CAMBRIDGE UNIVERSITY

ENGINEERING DEPARTMENT

MAXIMUM LIKELIHOOD MULTIPLE PROJECTION SCHEMES FOR HIDDEN MARKOV MODELS

M.J.F. Gales[1]

CUED/F-INFENG/TR.365

October 1999

Revised June 2000

Cambridge University Engineering Department

Trumpington Street

Cambridge. CB2 1PZ

England

E-mail: `mjfg@eng.cam.ac.uk`

`http://www-svr.eng.cam.ac.uk/~mjfg`

[1]This work was performed whilst the author was with the IBM T.J. Watson Research Center speech group

Abstract

The first stage in many pattern recognition tasks is to generate a good set of features from the observed data. In many complex pattern recognition tasks the choice of a good feature space in which to model the data varies depending on the signal content, for example phone dependent subspaces in speech recognition. In these cases multiple feature subspaces should be used. Handling multiple subspaces whilst still maintaining meaningful likelihood comparisons between classes is a complex problem. This paper deals with this problem by viewing multiple subspace projections within a maximum likelihood framework. Two new multiple projection schemes are described. These new schemes are evaluated on a large vocabulary speech recognition task in terms of performance, speed of likelihood calculation and number of model parameters.

1 Introduction

The first stage in many pattern recognition tasks is to generate a good set of features from the observed data. The set should be are compact and capture all class discriminating information. Features that contain little or no information should be removed since they increase the computational load and the number of model parameters to be estimated without improving performance. Furthermore, the features generated should be suited to the form of classifier being used, for example if diagonal covariance matrices are used the data should be decorrelated. For some complex signals the "best" feature set is class dependent. In these cases multiple, class specific, feature sets may be more appropriate than a single feature set. This paper considers multiple features sets, using large vocabulary continuous speech recognition (LVCSR) as the example task. For speech, the particular acoustic realisation may be best modelled in different acoustic subspaces depending on whether, for example, a vowel or consonant was generated. LVCSR is particularly interesting in this context because of the complexity of the signal and the large number of Gaussian components used in the typical systems.

The majority of previous work has concentrated on generating a single "good" feature subspace [10, 2, 11]. Some techniques may be applied directly using just the observed speech data, for example schemes based on principal component analysis (PCA). Others require class labels, such as linear discriminant analysis (LDA) [2, 10]. Recently the use of maximum likelihood (ML) estimation has been proposed for generating a linear subspace projection[2] with class labelled, or hypothesised class labels, heteroscedastic LDA (HLDA) [11]. As ML estimation is being used distributions must be specified for all dimensions to ensure a consistent space for likelihood calculation. For those dimensions that contain class information, *useful* dimensions, class specific distributions of an appropriate form are used. Those dimensions containing little or no class information, *nuisance* dimensions, use class independent distributions.

This paper examines schemes for generating multiple linear subspaces. One important issue when using multiple subspaces is how to compare likelihoods from models built in different subspaces. This paper addresses the problem by building multiple subspace projections within the ML framework. By tying model parameters and ensuring that all the constrained transforms span all of the original space there are no problems comparing likelihoods. Since LVCSR is the target application, hidden Markov models (HMMs) [13] are used as the underlying model with Gaussian mixture components representing each state. Multiple linear subspaces have previously been examined for HMMs [14, 17]. Factor analysis (FA) [17] uses a different subspace for each component. It may be viewed as a restricted form of covariance modelling. Though ML estimation for factor analysis has simple re-estimation formulae [15], the likelihood calculations are computationally expensive for LVCSR compared to other restricted covariance modelling schemes [8]. This is further discussed in section 5. In [14] multiple components share the same subspace. The subspace transforms are trained in a discriminative fashion, rather than using ML estimation. The authors obtained improvements in recognition results. However no mention was made of ensuring that the likelihoods obtained from different subspace could be correctly compared. The normalisation terms associated with feature-space transformations were ignored, both in the optimisation schemes and likelihood calculation. For the single transform case trained using discriminative training the normalisation terms naturally cancel, but this is not the case for multiple transforms. It must be assumed therefore that the discriminative optimisation criterion is restricting the ability of the transform normalisation terms to become significantly different. Unfortunately using discriminative training schemes for LVCSR is very computationally expensive, so the vast majority of LVCSR systems, and the system considered here, are trained using ML estimation.

The proposed subspace projection schemes in this paper are extensions to HLDA [11] enabling multiple subspaces to be incorporated. Rather than describing the proposed schemes directly in the HLDA framework, they are described in terms of tying the parameters of a *semi-tied covariance matrix* (STC) [8] system. This allows a variety of schemes to be described in the same consistent fashion, including LDA and HLDA. Two new forms of tying are investigated. The first *multiple*

[2]Here the term "projection" will be used to denote reducing the size of the feature so that dimensions that contain little or no class discriminative information are removed.

HLDA is a direct extension to HLDA allowing class-dependent subspaces. In this case there is no global nuisance subspace, as all dimensions contain some class information. Hence the likelihood must be computed in the full feature space to allow valid likelihood comparisons. The second form of tying, *multiple LDA* (MLDA), generates a common nuisance subspace for all transforms. For likelihood calculations these nuisance dimensions may be ignored, reducing the computational cost.

This paper is organised as follows. The next section describes LDA and the extension to HLDA. Section 3 discusses STC systems and how they may be efficiently trained. The two subspace projection schemes are then described in terms of tying the parameters of the STC system. In the following section these models are compared to the linear Gaussian models described in [15]. Finally experiments on a speaker independent task are presented.

2 Linear Discriminant Analysis

This section describes a standard linear subspace projection scheme, linear discriminant analysis (LDA). Two equivalent forms of optimisation are described, the standard one based on the between to within class covariance ratio and an ML estimation scheme. An extension to LDA, heteroscedastic LDA is then described. This adds an additional level of flexibility to the standard LDA scheme by removing the constraint that all within class covariance matrices are approximately the same.

2.1 Standard Linear Discriminant Analysis

LDA is a standard linear projection scheme [6]. The aim of the process is to obtain a $p \times n$ projection matrix, where n is the original vector size and $p \leq n$, that results in a feature space that is "good" for discrimination. Here discrimination is measured by the ratio of between class covariance to the average within class covariance. The standard optimisation for LDA is to find the $p \times n$ transform, $\mathbf{A}_{[p]}$, that maximises [7][3]

$$\mathcal{J}(\mathbf{A}_{[p]}; \mathcal{M}) = \frac{|\det(\mathrm{diag}(\mathbf{A}_{[p]}\mathbf{B}\mathbf{A}_{[p]}^T))|}{|\det(\mathrm{diag}(\mathbf{A}_{[p]}\mathbf{W}\mathbf{A}_{[p]}^T))|} \tag{1}$$

where $\mathbf{B}$ and $\mathbf{W}$ are the between and average within class covariance matrices. These are defined as[4]

$$\mathbf{B} = \frac{\sum_{m,\tau} \gamma_m(\tau)(\check{\boldsymbol{\mu}}^{(m)} - \check{\boldsymbol{\mu}})(\check{\boldsymbol{\mu}}^{(m)} - \check{\boldsymbol{\mu}})^T}{\sum_{m,\tau} \gamma_m(\tau)} \tag{2}$$

where

$$\check{\boldsymbol{\mu}}^{(m)} = \frac{\sum_{\tau} \gamma_m(\tau)\mathbf{o}(\tau)}{\sum_{\tau} \gamma_m(\tau)} \tag{3}$$

$$\check{\boldsymbol{\mu}} = \frac{\sum_{m,\tau} \gamma_m(\tau)\mathbf{o}(\tau)}{\sum_{m,\tau} \gamma_m(\tau)} \tag{4}$$

[3]This paper uses the following convention: capital bold letters refer to matrices e.g. $\mathbf{A}$, bold letters refer to vectors e.g. $\mathbf{b}$, and scalars are not bold e.g. c. When referring to elements of a matrix or vector subscripts are used e.g. $\mathbf{a}_i$ is the i^{th} row of matrix $\mathbf{A}$, a_{ij} is the element of row i column j of matrix $\mathbf{A}$ and b_i is element i of vector $\mathbf{b}$. Diagonal matrices are indicated by $\mathbf{A}_{\mathrm{diag}}$. Where sub-matrices are used this are indicated, for example, by $\mathbf{A}_{[p]}$, this is a $p \times n$ (n is the dimensionality of the feature vector). Where subsets of the diagonal matrices are specified the matrices are square, e.g. $\mathbf{A}_{\mathrm{diag}[p]}$ is $p \times p$ square diagonal matrix. $\mathbf{A}^T$ is the transpose of the matrix and $\det(\mathbf{A})$ is the determinant of the matrix. Model parameters in the original feature space will be marked as, for example, $\check{\boldsymbol{\mu}}$.

[4]These definitions have been slightly modified to reflect the use of HMMs, or Gaussian mixture models. This is similar to the LDA process described in [2].

and

$$\mathbf{W} = \frac{\sum_{m,\tau} \gamma_m(\tau)(\mathbf{o}(\tau) - \check{\boldsymbol{\mu}}^{(m)})(\mathbf{o}(\tau) - \check{\boldsymbol{\mu}}^{(m)})^T}{\sum_{m,\tau} \gamma_m(\tau)} \tag{5}$$

In both cases $\gamma_m(\tau)$ is the posterior probability of Gaussian component m and time instance τ given the current model set, $\mathcal{M}$, and all the training data. Note that for LDA there is the implicit assumption that the within class covariance matrices for each of the Gaussian components is the same. It can be shown that the optimal value and ordering of $\mathbf{A}_{[p]}$ is found by taken the top p right eigenvectors of $\mathbf{W}^{-1}\mathbf{B}$ ordered according to decreasing eigenvalues [7].

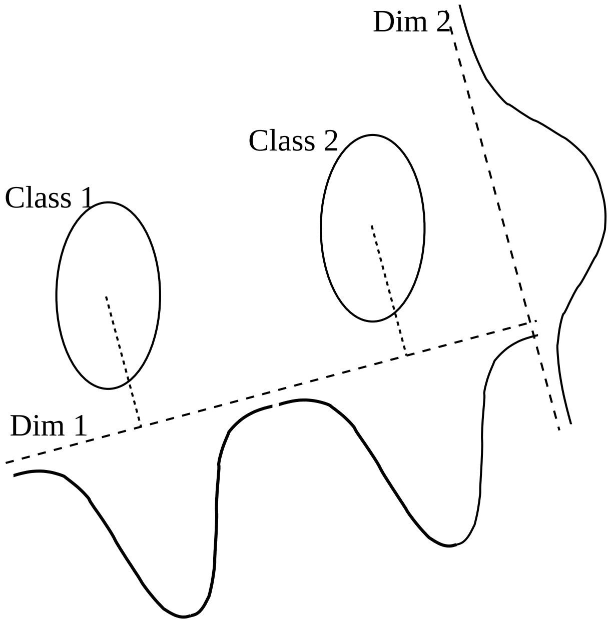

Figure 1: Linear Discriminant Analysis

Rather than expressing the optimisation in the form of equation 1 it may be cast in the form of ML estimation [3, 11]. A p-dimensional subspace of the original n-dimensional feature space is to be retained. Since ML estimation is to be used to find the transform a consistent feature space to compute the likelihoods in recognition is required. Therefore simple truncation of the transform cannot be used. Instead during training the (n-p) nuisance dimensions are modelled by a non-discriminating model, in this case a single, global, Gaussian distribution. This global distribution has mean $\check{\boldsymbol{\mu}}$, defined above, and covariance matrix $\check{\boldsymbol{\Sigma}}$, $\check{\boldsymbol{\Sigma}} = \mathbf{B}+\mathbf{W}$. A simple 2-dimensional example is shown in figure 1. There are two classes, each shown having the same within class covariance matrix. The original 2-dimensional space are projected to a single dimensional subspace ($n = 2$, $p = 1$), yielding the useful dimension, *Dim 1*. The nuisance dimension is labelled *Dim 2*. Each class has a separate distribution in *Dim 1*, and there is a single distribution in the nuisance dimension.

Optimising the likelihood, given the values of $\gamma_m(\tau)$, may be shown to require maximising

$$\begin{aligned}\mathcal{Q}_{LDA}(\mathbf{A};\mathcal{M}) \quad = \quad & \frac{1}{2}\sum_{m,\tau}\gamma_m(\tau)\log\left(\frac{|\det(\mathbf{A})|^2}{\left|\det\left(\mathrm{diag}\left(\mathbf{A}_{[p]}\mathbf{W}\mathbf{A}_{[p]}\right)\right)\right|\left|\det\left(\mathrm{diag}\left(\mathbf{A}_{[n-p]}\check{\boldsymbol{\Sigma}}\mathbf{A}_{[n-p]}\right)\right)\right|}\right) \\ & -\frac{n}{2}\sum_{m,\tau}\gamma_m(\tau)\left(\log(2\pi)+1\right)\end{aligned} \tag{6}$$

In [11] it is shown that the right eigenvectors of $\mathbf{W}^{-1}\mathbf{B}$ associated with the largest p eigenvalues are an ML solution. There are an infinite number of possible ML solutions, since there is no constraint when optimising equation 6 that $|\det(\mathbf{A})| = 1$. For example the variances can be scaled, which results in each row of the transform being scaled and the same likelihood value being obtained, though with a different transform.

2.2 Heteroscedastic LDA

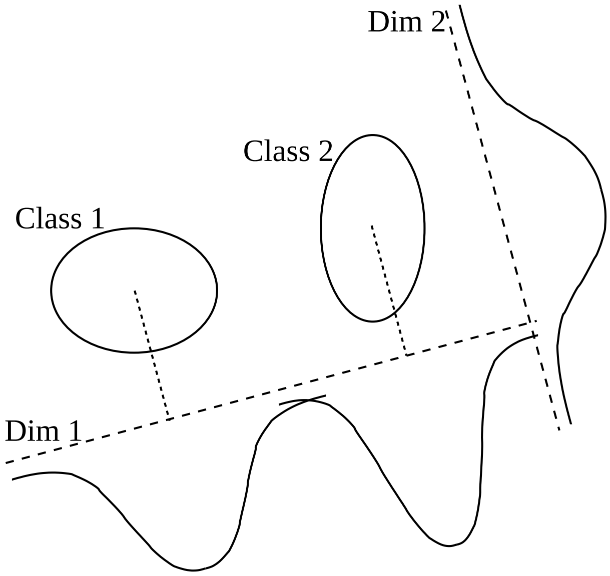

Figure 2: Heteroscedastic Linear Discriminant Analysis

Heteroscedastic LDA (HLDA) [11] is an extension to LDA in which the restriction that all within class covariance matrices are the same is removed. This is shown in figure 2. Classes 1 and 2 have different within class covariance matrices. Again the 2-dimensional data is projected into a single discriminating dimension, *Dim 1*, in which the class conditional distributions now have different covariance matrices. The nuisance dimension, *Dim 2*, has a single global Gaussian distribution associated with it.

Modifying equation 6 to reflect this added flexibility yields

$$\begin{aligned}\mathcal{Q}_{HLDA}(\mathbf{A};\mathcal{M}) &= \frac{1}{2}\sum_{m,\tau}\gamma_m(\tau)\log\left(\frac{|\det(\mathbf{A})|^2}{|\det(\mathrm{diag}(\mathbf{A}_{[p]}\mathbf{W}^{(m)}\mathbf{A}_{[p]}^T))||\det(\mathrm{diag}(\mathbf{A}_{[n-p]}\check{\boldsymbol{\Sigma}}\mathbf{A}_{[n-p]}^T))|}\right)\\ &\quad -\frac{n}{2}\sum_{m,\tau}\gamma_m(\tau)\left(\log(2\pi)+1\right) \qquad (7)\end{aligned}$$

where

$$\mathbf{W}^{(m)} = \frac{\sum_\tau \gamma_m(\tau)(\mathbf{o}(\tau)-\check{\boldsymbol{\mu}}^{(m)})(\mathbf{o}(\tau)-\check{\boldsymbol{\mu}}^{(m)})^T}{\sum_\tau \gamma_m(\tau)} \qquad (8)$$

Unfortunately in contrast to LDA, HLDA has no simple solution. Standard non-linear optimisation schemes may be used to obtain the parameters of $\mathbf{A}$ [11, 9]. Alternatively, a computationally efficient scheme for finding the values is given in [8]. This simple iterative scheme is guaranteed to find a locally optimal solution and to be stable. In [11] a comparison of LDA and HLDA is performed on a simple digit task. It was found HLDA out-performed LDA. This has also been observed for LVCSR [9]

For both LDA and HLDA the model parameters may be simply estimated once the projection matrix, $\mathbf{A}_{[p]}$, has been determined. The ML estimates of the subspace mean, $\boldsymbol{\mu}_{[p]}^{(m)}$, and diagonal covariance matrix, $\mathbf{\Sigma}^{(m)}_{\text{diag}[p]}$, for component m are

$$\boldsymbol{\mu}^{(m)}_{[p]} = \mathbf{A}_{[p]}\check{\boldsymbol{\mu}}^{(m)} \tag{9}$$

and

$$\mathbf{\Sigma}^{(m)}_{\text{diag}[p]} = \text{diag}\left(\mathbf{A}_{[p]}\mathbf{W}^{(m)}\mathbf{A}^{T}_{[p]}\right) \tag{10}$$

The likelihood is only calculated in the useful subspace specified by the projection matrix. Thus the cost[5] is $\mathcal{O}(Mp + np)$, where M is the number of Gaussian components in the system. For LVCSR, depending on the ratio of p to n, this can dramatically reduce the cost compared to the likelihood calculation in the original space which costs $\mathcal{O}(Mn)$. Furthermore, the number of parameters is also reduced. In the original space there are $(2Mn)$ parameters compared to $(2Mp + 2(n-p) + n^2)$ for the HLDA case.

3 Semi-Tied Covariance Matrices

For both LDA and HLDA only a single subspace was obtained. In many tasks the optimal feature space may be dependent on the specific class that generated the data. For example in speech recognition the best subspace may depend on whether the phone is a vowel or a consonant. In this case it is necessary to have multiple transformation classes. One such scheme is semi-tied covariance matrices (STC) [8]. In this form of modelling each of the classes (in this case Gaussian components) is uniquely assigned[6] to one of R disjoint sets, or transform classes, $M^{(1)}$ to $M^{(R)}$. Each of these transform classes has an independent transform associated with it. The choice of R is dependent on the amount of training data available and the computational power available during recognition. Rather than viewing the transforms as acting on a feature-space partitioned according to the component generating the observation, they may be viewed as acting on the model parameters, hence the name. The covariance matrix for a particular Gaussian component m, $\check{\mathbf{\Sigma}}^{(m)}$, may be expressed as

$$\check{\mathbf{\Sigma}}^{(m)} = \mathbf{F}^{(r_m)}\mathbf{\Sigma}^{(m)}_{\text{diag}}\mathbf{F}^{(r_m)T} \tag{11}$$

where $\mathbf{\Sigma}^{(m)}_{\text{diag}}$ is the component specific diagonal matrix[7], $\mathbf{F}^{(r)}$ is the transform class specific full matrix and r_m indicates the transform class that component m belongs to (i.e. $m \in M^{(r_m)}$). Thus, though the effective covariance matrix is full the number of free parameters is significantly fewer then for full-covariance modelling. For M full covariance matrix components there are $M(n+1)n/2$ parameters associated with the covariance matrices. For STC matrices with R transform classes there are $Rn^2 + Mn$ parameters. For typical cases where $R << M$ there is a dramatic decrease in the number of model parameters. In addition to reducing the number of model parameters, this form of covariance modelling allows efficient likelihood calculation. Using the inverse of the

[5]For all likelihood computation costs order of terms are given. For likelihood calculation the number of multiply accumulates is used for the log-likelihood calculation. No account is taken of the log-addition for multiple components (though this is not necessary if a simple max is used). For the number of parameters the weights and transition matrices are not included. These numbers will be used to determine the entries in table 1.

[6]There are various options for this assignment ranging from expert knowledge (such as the same phone) to ML versions similar to K-means clustering[8].

[7]This is not a strict requirement. Other forms such as block-diagonal could be used in theory, but for current state-of-the-art LVCSR, due to efficiency reasons, a diagonal form has always been used. The use of non-diagonal cases for HLDA was examined on a small vocabulary task in [11].

transform matrix, $\mathbf{A}^{(r)} = \mathbf{F}^{(r)-1}$,

$$\begin{aligned}
\mathcal{L}(\mathbf{o};\check{\boldsymbol{\mu}}^{(m)},\check{\boldsymbol{\Sigma}}^{(m)}) &= \mathcal{N}(\mathbf{o};\check{\boldsymbol{\mu}}^{(m)},\check{\boldsymbol{\Sigma}}^{(m)}) \\
&= |\det(\mathbf{A}^{(r_m)})|\mathcal{N}(\mathbf{A}^{(r_m)}\mathbf{o};\mathbf{A}^{(r_m)}\check{\boldsymbol{\mu}}^{(m)},\boldsymbol{\Sigma}^{(m)}_{\text{diag}}) \\
&= |\det(\mathbf{A}^{(r_m)})|\mathcal{N}(\mathbf{A}^{(r_m)}\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}_{\text{diag}}) \\
&= \mathcal{L}(\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}_{\text{diag}},\mathbf{A}^{(r_m)}) \qquad (12)
\end{aligned}$$

The cost of full-covariance matrix likelihood calculation is $\mathcal{O}(Mn^2)$ compared to $\mathcal{O}(Rn^2 + Mn)$ for the STC case.

The complete set of model parameters for STC modelling, $\mathcal{M}$, may be written as[8]

$$\mathcal{M} = \left\{\left\{\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(R)}\right\},\left\{\boldsymbol{\Sigma}^{(1)}_{\text{diag}},\ldots,\boldsymbol{\Sigma}^{(M)}_{\text{diag}}\right\},\left\{\boldsymbol{\mu}^{(1)},\ldots,\boldsymbol{\mu}^{(M)}\right\}\right\} \qquad (13)$$

where $\boldsymbol{\mu}^{(m)}$ is the mean of Gaussian component m. The transform parameters are described in terms of the inverse of the transform matrix because it has been found to yield efficient optimisation schemes [8] and simplifies the tying requirements that will be used later in this paper.

There is a close relationship between STC modelling and HLDA. STC with a single transform class is identical to HLDA with $p = n$. Moreover, when $p < n$ HLDA may be viewed as a particular form of tying of the STC model. The number of transform classes is restricted to be one ($R = 1$) and the final $n - p$ dimensions of the component means and variances, $\boldsymbol{\mu}^{(m)}$ and $\boldsymbol{\Sigma}^{(m)}_{\text{diag}}$ are tied to be the same. Now

$$\boldsymbol{\mu}^{(m)} = \begin{bmatrix} \boldsymbol{\mu}^{(m)}_{[p]} \\ \mathbf{A}_{[n-p]}\check{\boldsymbol{\mu}} \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}^{(m)}_{\text{diag}} = \begin{bmatrix} \boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]} & 0 \\ 0 & \text{diag}(\mathbf{A}_{[n-p]}\check{\boldsymbol{\Sigma}}\mathbf{A}^T_{[n-p]}) \end{bmatrix} \qquad (14)$$

It is simple to show that this form of tying yields the same ML optimisation as HLDA. For standard LDA an additional level of tying is required where $\boldsymbol{\Sigma}^{(m)}_{\text{diag}} = \boldsymbol{\Sigma}_{\text{diag}}$. The rest of this section details the optimisation of the STC system. The modifications required for the standard HLDA system may be viewed as a restricted version of the multiple HLDA system where $R = 1$, discussed later in this paper.

ML estimation is used to find the STC model parameters. For the labelled training data, $\mathbf{O} = \{\mathbf{o}(1),\ldots,\mathbf{o}(T)\}$, the following function must be optimised with respect to the model parameters $\mathcal{M}$

$$\mathcal{L}(\mathbf{O}|\mathcal{M}) = \sum_{\boldsymbol{\Theta}}\prod_{\tau}\left(p(q(\tau)|q(\tau-1))\sum_{m\in\theta(\tau)} w^{(m)}\mathcal{L}(\mathbf{o}(\tau);\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}_{\text{diag}},\mathbf{A}^{(r_m)})\right) \qquad (15)$$

where $\boldsymbol{\Theta}$ is the set of all valid state sequences according to the transcription for the data, $q(\tau)$ is the state at time τ of the current path and $\theta(\tau)$ is set of Gaussian components belonging to state $q(\tau)$. Directly optimising equation 15 is a very large optimisation task, as there are typically millions of model parameters. Alternatively, as is common with standard HMM training, an expectation-maximisation (EM) [5] approach is used. In contrast to standard HMM training a generalised EM scheme is required since there are no simple closed-form solutions to find $\mathbf{A}^{(r)}$.

In generalised EM the auxiliary function $\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}})$, where $\hat{\mathcal{M}}$ is the set of "old" model parameters and $\mathcal{M}$ is the set of "new" model parameters, is optimised with respect to the new model parameters. The auxiliary function is given by

$$\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) = \frac{1}{2}\sum_{r,m\in M^{(r)},\tau}\gamma_m(\tau)\log\left(\frac{|\det\left(\mathbf{A}^{(r)}\right)|^2}{\left|\det\left(\text{diag}\left(\mathbf{A}^{(r)}\mathbf{W}^{(m)}\mathbf{A}^{(r)T}\right)\right)\right|}\right) - \frac{n\beta}{2}(\log(2\pi)+1) \qquad (16)$$

[8]Here the set of Gaussian component priors, or weights, $\{w^{(1)},\ldots,w^{(M)}\}$, and the state transition probabilities, have not been included. The estimation of these parameters is identical to the standard HMM parameter estimation [13].

where

$$\beta = \sum_r \beta^{(r)} = \sum_{r, m \in M^{(r)}, \tau} \gamma_m(\tau) \tag{17}$$

and $\gamma_m(\tau)$ is the posterior probability of Gaussian component m and time instance τ given the old model set and all the training data.

Though equation 16 could be directly optimised using non-linear optimisation schemes a simple efficient iterative scheme is possible [8]. The following scheme is used

1. Estimate the within class covariance matrix for each Gaussian component in the system, $\mathbf{W}^{(m)}$.

2. Using the current estimate of the transform, $\mathbf{A}^{(r)}$, obtain the ML estimate of the set of component specific diagonal covariance matrices incorporating the appropriate parameter tying as required. This set of parameters will be denoted as $\left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\} = \left\{\boldsymbol{\Sigma}^{(1)}_{\text{diag}}, \ldots, \boldsymbol{\Sigma}^{(M)}_{\text{diag}}\right\}$. This is a standard problem and will only be mentioned at each stage briefly.

3. Estimate the new transform $\mathbf{A}^{(r)}$ using the current set $\left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\}$.

4. Goto (2) until convergence, or appropriate criterion satisfied.

At each stage the likelihood is guaranteed to increase or remain the same. This form of optimisation is preferable to the direct non-linear optimisation as it is computationally efficient, typically the algorithm converges after 4 or 5 iterations, and is guaranteed to be stable [8].

For STC, step (3) of the above scheme is itself an iterative estimation scheme. Rewriting equation 16 using the fixed set $\left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\}$ (all terms independent of $\mathbf{A}^{(r)}$ are ignored) yields[9]

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}; \left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\}) = \sum_r \left\{ \beta^{(r)} \log\left((\mathbf{c}_i^{(r)} \mathbf{a}_i^{(r)T})^2 \right) - \sum_j \left(\mathbf{a}_j^{(r)} \mathbf{G}^{(rj)} \mathbf{a}_j^{(r)T} \right) \right\} \tag{18}$$

where $\mathbf{a}_i^{(r)}$ is the i^{th} row of $\mathbf{A}^{(r)}$, the $1 \times n$ row vector $\mathbf{c}_i^{(r)}$ is the i^{th} row vector of cofactors of $\mathbf{A}^{(r)}$ and

$$\mathbf{G}^{(rj)} = \sum_{m \in M^{(r)}} \frac{1}{\sigma^{(m)2}_{\text{diag}j}} \mathbf{W}^{(m)} \sum_\tau \gamma_m(\tau) \tag{19}$$

$\sigma^{(m)2}_{\text{diag}j}$ is the j^{th} leading diagonal element of $\boldsymbol{\Sigma}^{(m)}_{\text{diag}}$. It is possible to show that the ML solution for row i is[10]

$$\mathbf{a}_i^{(r)} = \mathbf{c}_i^{(r)} \mathbf{G}^{(ri)-1} \sqrt{\left(\frac{\beta^{(r)}}{\mathbf{c}_i^{(r)} \mathbf{G}^{(ri)-1} \mathbf{c}_i^{(r)T}} \right)} \tag{20}$$

The optimisation is thus an iterative one where each row of $\mathbf{A}^{(r)}$ is optimised given the current value of all the other rows. The computational cost of this inner loop is low since there are simple sufficient statistics. The major cost is calculating the cofactors of the transform matrices which is small compared to obtaining the $\mathbf{G}^{(rj)}$ accumulates for large values of M.

An alternative scheme for generating the semi-tied transforms is to use non-ML estimation, such as state-specific rotations [12]. In [8] a comparison of STC matrices with state-specific rotations was performed. It was found that STC modelling out-performed the state-specific rotation scheme.

[9] This uses the equality $|\det(\mathbf{A}^{(r)})| = \mathbf{c}_i^{(r)} \mathbf{a}_i^{(r)T}$. For further details of the derivation of this equation see [8].

[10] There are two possible, equivalent, solutions one positive, the other negative. It makes no difference which is selected as they will yield the same likelihood. For this paper the positive root is always selected.

4 Multiple Subspace Projections

The previous two sections have first described the use of LDA and HLDA to perform projections to an "optimal" feature space. Then STC systems were described which may be viewed as using multiple linear transformations of the feature-space without performing any projections. This section describes how the two forms of modelling may be combined to give multiple subspace projections that can be trained in an ML fashion. In the same way that HLDA was described, the multiple subspace projection schemes will be described in terms of tying the parameters of an STC system.

4.1 Multiple HLDA

Heteroscedastic LDA (HLDA) [11] was described in section 2. Multiple HLDA may be viewed as an extension to HLDA where the classes are partitioned into distinct sets, then a separate HLDA transform estimated for each of these sets. Multiple HLDA may also be considered as varying the model complexity[11], rather than a projection scheme. This is illustrated for a simple case in figure 3.

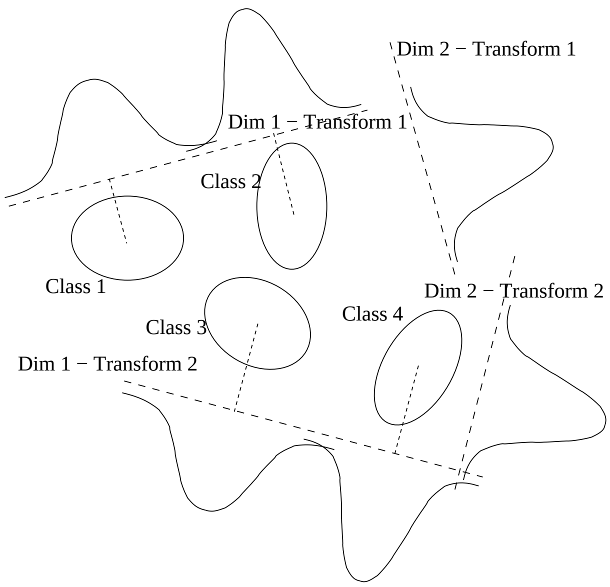

Figure 3: Multiple HLDA

Two transform classes are shown, with classes 1 and 2 assigned to transform class 1, and classes 3 and 4 assigned to transform class 2. In both cases the 2-dimensional case is projected down to a single useful dimension *Dim 1 - Transform 1* and *Dim 1- Transform 2*. There are two important things to notice. First both transforms span the same, original, space[12], thus when suitably normalised the likelihoods of the transform classes may be directly compared. Second, though the nuisance dimensions for each transform are a single Gaussian, they discriminate between the transform classes. Figure 3 may also be used to illustrate the interpretation of multiple HLDA as varying the model complexity. Consider the situation where classes 1 and 2 are components a mixture, similarly classes 3 and 4. Using multiple LDA mixture distributions are only kept for *Dim 1*, the useful dimension, whereas *Dim 2* has a single Gaussian distribution associated with it. The advantage of this approach over the standard multiple stream approach [21] is that there is no *a-priori* specification of which dimensions are assigned to have the more complex distributions.

[11] In this case the model complexity refers to the number of Gaussian components used to model a particular dimension and the number of states in the system that share these Gaussian components.

[12] This is guaranteed since during training when $|\det(\mathbf{A}^{(r)})| = 0$ the log-likelihood is $-\infty$.

For multiple HLDA the feature vector will be split into two subspaces[13] for each transform class; the useful p-dimensional subspace[14], and $(n-p)$-dimensional nuisance subspace where a simple single Gaussian component nuisance model is used for that transform class[15]. Thus

$$\mathbf{A}^{(r)} = \begin{bmatrix} \mathbf{A}^{(r)}_{[p]} \\ \mathbf{A}^{(r)}_{[n-p]} \end{bmatrix} \tag{21}$$

and

$$\boldsymbol{\mu}^{(m)} = \begin{bmatrix} \boldsymbol{\mu}^{(m)}_{[p]} \\ \boldsymbol{\mu}^{(r_m)}_{[n-p]} \end{bmatrix} \quad , \quad \boldsymbol{\Sigma}^{(m)}_{\text{diag}} = \begin{bmatrix} \boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}^{(r_m)}_{\text{diag}[n-p]} \end{bmatrix} \tag{22}$$

where

$$\boldsymbol{\mu}^{(r)}_{[n-p]} = \mathbf{A}^{(r)}_{[n-p]}\check{\boldsymbol{\mu}}^{(r)} \tag{23}$$

$$\boldsymbol{\Sigma}^{(r)}_{\text{diag}[n-p]} = \text{diag}\left(\mathbf{A}^{(r)}_{[n-p]}\check{\boldsymbol{\Sigma}}^{(r)}\mathbf{A}^{(r)T}_{[n-p]}\right) \tag{24}$$

$$\check{\boldsymbol{\Sigma}}^{(r)} = \frac{\sum_{m\in M^{(r)},\tau}\gamma_m(\tau)\left(\mathbf{o}(\tau)-\check{\boldsymbol{\mu}}^{(r)}\right)\left(\mathbf{o}(\tau)-\check{\boldsymbol{\mu}}^{(r)}\right)^T}{\sum_{m\in M^{(r)},\tau}\gamma_m(\tau)} \tag{25}$$

and

$$\check{\boldsymbol{\mu}}^{(r)} = \frac{\sum_{m\in M^{(r)},\tau}\gamma_m(\tau)\mathbf{o}(\tau)}{\sum_{m\in M^{(r)},\tau}\gamma_m(\tau)} \tag{26}$$

The likelihood calculation for multiple HLDA is more efficient than the computation for STC, since certain dimensions of the feature-vector are modelled with simple transform class dependent distributions. The likelihood is computed as

$$\mathcal{L}(\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}_{\text{diag}},\mathbf{A}^{(r_m)}) = l^{(r_m)}|\det(\mathbf{A}^{(r_m)})|\mathcal{N}(\mathbf{A}^{(r_m)}_{[p]}\mathbf{o};\boldsymbol{\mu}^{(m)}_{[p]},\boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]}) \tag{27}$$

where

$$l^{(r)} = \mathcal{N}(\mathbf{A}^{(r)}_{[n-p]}\mathbf{o};\boldsymbol{\mu}^{(r)}_{[n-p]},\boldsymbol{\Sigma}^{(r)}_{\text{diag}[n-p]}) \tag{28}$$

$l^{(r)}$ need only be calculated once for each transform class. Thus the likelihood calculation cost is $\mathcal{O}(Mp + Rn^2 + R(n-p))$. Note multiple HLDA is not a true projection scheme, since all the dimensions contain some class information with the transform-class specific nuisance dimensions discriminating between transform classes. Therefore the likelihoods used during recognition must be computed, though efficiently, in the original n-dimensional space. The number of parameters is reduced compared to the STC case. For multiple HLDA there are $(2Mp + R(n^2 + 2(n-p)))$ parameters to be estimated compared to $(2Mn + Rn^2)$ for STC systems.

For the single HLDA case the computational cost is less (even ignoring the need to do the transform class specific matrix-vector multiplications). The final $n-p$ dimensions need not be

[13] Greater flexibility in complexity of the models may be obtained by using a hierarchy of tying, though this will not be investigated in this paper and is briefly described in appendix B.

[14] It is not necessary for p to be the same for all transform classes. However in this work p was constrained to be the same for all classes.

[15] More complicated multiple component noise models may be simply introduced. The nuisance dimensions components are treated as a separate stream [21] to the useful parameters. In terms of the optimisation, the change is that $\mathbf{K}^{(rj)}$ in equation 31 is now a sum over all the nuisance components instead of the single noise Gaussian. This is described in more detail in appendix A.

computed since they do not discriminate between the classes (single HLDA is a true projection scheme). Thus the "likelihood" may be computed as[16]

$$\mathcal{L}(\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}_{\text{diag}},\mathbf{A}) = |\det(\mathbf{A})|\mathcal{N}(\mathbf{A}_{[p]}\mathbf{o};\boldsymbol{\mu}^{(m)}_{[p]},\boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]}) \tag{29}$$

The likelihood calculation cost is therefore only $\mathcal{O}(Mp+np)$.

Optimising the parameters of the multiple HLDA transform case is a modification to the semi-tied transform case. It is possible to rewrite equation 16 as (ignoring all expressions that are independent of $\mathbf{A}^{(r)}$)

$$\begin{aligned}\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}};\left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\}) &= \sum_r \left\{\beta^{(r)}\log\left((\mathbf{c}_i^{(r)}\mathbf{a}_i^{(r)T})^2\right) - \sum_{j\le p}\left(\mathbf{a}_j^{(r)}\mathbf{G}^{(rj)}\mathbf{a}_j^{(r)T}\right)\right.\\ &\qquad \left. - \sum_{j>p}\mathbf{a}_j^{(r)}\mathbf{K}^{(rj)}\mathbf{a}_j^{(r)T}\right\} \end{aligned}\tag{30}$$

where

$$\mathbf{K}^{(rj)} = \frac{1}{\sigma^{(r)2}_{\text{diag}j}}\check{\boldsymbol{\Sigma}}^{(r)}\sum_{m\in M^{(r)},\tau}\gamma_m(\tau) \tag{31}$$

For rows $i \le p$, differentiating equation 30 with respect to $\mathbf{a}_i^{(r)}$ and equating to zero yields the standard STC re-estimation formula given in equation 20. For rows $i > p$ following the same procedure gives

$$\mathbf{a}_i^{(r)} = \mathbf{c}_i^{(r)}\mathbf{K}^{(ri)-1}\sqrt{\left(\frac{\beta^{(r)}}{\mathbf{c}_i^{(r)}\mathbf{K}^{(ri)-1}\mathbf{c}_i^{(r)T}}\right)} \tag{32}$$

The estimates for the mean and variance have the standard form described for LDA and HLDA.

4.2 Multiple LDA

In the previous section, all dimensions using multiple HLDA transforms were found to contain some class information. This section describes the additional tying of the STC system required to generate a global nuisance subspace, whilst using multiple transformations. To generate the global nuisance subspace the final $n-p$, the nuisance dimensions, distribution parameters and the transform parameters are tied. Thus

$$\mathbf{A}^{(r)} = \begin{bmatrix}\mathbf{A}^{(r)}_{[p]}\\ \mathbf{A}_{[n-p]}\end{bmatrix} \tag{33}$$

and

$$\boldsymbol{\mu}^{(m)} = \begin{bmatrix}\boldsymbol{\mu}^{(m)}_{[p]}\\ \mathbf{A}_{[n-p]}\check{\boldsymbol{\mu}}\end{bmatrix} \quad , \quad \boldsymbol{\Sigma}^{(m)}_{\text{diag}} = \begin{bmatrix}\boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]} & \mathbf{0}\\ \mathbf{0} & \text{diag}\left(\mathbf{A}_{[n-p]}\check{\boldsymbol{\Sigma}}\mathbf{A}^T_{[n-p]}\right)\end{bmatrix} \tag{34}$$

This form of transform will be called multiple LDA (MLDA)[17].

A simple example of MLDA is shown in figure 4. Four classes and two transform classes are shown, with classes 1 and 2 assigned to transform class 1, 3 and 4 to transform class 2. *Dim 2* is the

[16] This is, of course, not the likelihood in the original space, but in the projected space of the useful dimensions.

[17] The name is motivated by the fact that the scheme can be used for optimising multiple LDA transforms by simply assuming that all the within class covariance matrices are the same. Though a closed-form solution to the optimisation is still not possible, the sufficient statistics to obtain the model parameters are greatly simplified.

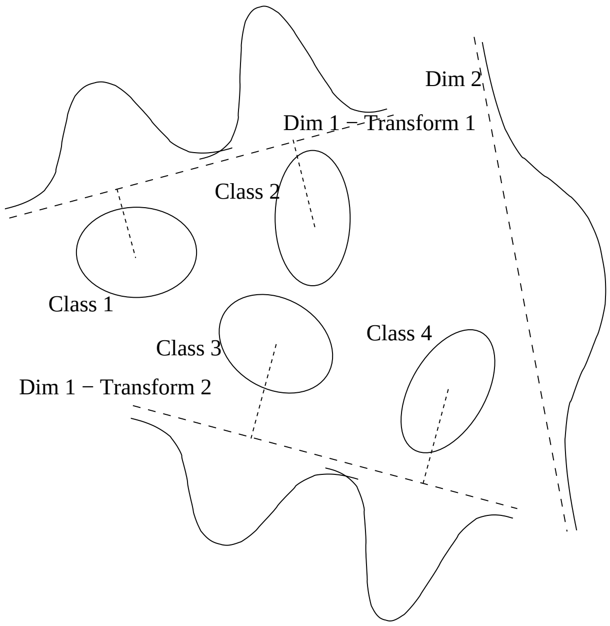

Figure 4: Multiple Linear Discriminant Analysis

nuisance dimension, common to both transform classes and all the data in *Dim 2* is modelled by a single Gaussian distribution. There are two different projections for the useful dimension, labelled *Dim 1 - Transform 1* and *Dim 1 - Transform 2*. In both cases, the nuisance and useful dimensions must span the original feature space[18]. Thus, when appropriately normalised, likelihoods in the new spaces will yield valid likelihoods in the 2-dimensional space. However now, since the nuisance parameters, the distribution in *Dim 2*, no longer achieve any class discrimination, a more efficient likelihood calculation is possible.

During recognition the likelihood is calculated as

$$\mathcal{L}(\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}_{\text{diag}},\mathbf{A}^{(r_m)}) = |\det(\mathbf{A}^{(r_m)})|\mathcal{N}(\mathbf{A}^{(r_m)}_{[p]}\mathbf{o};\boldsymbol{\mu}^{(m)}_{[p]},\boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]}) \tag{35}$$

Thus for the example in figure 4 the likelihoods for classes 1 and 2 need only be computed in *Dim 1 - Transform 1*. The total likelihood cost is therefore $\mathcal{O}(Mp+Rpn)$, cheaper than both STC and the equivalent multiple HLDA systems. In terms of the number of model parameters, there are $(2Mp+Rnp+2(n-p)+n(n-p))$ parameters to estimate, again less than STC or the multiple HLDA schemes.

Optimising the parameters of the MLDA system is more complicated than for STC or HLDA systems. Again a variation on the simple iterative scheme used to optimise the STC parameters is used. The additional complications occurs in estimating $\mathbf{A}^{(r)}$ given the current estimates of the, tied, covariance matrices. Depending on the row of the transform to be estimated, the auxiliary function has two different forms. For the useful dimensions, rows 1 to p, $\mathbf{a}^{(r)}_i$ varies according to the transform class. The auxiliary function is

$$\begin{aligned}\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}};\left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\}) &= \sum_r \left\{\beta^{(r)}\log\left((\mathbf{c}^{(r)}_i\mathbf{a}^{(r)T}_i)^2\right) - \sum_{j\leq p}\left(\mathbf{a}^{(r)}_j\mathbf{G}^{(rj)}\mathbf{a}^{(r)T}_j\right)\right\} \\ &\quad -\sum_{j>p}\mathbf{a}_j\mathbf{K}^{(j)}\mathbf{a}^T_j\end{aligned} \tag{36}$$

[18]This is guaranteed by the optimisation since $|\det(\mathbf{A}^{(r_m)})| = 0$ results in a log-likelihood of $-\infty$.

where $\mathbf{G}^{(rj)}$ is defined in equation 19 and

$$\mathbf{K}^{(j)} = \frac{1}{\sigma^2_{\text{diag}j}} \check{\boldsymbol{\Sigma}} \sum_{m,\tau} \gamma_m(\tau) \tag{37}$$

Differentiating this with respect to $\mathbf{a}_i^{(r)}$ and equating to zero yields the standard STC estimation formula given in equation 20 for rows $i \leq p$. For the nuisance dimensions, rows $i > p$, the auxiliary function may be simplified, since $\mathbf{a}_i$ is tied so as not to depend on the transform class. Equation 36 may be rewritten as

$$\begin{aligned} \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}; \left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\}) \;\; = \;\; & \sum_r \left\{ \beta^{(r)} \log\left((\mathbf{c}_i^{(r)} \mathbf{a}_i^T)^2 \right) - \sum_{j \leq p} \left(\mathbf{a}_j^{(r)} \mathbf{G}^{(rj)} \mathbf{a}_j^{(r)T} \right) \right\} \\ & - \sum_{j>p} \mathbf{a}_j \mathbf{K}^{(j)} \mathbf{a}_j^T \end{aligned} \tag{38}$$

Now differentiating with respect to $\mathbf{a}_i$

$$\mathbf{f}^{(i)} = \frac{\partial \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}; \left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\})}{\partial \mathbf{a}_i} = 2 \sum_r \frac{\beta^{(r)} \mathbf{c}_i^{(r)}}{\mathbf{c}_i^{(r)} \mathbf{a}_i^T} - 2\mathbf{a}_i \mathbf{K}^{(i)} \tag{39}$$

There are two situations to consider. The first is when $R < n$ (i.e. there are fewer transforms than dimensions). Here it is possible to simplify the optimisation. Assuming that $\mathbf{K}^{(i)}$ is of full rank then it is clear that at the ML solution (i.e. $\mathbf{f}^{(i)} = \mathbf{0}$)

$$\mathbf{a}_i = \sum_r \lambda_r^{(i)} \mathbf{c}_i^{(r)} \mathbf{K}^{(i)-1} \tag{40}$$

Now, the problem is to optimise the likelihood with respect to $\boldsymbol{\lambda}^{(i)}$, an R-dimensional vector. Thus an n-dimensional optimisation problem, finding $\mathbf{a}_i$ directly, has been transformed into an R-dimensional problem of finding $\boldsymbol{\lambda}^{(i)}$. Optimisation for this particular version follows the same general scheme as that described below. For the special case when $R = 1$ equating equation 39 to zero gives the standard HLDA re-estimation formulae.

The second, more general, situation allows $R \geq n$. Here there are no gains in transforming the problem as above since the cofactor vectors will span all possible space, thus yielding no useful constraints. Proceeding directly with a Newton-based optimisation scheme, the Hessian for row i, $\mathbf{H}^{(i)}$, is

$$\mathbf{H}^{(i)} = \frac{\partial^2 \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}; \left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\})}{\partial \mathbf{a}_i^2} = -2 \sum_r \frac{\beta^{(r)} \mathbf{c}_i^{(r)T} \mathbf{c}_i^{(r)}}{(\mathbf{c}_i^{(r)} \mathbf{a}_i^T)^2} - 2\mathbf{K}^{(i)} \tag{41}$$

The calculation of the Hessian is highly efficient, dominated by the calculation of the cofactors which must be calculated for the gradient. As expected this expression is semi-negative definite indicating a concave error surface[19]. The update formula becomes

$$\mathbf{a}_i = \hat{\mathbf{a}}_i - \Delta \mathbf{f}^{(i)} \mathbf{H}^{(i)-1} \tag{42}$$

where Δ is determined using a line search technique. Note for the line search it is only necessary to evaluate those elements dependent on the particular row of interest, in this case row i. Thus at each point $\mathcal{Q}^{(i)}(\mathcal{M}, \hat{\mathcal{M}}; \left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\})$ is evaluated where

$$\mathcal{Q}^{(i)}(\mathcal{M}, \hat{\mathcal{M}}; \left\{\tilde{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\}) = \sum_r \beta^{(r)} \log\left((\mathbf{c}_i^{(r)} \mathbf{a}_i^T)^2 \right) - \mathbf{a}_i \mathbf{K}^{(i)} \mathbf{a}_i^T \tag{43}$$

[19]There are two concave surfaces separated by a discontinuity when the determinant of $\mathbf{A}$ is zero.

This scheme was found to be highly stable, $\Delta = 1$ was the typical value used. The total number of outer iterations, model updates, was about the same speed as required to train the STC systems.

There is a simpler approach to obtaining an MLDA-like set of transformations. First a single HLDA transform may be performed to find the p useful dimensions. Then multiple, transform class specific, semi-tied transformations are obtained in the reduced p-dimensional space. Thus the overall transform for transform class r may be written as

$$\mathbf{A}_{[p]}^{(r)} = \mathbf{A}_{\mathbf{stc}}^{(r)} \mathbf{A}_{\mathbf{hlda}[p]} \tag{44}$$

and

$$\mathbf{A}_{[n-p]} = \mathbf{A}_{\mathbf{hlda}[n-p]} \tag{45}$$

where $\mathbf{A}_{\mathbf{stc}}^{(r)}$ is the transform class specific $p \times p$ semi-tied transform and $\mathbf{A}_{\mathbf{hlda}}$ is the $n \times n$ single HLDA transform. The disadvantage of this approach is that it is distinctly two stage, whereas MLDA does the simultaneous optimisation. This two stage MLDA computational will be referred to as MLDA(2). The likelihood of this scheme may be efficiently computed as

$$\mathcal{L}(\mathbf{o}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}_{\text{diag}}^{(m)}, \mathbf{A}^{(r_m)}) = |\det(\mathbf{A}^{(r_m)})| \mathcal{N}(\mathbf{A}_{\mathbf{stc}}^{(r_m)} \tilde{\mathbf{o}}_{[p]}; \boldsymbol{\mu}_{[p]}^{(m)}, \boldsymbol{\Sigma}_{\text{diag}[p]}^{(m)}) \tag{46}$$

where

$$\tilde{\mathbf{o}}_{[p]} = \mathbf{A}_{\mathbf{hlda}[p]} \mathbf{o} \tag{47}$$

The likelihood cost with this modified form is $\mathcal{O}(Mp + Rp^2 + np)$, typically slightly less than MLDA. MLDA and MLDA(2) are very similar when the space spanned by the nuisance dimensions is orthogonal to all the useful dimension spaces. In this case the efficient likelihood calculation of equation 46 may be used for MLDA and the effective number of free parameters reduced accordingly. However, the optimisation given for MLDA does not ensure this constraint. Due to this constraint there are usually slightly fewer free parameters, $(2Mp + Rp^2 + n^2 + 2(n-p))$ with MLDA(2), than MLDA

5 Linear Gaussian Models

The schemes described up to this point should be compared to the use of subspace modelling of signals in other areas of machine learning. Based on linear Gaussian models, they have been used for low-dimensional visualisation [19] and determination of independent hidden sources [1]. Since the speed of computing the likelihood is not as important in these tasks as for the density modelling described in this paper, the form of the model is slightly different to the one previously described. These differences have considerable impact in both training and computing the likelihoods of such schemes.

Low dimensional modelling schemes based on ML linear subspaces have previously been proposed. These include schemes such as factor analysis (FA) [16] and probabilistic PCA (PPCA) [19]. In [15] a general framework for linear Gaussian models is given.

$$\mathbf{z}(\tau+1) = \mathbf{C}\mathbf{z}(\tau) + \mathbf{w} \tag{48}$$
$$\mathbf{o}(\tau) = \mathbf{E}_{[p]}^{T}\mathbf{z}(\tau) + \mathbf{v} \tag{49}$$

where $\mathbf{C}$ is the $p \times p$ state transition matrix, the state noise, $\mathbf{w}$, is p-dimensional zero mean Gaussian distributed, the sensor noise, $\mathbf{v}$, is n-dimensional zero-mean Gaussian distributed and $\mathbf{E}_{[p]}$ is the $p \times n$ observation matrix. A simplification of this is to consider only static data modelling. Here

$$\mathbf{z}(\tau+1) = \mathbf{w} \tag{50}$$

For this form of modelling the likelihood is computed in the original feature space since the sensor noise is modelled in the original space. There are therefore no issues in comparing likelihoods from multiple subspaces.

LDA and the other schemes described in the previous sections do not directly fit into this framework. They differ in two ways. First the state noise is distributed according to a Gaussian mixture model or from an HMM, though non-Gaussian state-noise distributions have previously been considered such as independent factor analysis (IFA) [1]. Second there are restrictions on the form of the sensor noise. The generative model for LDA requires a modification to equation 49, so that

$$\mathbf{o}(\tau) \quad = \quad \mathbf{E}\left[\begin{array}{c} \mathbf{z}(\tau) \\ \mathbf{v} \end{array}\right] \tag{51}$$

where v is now an $n-p$-dimensional random variable, whose distribution is determined by the form of the noise model. This modification alters both the training and likelihood calculation costs of the two schemes. All the form of models considered in [15] and IFA [1][20] may be trained using EM. None of the subspace projection schemes discussed in this paper may be trained in this fashion, they require non-linear optimisation schemes to be used. There are also significant differences in the cost of computing the likelihood. Consider the case of IFA with p factors. The cost of calculating the likelihood for IFA is $\mathcal{O}(M(n+p^2)+np+n^2)$ (for more detail see appendix C). In contrast for the case of HLDA the cost is simply $\mathcal{O}(Mp+pn)$. Hence, for speech recognition where the number of Gaussian components can be tens of thousands and the number of dimensions typically around 40 the difference for any reasonable number of factors is large.

6 Results

The results presented in this section are to illustrate the various multiple subspace projection schemes on a speech recognition task. They are not aimed at achieving the best performance on this task. Though the models used are "good" models based on state-of-the-art HMM training techniques.

An internal IBM dictation task was used to examine the schemes. The training data is a large multiple speaker training set consisting of read speech data recorded in a quiet environment. The test data was set up to be a speaker-independent task (i.e. there is no overlap between the training and test speakers) with read speech recorded in a clean environment with the same microphone. The training data consists of around 300000 sentences of both scripted and unscripted data. A state clustered decision tree system was used throughout with 3430 states. There were approximately 12 Gaussian components used to model each state giving a total of 41268 Gaussian components in the system. The front end consisted of 24 cepstrum coefficients. Nine frames were spliced together and the dimensionality of the resultant feature vector reduced from 216 to 40 dimensions using LDA[21]. A single semi-tied transform was then calculated to further improve the diagonal Gaussian component approximation. On previous experiments using similar systems with this task, in common with other reported results, the performance using a standard LDA derived frontend was around 10% worse than the use of a single STC transform [9, 8]. Thus the baseline system considered was the single global STC system. For these experiments, where a projection scheme was used the nuisance dimension size was set to be 10. No attempt was made to optimise this value, though various complexity schemes, such as BIC [18] could be used to determine the optimal size. In addition the number of transform classes was fixed at the number of states, 3430. Also the assignment of component to transform class was determined by which state the component belonged to. The test set consists of 15 speakers uttering a total of around 20,000 words. The results quoted are the average word error rates over all test speakers. A trigram language model was used in all tests with a 64000 word vocabulary.

Table 1 show the number of free parameters for each of the schemes. For a global transform HLDA, MLDA and MLDA(2) are all identical. For the multiple state-level transformations the

[20] This is not true when IFA is trained with $p = n$. In this case similar optimisation schemes to STC may be

System	Useful Dim.	Parameters ($\times 10^6$)	
		global	state
STC	40	3.30	8.79
HLDA	30	2.48	8.03
MLDA	30	—	6.59
MLDA(2)	30	—	5.56

Table 1: Number of free parameters (ignoring component priors and transition probabilities) for STC, HLDA, MLDA and MLDA(2) systems using either a global or state level transformation tying.

MLDA and MLDA(2) schemes have the fewest parameters, since the nuisance dimensions are global.

System	Useful Dim	Error rate (%)	
		global	state
STC	40	11.2	10.2
HLDA	30	12.1	10.1
MLDA	30	—	10.7
MLDA(2)	30	—	11.4

Table 2: Performance of a STC, HLDA, MLDA and MLDA(2) systems using either a global or state level transformation tying.

Table 2 shows the recognition performance of various systems using either global or state level tying of the transformation matrices. For HLDA the last 10 elements of the feature-vector, the nuisance dimensions, were modelled with a single Gaussian component per transform. For MLDA the nuisance dimensions were modelled using a single Gaussian components for all transforms. For the global case the STC system out performed the HLDA. This indicates that the projection from 40 dimensions to 30 dimensions (as previously mentioned a single HLDA transform performs ML projection) degrades performance. However, it reduces the number of model parameters by approximately 25%. Increasing the number of STC transforms, so there is a separate transform for each state, reduced the word error rate by about 10%, but increased the number of model parameters by a factor of about 2.5. Performing HLDA with the state-tied transform achieved the same performance as the STC system, with a small decrease in the number of model parameters. It is interesting that reducing the model complexity for 10 of the dimensions had no effect on the performance. Performing MLDA with the state-level transforms increased the word error rate by about 5% over the STC system but reduced the number of parameters by about 25%. Using the MLDA system also reduced the computational cost by about 25%.

In addition table 2 shows the performance of an MLDA(2) system. This is the two stage version of MLDA. The performance is slightly worse than that of MLDA. This indicates that the simultaneous optimisation of both the projection and the transform class specific transforms may lead to a reduction in word error rate. However, the use of MLDA does slightly increase the number of parameters.

Table 2 shows a general trend that performance improves as the number of model parameters increases. Rather than using multiple transforms, the number of parameters may be increased by simply increasing the number of Gaussian components. To obtain comparable performance to the multiple HLDA state system the number of components had to be increased by a factor of 4.

---

used.

[21]It would be preferable to perform this projection in an ML fashion. Unfortunately for the size of system used for large vocabulary speech recognition this is impractical at the present time.

Again a single global STC transform was used. This gave a word error rate of 9.9%. The total number of parameters in the system 13.3 million parameters, 50% larger than the HLDA state system that gave comparable performance.

7 Conclusions

This paper has examined the use of multiple, ML trained, subspace projections. Two forms of subspace projection were examined. Both forms were obtained by tying the parameters in a semi-tied covariance matrix system. The first, a multiple transform version of HLDA, was not a strict projection scheme, in the sense that there were no global nuisance dimensions, but rather a scheme for varying the model complexity over the dimensions. The second, MLDA, was a true projection scheme, in the sense that the nuisance dimensions had no class information, using multiple transforms. These forms of subspace projection were compared to the general class of linear Gaussian models. Though closely related, the two forms of model were found to differ in the form of the noise model. This had significant effects on both the estimation of the model parameters and the cost of calculating the component likelihoods. Specifically, calculating the likelihood of the linear Gaussian model was significantly more computationally expensive than the HLDA or MLDA systems.

The performance of the multiple subspace projection schemes were evaluated on a speaker independent speech recognition task. Using the multiple HLDA transforms to control the model complexity, it was found that was little difference in performance compared to the full system. This indicates that there are possible gains, in terms of both speech and model size, in having flexible model complexities over the dimensions. It was also found that simultaneously optimising both the subspace projection and the semi-tied transforms was better than a two-stage process. Though the results did not show gains of the use of multiple semi tied transforms, or increasing the number of Gaussian components in each state, they illustrate that using multiple subspace projections is another possibly useful option when designing LVCSR systems.

There are a number of points that this paper has not addressed. Methods for determining the appropriate number of useful features have not been examined. The complexity of the nuisance dimension model was set to a single Gaussian component. The use of more complex nuisance dimension models should be investigated. A simple hierarchical version of MLDA is also possible allowing even greater flexibility in the models choice. Furthermore ML estimation has been used to obtain the model parameters, rather than the discriminatively trained subspaces described in [14, 17]. It would be preferable to use a discriminative training scheme, such as maximum mutual information estimation (MMIE) [2]. However, for LVCSR tasks this is computationally very expensive, though recently MMIE has been successfully applied to an LVCSR task [20]. All these aspects will be addressed in future work.

A Nuisance Dimension Modelling

In the descriptions of the projection schemes the nuisance dimensions have been modelled by a single Gaussian distributions. This is not a necessary restriction, it is only required that the nuisance dimensions be modelled by a non-class discriminating model. The most obvious alternative model to use for the nuisance dimensions is a single Gaussian mixture model. The training then becomes closely akin to *multiple stream* systems [21], with the useful dimensions being one stream, the nuisance ones a second stream. However, in contrast to standard multiple stream systems, the dimensions assigned to each of the streams is not fixed a-priori, but varies according to the linear transform being applied. Let the set of Gaussian components associated with the nuisance dimensions be $M^{(n)}$ and the original model set, $\mathcal{M}$ be split into two, $\mathcal{M}_{[p]}$ for the useful dimensions and $\mathcal{M}_{[n-p]}$ for the nuisance dimensions. To simplify the discussion only a single HLDA transform will be considered (the dependence on the transform class will be dropped). Thus the model set for the nuisance dimensions is[22]

$$\mathcal{M}_{[n-p]} = \left\{\mathbf{A}_{[n-p]}, \left\{\mathbf{\Sigma}^{(1)}_{\text{diag}[n-p]},\ldots,\mathbf{\Sigma}^{(N)}_{\text{diag}[n-p]}\right\}, \left\{\boldsymbol{\mu}^{(1)}_{[n-p]},\ldots,\boldsymbol{\mu}^{(N)}_{[n-p]}\right\}\right\} \tag{52}$$

The standard single HLDA transform case is when $N = 1$. Using the multiple stream analogy the posteriors for these nuisance dimensions are computed independently from the useful dimensions. Let these posteriors be $\gamma_n(\tau)$. Thus at any time instances there are two sets of posteriors, those associated with the useful dimensions, $\gamma_m(\tau)$ and those with the nuisance dimensions, $\gamma_n(\tau)$. In terms of the optimisation, this modifies the definition of $\mathbf{K}^{(j)}$ to

$$\mathbf{K}^{(j)} = \sum_{n\in M^{(n)}} \frac{1}{\sigma^{(n)2}_{\text{diag}j}} \mathbf{W}^{(n)} \sum_{\tau} \gamma_n(\tau) \tag{53}$$

where

$$\gamma_n(\tau) = p(q_n(\tau)|\mathbf{O}, \mathcal{M}_{[n-p]}) \tag{54}$$

$q_n(\tau)$ indicates occupancy of Gaussian component n of the nuisance dimensions model at time instance τ. Similarly the posterior for the useful dimensions is given as

$$\gamma_m(\tau) = p(q_m(\tau)|\mathbf{O}, \mathcal{M}_{[p]}) \tag{55}$$

It is then possible to use the standard optimisation schemes to obtain both the linear transform, the useful model parameters and the nuisance model parameters.

It is not necessary to update the complete dataset for both the useful and nuisance dimensions, since they are independent. Multiple iterations optimising $\mathcal{M}_{[p]}$ may be performed followed by multiple iterations to find $\mathcal{M}_{[n-p]}$. Of course, in the case of a single Gaussian component model for the nuisance dimensions there is no need to estimate the complete dataset for the nuisance dimensions since the alignment is deterministic.

B Hierarchical Modelling

The flexibility of the MLDA system is limited as dimensions are either useful, or nuisance dimensions. A more general framework allows, for example, certain dimensions to be useful for some classes and nuisance for others. This naturally leads to a hierarchical model.

Figure 5 illustrates one such hierarchical scheme. There are two transform classes. However now there are both global nuisance dimensions (similar to MLDA) and transform class specific nuisance dimensions (similar to HLDA). The set of model parameters may be written as[23]

$$\mathbf{A}^{(r)} = \begin{bmatrix} \mathbf{A}^{(r)}_{[u^{(r)}]} \\ \mathbf{A}^{(r)}_{[p^{(r)}]} \\ \mathbf{A}_{[n^{(r)}]} \end{bmatrix} \tag{56}$$

$$\boldsymbol{\mu}^{(m)} = \begin{bmatrix} \boldsymbol{\mu}^{(m)}_{[u^{(r_m)}]} \\ \boldsymbol{\mu}^{(r_m)}_{[p^{(r_m)}]} \\ \boldsymbol{\mu}_{[n^{(r_m)}]} \end{bmatrix} \tag{57}$$

and

$$\mathbf{\Sigma}^{(m)}_{\text{diag}} = \begin{bmatrix} \mathbf{\Sigma}^{(m)}_{\text{diag}[u^{(r_m)}]} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}^{(r_m)}_{\text{diag}[p^{(r_m)}]} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}_{\text{diag}[n^{(r_m)}]} \end{bmatrix} \tag{58}$$

[22]In this case there are not transition probabilities, just priors (again not included), since a Gaussian mixture model is being used as the nuisance dimension model.

[23]Here a single Gaussian component is used to model the nuisance parameters. Again, this is not a necessary restriction.

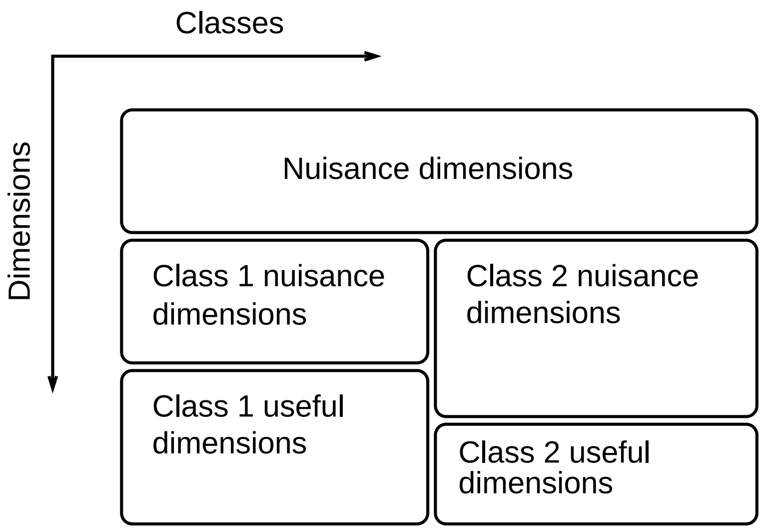

Figure 5: Hierarchical projection scheme

where $u^{(r)}$ and $p^{(r)}$ are the number of useful and class nuisance dimensions for transform class r, and $n^{(r)} = n - u^{(r)} - p^{(r)}$. Training the model parameters is a combination of the MLDA and HLDA training schemes.

Hierarchical modelling of this form is only useful if it is possible to efficient compute the appropriate model complexity for each of the transform classes. This is not necessary, for example, with an MLDA scheme since it is possible to perform a search over the complete range of the possible number of useful dimensions. One model complexity control scheme that is currently popular in the area of speech recognition is the Bayesian information criterion (BIC), particularly in the form of penalised BIC [4]. The application of BIC and other complexity control schemes will be examined in future work.

C Efficient Likelihood Computation for IFA

To compute the likelihood using IFA directly involves the calculation of a full covariance matrix. An alternative more efficient computation of the likelihood is possible. The likelihood may be written as

$$\begin{aligned}\mathcal{L}(\mathbf{o};\boldsymbol{\mu}_{[p]}^{(m)},\boldsymbol{\Sigma}_{[p]}^{(m)},\boldsymbol{\Lambda},\mathbf{E}_{[p]}) &= \mathcal{N}(\mathbf{o};\mathbf{E}_{[p]}^T\boldsymbol{\mu}_{[p]}^{(m)},\mathbf{E}_{[p]}^T\boldsymbol{\Sigma}_{\mathrm{diag}[p]}^{(m)}\mathbf{E}_{[p]}+\boldsymbol{\Lambda}) \\ &= \frac{k\exp\left(-\frac{1}{2}\mathcal{F}(\mathbf{o};\mathbf{E}_{[p]}^T\boldsymbol{\mu}_{[p]}^{(m)},\mathbf{E}_{[p]}^T\boldsymbol{\Sigma}_{\mathrm{diag}[p]}^{(m)}\mathbf{E}_{[p]}+\boldsymbol{\Lambda})\right)}{|\det\left(\mathbf{E}_{[p]}^T\boldsymbol{\Sigma}_{\mathrm{diag}[p]}^{(m)}\mathbf{E}_{[p]}+\boldsymbol{\Lambda}\right)|^{\frac{1}{2}}}\end{aligned} \tag{59}$$

where k is the standard normalising constant and

$$\mathcal{F}(\mathbf{o};\boldsymbol{\mu},\boldsymbol{\Sigma}) = (\mathbf{o}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{o}-\boldsymbol{\mu}) \tag{60}$$

This may be rewritten as

$$\mathcal{F}(\mathbf{o};\mathbf{E}_{[p]}^T\boldsymbol{\mu}_{[p]}^{(m)},\mathbf{E}_{[p]}^T\boldsymbol{\Sigma}_{\mathrm{diag}[p]}^{(m)}\mathbf{E}_{[p]}+\boldsymbol{\Lambda}) = \mathcal{F}(\mathbf{M}^{T-1}\mathbf{o};\tilde{\mathbf{E}}_{[p]}^T\boldsymbol{\mu}_{[p]}^{(m)},\tilde{\mathbf{E}}_{[p]}^T\boldsymbol{\Sigma}_{\mathrm{diag}[p]}^{(m)}\tilde{\mathbf{E}}_{[p]}+\mathbf{I}) \tag{61}$$

where $\tilde{\mathbf{E}}_{[p]}^T = \mathbf{M}^{T-1}\mathbf{E}_{[p]}^T$ and $\mathbf{M}^T\mathbf{M} = \mathbf{\Lambda}$. Using standard equalities this may be rewritten as

$$\begin{aligned}\mathcal{F}(\mathbf{o}; \mathbf{E}_{[p]}^T\boldsymbol{\mu}_{[p]}^{(m)}, \mathbf{E}_{[p]}^T\mathbf{\Sigma}_{\text{diag}[p]}^{(m)}\mathbf{E}_{[p]} + \mathbf{\Lambda}) &= \mathcal{F}(\mathbf{M}^{T-1}\mathbf{o}; \tilde{\mathbf{E}}_{[p]}^T\boldsymbol{\mu}_{[p]}^{(m)}, \mathbf{I}) \\ &\quad -\mathcal{F}(\tilde{\mathbf{E}}_{[p]}\mathbf{M}^{T-1}\mathbf{o}; \tilde{\mathbf{E}}_{[p]}\tilde{\mathbf{E}}_{[p]}^T\boldsymbol{\mu}_{[p]}^{(m)}, \mathbf{D}^{(m)}) \\ &= \mathcal{F}(\mathbf{M}^{T-1}\mathbf{o}; \tilde{\mathbf{E}}_{[p]}^T\boldsymbol{\mu}_{[p]}^{(m)}, \mathbf{I}) \\ &\quad -\mathcal{F}(\mathbf{E}_{[p]}\mathbf{\Lambda}^{-1}\mathbf{o}; \mathbf{E}_{[p]}\mathbf{\Lambda}^{-1}\mathbf{E}_{[p]}^T\boldsymbol{\mu}_{[p]}^{(m)}, \mathbf{D}^{(m)}) \qquad (62)\end{aligned}$$

where

$$\mathbf{D}^{(m)} = \mathbf{L}^{(m)-1}\left(\mathbf{I} + \mathbf{L}^{(m)}\mathbf{E}_{[p]}\mathbf{\Lambda}^{-1}\mathbf{E}_{[p]}^T\mathbf{L}^{(m)T}\right)\mathbf{L}^{(m)T-1} \qquad (63)$$

$$\mathbf{L}^{(m)T}\mathbf{L}^{(m)} = \mathbf{\Sigma}_{\text{diag}[p]}^{(m)} \qquad (64)$$

Thus the calculation of the likelihood may be computed as a combination of

1. a global $n \times n$ matrix vector product ($\mathbf{M}^{-1}\mathbf{o}$) and an n-dimensional Euclidean distance per component.
2. a global $p \times n$ matrix vector product ($\mathbf{E}_{[p]}\mathbf{\Lambda}^{-1}\mathbf{o}$) and a p-dimensional full covariance calculation per component with mean $\mathbf{E}_{[p]}\mathbf{\Lambda}^{-1}\mathbf{E}_{[p]}^T\boldsymbol{\mu}_{[p]}^{(m)}$ and variance $\mathbf{D}^{(m)}$.

In a similar fashion to the multiple HLDA scheme, if multiple transform classes are used, then the two matrix-vector products are performed per transform class rather than globally.

References

[1] H Attias. Independent factor analysis. *Neural Computation*, 11:803–851, 1999.

[2] P Brown. *The Acoustic-Modelling Problem in Automatic Speech Recognition.* PhD thesis, IBM T.J. Watson Research Center, 1987.

[3] N Campbell. Canonical variate analysis - a general formulation. *Australisan Journal of Statistics*, 26:86–96, 1984.

[4] S S Chen and R A Gopinath. Model selection in acoustic modelling. In *Proceedings Eurospeech*, pages 1087–1090, 1999.

[5] 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.

[6] R O Duda and P B Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

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

[8] M J F Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions Speech and Audio Processing*, 7:272–281, 1999.

[9] R Gopinath. Maximum likelihood modeling with Gaussian distributions for classification. In *Proc. ICASSP*, volume II, pages II–661–II–664, 1998.

[10] M J Hunt and C Lefebre. A comparison of several acoustic representations for speech recognition with degraded and undegraded speech. In *Proceedings ICASSP*, pages 262–265, 1989.

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

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

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

[14] C Rathinavalu and L Deng. HMM-based speech recognition using state-dependent discriminatively derived transforms on Mel-warped DFT features. *IEEE Transactions Speech and Audio Processing*, pages 243–256, 1997.

[15] S Roweiss and Z Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345, 1999.

[16] D B Rubin and D T Thayer. EM algorithms for ML factor analysis. *Psychmetrika*, 47:69–76, 1982.

[17] L K Saul and M G Rahim. Maximum likelihood and minimum classification error factor analysis for automatic speech recognition. *IEEE Transactions Speech and Audio Processing*, 8:115–125, 2000.

[18] G Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1973.

[19] M Tipping and C Bishop. Mixtures of probabilistic principle component analyzers. *Neural Computation*, 11:435–474, 1999.

[20] P C Woodland and D Povey. Very large scale MMIE training for conversational telephone speech recognition. In *Proceeding of the 2000 Speech Transcription Workshop*, June 2000. To appear.

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