} \tag{32}$$

where $\text{cof}(\mathbf{A}_{ij})$ is the cofactor of element a_{ij} and assuming that the determinant of $\mathbf{A}$ is non-zero

$$(-)|\mathbf{A}|\frac{\partial \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})}{\partial a_{ij}} = \sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[-\text{cof}(\mathbf{A}_{ij}) + \frac{1}{\sigma_i^{(m)2}}\left(\sum_{k=1}^{n} a_{ik}\hat{o}_k^{(m)}(\tau)\hat{o}_j^{(m)}(\tau)\right)\left(\sum_{k=1}^{n} a_{ik}\text{cof}(\mathbf{A}_{ik})\right)\right] \tag{33}$$

where $\sigma_i^{(m)2}$ is the i^{th} leading diagonal element of $\mathbf{\Sigma}_{\text{diag}}^{(m)}$. Equating this expression to zero and re-arranging into the form

$$c_1 a_{ij}^2 - c_2 a_{ij} - c_3 = 0 \tag{34}$$

where

$$\begin{aligned}
c_1 &= \sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[\frac{1}{\sigma_i^{(m)2}}\text{cof}(\mathbf{A}_{ij})\hat{o}_j^{(m)}(\tau)^2\right] \\
c_2 &= (-)\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[\frac{1}{\sigma_i^{(m)2}}\left(\text{cof}(\mathbf{A}_{ij})\left(\sum_{k\neq j} a_{ik}\hat{o}_k^{(m)}(\tau)\hat{o}_j^{(m)}(\tau)\right) + |\mathbf{A}|^{(\bar{j})}\hat{o}_j^{(m)}(\tau)^2\right)\right] \\
c_3 &= \sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[\text{cof}(\mathbf{A}_{ij}) - \frac{1}{\sigma_i^{(m)2}}\left(|\mathbf{A}|^{(\bar{j})}\sum_{k\neq j} a_{ik}\hat{o}_k^{(m)}(\tau)\hat{o}_j^{(m)}(\tau)\right)\right]
\end{aligned}$$

and

$$|\mathbf{A}|^{(\bar{j})} = \sum_{k\neq j} a_{ik}\text{cof}(\mathbf{A}_{ik}) \tag{35}$$

Solving this is a standard problem, thus

$$a_{ij} = \frac{c_2 \pm \sqrt{c_2^2 + 4c_1c_2}}{2c_1} \tag{36}$$

There are two solutions, so there is the question of which root is to be chosen. Differentiating again

$$(-)\frac{\partial}{\partial a_{ij}}\left(\frac{\partial \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})}{\partial a_{ij}}\right) = \frac{(2c_1a_{ij} - c_2)}{|\mathbf{A}|} \tag{37}$$

and substituting equation 36 yields

$$(-)\frac{\partial}{\partial a_{ij}}\left(\frac{\partial \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})}{\partial a_{ij}}\right) = \frac{\pm\sqrt{c_2^2 + 4c_1c_2}}{|\mathbf{A}|} \tag{38}$$

If the determinants of $\mathbf{A}$ is constrained to be positive, it is clear that the positive solution is the one required. There is no strict constraint that $|\mathbf{A}|$ be positive, but in practice this was invariably the case from the identity matrix initial value used.

Thus by using the current estimate of the cofactors, each row of $\mathbf{A}$ may be optimised independently. By then iteratively running through the rows the complete transform may be optimised efficiently and robustly. Each iteration is guaranteed to increase likelihood of the training data

B Optimisation using SVD

The objective is to maximise the following expression (ignoring the terms independent of the transform $\mathbf{A}$)

$$\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) = -\frac{1}{2}\left\{\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[-2\log\left(\sum_{l=1}^{n}\hat{a}_{il}\sum_{k=1}^{n}u_{kl}^{(i)}\mathrm{cof}(\mathbf{A}_{ik})\right)\right] + \sum_{i=1}^{n}\sum_{l=1}^{n}\hat{a}_{il}^2\lambda_{ll}^{(i)}\right\} \tag{39}$$

where

$$\hat{\mathbf{a}}_i^T = \mathbf{U}^{(i)T}\mathbf{a}_i^T \tag{40}$$

and

$$\mathbf{G}^{(i)} = \sum_{m=1}^{M}\frac{1}{\sigma_i^{(m)2}}\sum_{\tau=1}^{T}\gamma_m(\tau)\hat{\mathbf{o}}^{(m)}(\tau)\hat{\mathbf{o}}^{(m)}(\tau)^T = \mathbf{U}^{(i)}\mathbf{\Lambda}^{(i)}\mathbf{U}^{(i)T} \tag{41}$$

$\lambda_{ll}^{(i)}$ is an element of $\mathbf{\Lambda}^{(i)}$. The dimensions whose eigenvalues are deemed to be too small to be accurate are not updated. These may be left as the elements that result from the identity matrix transform.

Differentiating this with respect to $\hat{a}_{ij}$ gives

$$\frac{\partial\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}})}{\partial\hat{a}_{ij}} = \sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[\frac{\sum_{k=1}^{n}u_{kj}^{(i)}\mathrm{cof}(\mathbf{A}_{ik})}{\left(\sum_{l=1}^{n}\hat{a}_{il}\sum_{k=1}^{n}u_{kl}^{(i)}\mathrm{cof}(\mathbf{A}_{ik})\right)}\right] - \hat{a}_{ij}\lambda_{jj}^{(i)}$$

Again assuming that the determinant is non-zero and equating to zero

$$\sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[\sum_{k=1}^{n}u_{kj}^{(i)}\mathrm{cof}(\mathbf{A}_{ik})\right] - \hat{a}_{ij}\lambda_{jj}^{(i)}\left[\sum_{l=1}^{n}\hat{a}_{il}\sum_{k=1}^{n}u_{kl}^{(i)}\mathrm{cof}(\mathbf{A}_{ik})\right] = 0 \tag{42}$$

Rearranging this into a standard quadratic yields

$$c_1\hat{a}_{ij}^2 - c_2\hat{a}_{ij} - c_3 = 0 \tag{43}$$

where

$$\begin{aligned}
c_1 &= \lambda_{jj}^{(i)}\sum_{k=1}^{n}u_{kj}^{(i)}\mathrm{cof}(\mathbf{A}_{ik}) \\
c_2 &= (-)\lambda_{jj}^{(i)}\left[\sum_{l\neq j}\hat{a}_{il}\sum_{k=1}^{n}u_{kl}^{(i)}\mathrm{cof}(\mathbf{A}_{ik})\right] \\
c_3 &= \sum_{m=1}^{M}\sum_{\tau=1}^{T}\gamma_m(\tau)\left[\sum_{k=1}^{n}u_{kj}^{(i)}\mathrm{cof}(\mathbf{A}_{ik})\right]
\end{aligned}$$

These may then be easily solved in the same way as the standard optimisation in appendix A.

References

[1] L R Bahl, P V de Souza, P S Gopalkrishnan, D Nahamoo, and M A Picheny. Context dependent modelling of phones in continuous speech using decision trees. In *Proceedings DARPA Speech and Natural Language Processing Workshop*, pages 264–270, 1991.

[2] S B Davis and P Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions ASSP*, 28:357–366, 1980.

[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] K Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1972.

[5] 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.

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

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

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

[9] D M Hindle, A Ljolje, and M D Riley. Recent improvements in the AT&T speech-to-text (STT) system. In *Proceedings ARPA Speech Recognition Workshop*, 1996.

[10] B H Juang. Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Technical Journal*, 64:1235–1249, 1985.

[11] N Kumar. *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition.* PhD thesis, John Hopkins University, (to be published) 1997.

[12] 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.

[13] L A Liporace. Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Transactions Information Theory*, 28:729–734, 1982.

[14] A Ljolje. The importance of cepstral parameters correlations in speech recognition. *Computer Speech and Language*, 8:223–232, 1994.

[15] 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.

[16] M Ostendorf and S Roukos. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Transactions ASSP*, 37:1857–1869, 1989.

[17] L R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, February 1989.

[18] 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.

[19] S J Young, J Jansen, J Odell, D Ollason, and P Woodland. *The HTK Book (for HTK Version 2.0).* Cambridge University, 1996.

[20] 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.