Maximum Likelihood Multiple Subspace Projections for Hidden Markov Models

Mark J. F. Gales

***Abstract*—The first stage in many pattern recognition tasks is to generate a good set of features from the observed data. Usually, only a single feature space is used. However, in some complex pattern recognition tasks the choice of a good feature space may vary depending on the signal content. An example is in speech recognition where phone dependent feature subspaces may be useful. Handling multiple subspaces while still maintaining meaningful likelihood comparisons between classes is a key issue. This paper describes two new forms of multiple subspace schemes. For both schemes, the problem of handling likelihood consistency between the various subspaces is dealt with by viewing the projection schemes within a maximum likelihood framework. Efficient estimation formulae for the model parameters for both schemes are derived. In addition, the computational cost for their use during recognition are given. These new projection schemes are evaluated on a large vocabulary speech recognition task in terms of performance, speed of likelihood calculation and number of model parameters.**

I. 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 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 feature sets using large vocabulary continuous speech recognition (LVCSR) as the example task. 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. For speech, the particular acoustic realization may be best modeled in different acoustic subspaces depending on whether, for example, a vowel or consonant was generated.

The majority of previous work in projection schemes has concentrated on generating a single "good" feature subspace [1]–[3]. Some techniques may be applied directly using just the observed data, for example schemes based on principal component analysis (PCA). Others require class labels, such as linear discriminant analysis (LDA) [1], [2]. Recently the use of maximum likelihood (ML) estimation has been proposed for generating a linear subspace projection[1] with class labeled data, heteroscedastic LDA (HLDA) [3]. As ML estimation is being used distributions must be specified to span all dimensions of the original feature space to enable consistent parameter optimization. The dimensions of the transformed feature space are split into two distinct groups. 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, are modeled using 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 ensuring that all the feature transforms span the original space there are no problems comparing likelihoods. Subspace projections are obtained by appropriately tying model parameters. Since LVCSR is the target application, hidden Markov models (HMMs) [4] are used as the underlying model with Gaussian mixture models representing each state. Multiple linear subspaces have previously been examined for HMMs [5], [6]. Factor analysis (FA) [6] uses a different subspace for each Gaussian component. It may be viewed as a restricted form of covariance modeling. Though ML estimation for factor analysis has simple re-estimation formulae [7], the likelihood calculations are computationally expensive for LVCSR compared to other restricted covariance modeling schemes [8]. This is further discussed in Section V. In [5], 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 normalization terms associated with feature-space transformations were ignored, both in the optimization schemes and likelihood calculation. For the single transform case trained using discriminative training the normalization terms naturally cancel, but this is not the case for multiple transforms. It must be assumed therefore that the discriminative optimization criterion is restricting the ability

Manuscript received November 8, 1999; revised October 26, 2001. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hsiao-Wuen Hon.

The author was with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA. He is now with the Engineering Department, Cambridge University, Cambridge CB2 1PZ, U.K. (e-mail: mjfg@eng.cam.ac.uk).

Publisher Item Identifier S 1063-6676(02)01521-3.

[1]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.

of the transform normalization 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 [3] 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, including LDA and HLDA, to be described in a consistent fashion. Two new forms of projection are investigated. The first, *multiple HLDA*, is a direct extension to HLDA to allow class-dependent subspaces. In this case there is no global nuisance subspace, as all dimensions contain some class information. The second form, *multiple LDA* (MLDA), generates a common nuisance subspace for all transforms. For likelihood calculations during recognition, these nuisance dimensions may be ignored, reducing the computational cost.

This paper is organized as follows. Section II describes LDA and its extension to HLDA. Section III 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 Section II these models are compared to the linear Gaussian models described in [7]. Finally, experiments on a speaker independent task are presented.

II. Linear Discriminant Analysis

This section describes a standard linear subspace projection scheme, linear discriminant analysis (LDA). Two equivalent forms of optimization 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. A third scheme, heteroscedastic discriminant analysis, is also described. In contrast to the other schemes this is not based on an ML criterion. For all projection schemes in this paper diagonal covariance matrices will be used for the Gaussian components in the final, projected, feature space. This is not a requirement. For example in [3] HLDA is investigated with full-covariance matrices in the projected space. However the vast majority of speech recognition systems use multiple diagonal covariance Gaussian components. Furthermore, all schemes will use each Gaussian component as a separate class.

A. Standard Linear Discriminant Analysis

LDA is a standard linear projection scheme [9]. 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 optimization for LDA is to find the $p \times n$ transform, $\mathbf{A}_{[p]}$, that maximizes [10][2]

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

where $\check{\mathbf{B}}$ and $\check{\mathbf{W}}$ are the between and average within class covariance matrices in the original feature space. These are defined as[3]

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

where

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

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

and

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

In both cases, $\gamma_m(\tau)$ is the posterior probability of Gaussian component m at 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 $\check{\mathbf{W}}^{-1}\check{\mathbf{B}}$ ordered according to decreasing eigenvalues [10].

Rather than expressing the optimization in the form of (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 is required. Therefore, simple truncation of the transform cannot be used. Instead during training the $(n-p)$ nuisance dimensions are modeled by a nondiscriminating model, in this case a single, global, Gaussian distribution. This global distribution in the original feature space has mean $\check{\boldsymbol{\mu}}$, defined previously, and covariance matrix $\check{\boldsymbol{\Sigma}}$, $\check{\boldsymbol{\Sigma}} = \check{\mathbf{B}} + \check{\mathbf{W}}$. A simple two-dimensional (2-D) example is shown in Fig. 1. There are two classes, each shown having the same within class covariance matrix. The original 2-D space is projected to a single dimensional subspace

[2]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 ith 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 submatrices are used they are indicated, for example, by $\mathbf{A}_{[p]}$, this is a $p \times n$ matrix (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 a $p \times p$ 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, $\check{\boldsymbol{\mu}}$

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

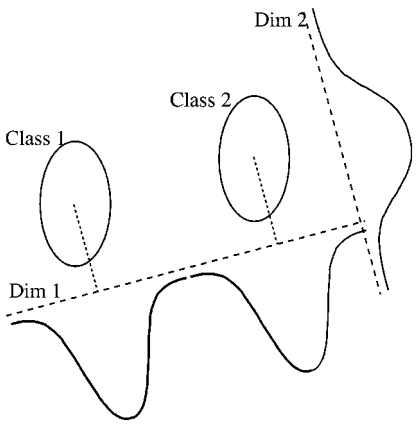

Fig. 1. Linear discriminant analysis.

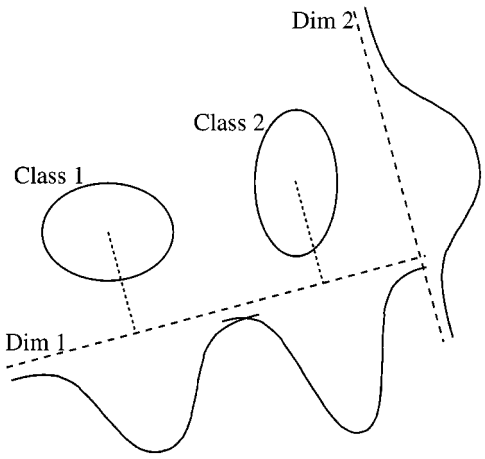

Fig. 2. Heteroscedastic linear discriminant analysis.

($n = 2$, $p = 1$), yielding the useful dimension, *Dim 1*. The nuisance dimension is labeled *Dim 2*. Each class has a separate distribution in *Dim 1*, and there is a single, common, distribution in the nuisance dimension.

Optimizing the likelihood, given the values of $\gamma_m(\tau)$, may be shown to require maximizing (see (6) at the bottom of the page) with respect to $\mathbf{A}$. In [3] it is shown that the right eigenvectors of $\check{\mathbf{W}}^{-1}\check{\mathbf{B}}$ 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 optimizing (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. All solutions will yield the same performance classifier when using continuous density HMMs.

B. Heteroscedastic LDA

Heteroscedastic LDA (HLDA) [3] is an extension to LDA in which the restriction that all within class covariance matrices are the same is removed. This is shown in Fig. 2. Classes 1 and 2 have different within class covariance matrices. Again the 2-D 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 (6) to reflect this added flexibility yields (see (7) at the bottom of the page) where

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

Unfortunately, in contrast to LDA, HLDA has no simple optimization scheme. Standard nonlinear optimization schemes may be used to obtain the parameters of $\mathbf{A}$ [3], [12]. 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 [3] a comparison of LDA and HLDA is performed on a simple digit task. It was found HLDA outperformed LDA. This has also been observed for LVCSR [12].

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, $\boldsymbol{\Sigma}_{\text{diag}[p]}^{(m)}$, for component m are

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

and

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

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

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

The likelihood need only be calculated in the useful subspace specified by the projection matrix. Thus, the cost[4] 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 reduced. In the original space there are $(2Mn)$ parameters compared to $(2Mp+2(n-p)+n^2)$ for the HLDA case.

C. Heteroscedastic Discriminant Analysis

Recently, an alternative interpretation of LDA, which is not based on maximizing the likelihood, has been proposed [13]. Heteroscedastic discriminant analysis (HDA) uses the following objective function to obtain the projection matrix $\mathbf{A}_{[p]}$

$$\mathcal{Q}_{\text{HDA}}\left(\mathbf{A}_{[p]};\mathcal{M}\right) = \frac{1}{2}\sum_{m,\tau}\gamma_m(\tau) \times \log\left(\frac{\left|\det\left(\mathbf{A}_{[p]}\check{\mathbf{B}}\mathbf{A}_{[p]}^T\right)\right|}{\left|\det\left(\mathbf{A}_{[p]}\check{\mathbf{W}}^{(m)}\mathbf{A}_{[p]}^T\right)\right|}\right). \quad (11)$$

In contrast to HLDA, there is no modeling of the $(n-p)$ nuisance dimensions. Only a truncated projection matrix $\mathbf{A}_{[p]}$ is estimated. This expression is closely related to the optimization criterion for LDA given (1). However, as with HLDA, the within-class covariance matrices are not constrained to be the same. This form has the elegant intuitive interpretation of obtaining a feature space that explicitly maximizes the between class variance to within class variance without constraining the within class covariance matrices to be the same. The optimization of this expression requires the use of standard quasi-Newton methods, rather than the simple iterative scheme for HLDA.

Though HDA has an elegant interpretation, it is not appropriate for use with multiple projection schemes. For all multiple projection schemes it must be possible to compare likelihoods across the difference subspaces. This requires that the complete feature space must be modeled in some form (if not necessarily evaluated). As HDA does not model the nuisance dimensions it is not useful for multiple projection schemes. The same problem applies to other nonlikelihood projection schemes such as minimum Bayes feature selection [14]. For this reason, the multiple projection schemes in this paper are based on maximum likelihood schemes.

III. Semi-Tied Covariance Matrices

Using LDA and HLDA only a single subspace can be 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 useful to have multiple feature spaces. One scheme to enable this, though without reducing the feature space size, is semi-tied covariance matrices (STC) [8]. In this form of modeling, each of the classes (in this case Gaussian components) is uniquely assigned[5] 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 which defines the feature space. 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 in the original feature space, $\check{\boldsymbol{\Sigma}}^{(m)}$, may be expressed as

$$\check{\boldsymbol{\Sigma}}^{(m)} = \mathbf{F}^{(r_m)}\boldsymbol{\Sigma}_{\text{diag}}^{(m)}\mathbf{F}^{(r_m)T} \quad (13)$$

where $\boldsymbol{\Sigma}_{\text{diag}}^{(m)}$ is the component-specific diagonal matrix in the transformed space,[6] $\mathbf{F}^{(r_m)}$ 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 modeling. 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 $R \ll M$ there is a dramatic decrease in the number of model parameters. In addition to reducing the number of model parameters, this form of covariance modeling allows efficient likelihood calculation. Using the inverse of the transform matrix, $\mathbf{A}^{(r_m)} = \mathbf{F}^{(r_m)-1}$

$$\begin{aligned}\mathcal{L}\left(\mathbf{o};\check{\boldsymbol{\mu}}^{(m)},\check{\boldsymbol{\Sigma}}^{(m)}\right) &= \mathcal{N}\left(\mathbf{o};\check{\boldsymbol{\mu}}^{(m)},\check{\boldsymbol{\Sigma}}^{(m)}\right)\\ &= \left|\det\left(\mathbf{A}^{(r_m)}\right)\right| \mathcal{N}\times\left(\mathbf{A}^{(r_m)}\mathbf{o};\mathbf{A}^{(r_m)}\check{\boldsymbol{\mu}}^{(m)},\boldsymbol{\Sigma}_{\text{diag}}^{(m)}\right)\\ &= \left|\det\left(\mathbf{A}^{(r_m)}\right)\right| \mathcal{N}\left(\mathbf{A}^{(r_m)}\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}_{\text{diag}}^{(m)}\right)\\ &= \mathcal{L}\left(\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}_{\text{diag}}^{(m)},\mathbf{A}^{(r_m)}\right) \quad (14)\end{aligned}$$

remembering that $\check{\boldsymbol{\mu}}^{(m)}$ and $\boldsymbol{\mu}^{(m)}$ are the component means in the original and transformed spaces respectively. The cost of

[4]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.

[5]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]. In this work, a "hard" assignment of each Gaussian component to a single transformation class is used. An interesting alternative approach is to use a probabilistic assignment. The likelihood of a particular observation $\mathbf{o}$ being generated by state m is given by

$$\mathcal{L}(\mathbf{o}|m) = \sum_r P(r|m)\mathcal{L}\left(\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}_{\text{diag}}^{(m)},\mathbf{A}^{(r)}\right). \quad (12)$$

The optimization of such a model is a straightforward application of a mixture model [15] to the optimization described in this paper. For computational reasons, this form of model is not considered in this paper.

[6]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 nondiagonal cases for HLDA was examined on a small vocabulary task in [3].

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 modeling, $\mathcal{M}$, may be written as[7]

$$\mathcal{M} = \left\{ \left\{ \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(R)} \right\}, \left\{ \boldsymbol{\Sigma}_{\text{diag}}^{(1)}, \ldots, \boldsymbol{\Sigma}_{\text{diag}}^{(M)} \right\}, \left\{ \boldsymbol{\mu}^{(1)}, \ldots, \boldsymbol{\mu}^{(M)} \right\} \right\}. \quad (15)$$

The transform parameters are described in terms of the inverse of the transform matrix because it has been found to yield efficient optimization schemes [8] and simplifies the tying requirements that will be used later in this paper.

There is a close relationship between STC modeling 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 $\boldsymbol{\Sigma}_{\text{diag}}^{(m)}$, are tied to be the same. Now

$$\boldsymbol{\mu}^{(m)} = \begin{bmatrix} \boldsymbol{\mu}_{[p]}^{(m)} \\ \mathbf{A}_{[n-p]} \check{\boldsymbol{\mu}} \end{bmatrix},$$
$$\boldsymbol{\Sigma}_{\text{diag}}^{(m)} = \begin{bmatrix} \boldsymbol{\Sigma}_{\text{diag}[p]}^{(m)} & 0 \\ 0 & \text{diag}\left(\mathbf{A}_{[n-p]} \check{\boldsymbol{\Sigma}} \mathbf{A}_{[n-p]}^T \right) \end{bmatrix}. \quad (16)$$

It is simple to show that this form of tying yields the same objective function as HLDA. For standard LDA an additional level of tying is required where $\boldsymbol{\Sigma}_{\text{diag}}^{(m)} = \boldsymbol{\Sigma}_{\text{diag}}$. The rest of this section details the optimization 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. In common with standard HMM training, an expectation-maximization (EM) [16] approach is used. However, a generalized EM scheme is required since there are no simple closed-form solutions to find $\mathbf{A}^{(r)}$. In generalized 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 optimized with respect to the new model parameters. The auxiliary function is given by

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \frac{1}{2} \sum_{r, m \in M^{(r)}, \tau} \gamma_m(\tau) \times \log \left(\frac{\left| \det\left(\mathbf{A}^{(r)} \right) \right|^2}{\left| \det\left(\text{diag}\left(\mathbf{A}^{(r)} \check{\mathbf{W}}^{(m)} \mathbf{A}^{(r)T} \right) \right) \right|} \right) - \frac{n\beta}{2} (\log(2\pi) + 1) \quad (17)$$

where

$$\beta = \sum_r \beta^{(r)} = \sum_{r, m \in M^{(r)}, \tau} \gamma_m(\tau) \quad (18)$$

and $\gamma_m(\tau)$ is the posterior probability of Gaussian component m a time instance τ given the old model set and all the training data. Although (17) could be directly optimized using nonlinear optimization 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, $\check{\mathbf{W}}^{(m)}$.
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 $\left\{ \tilde{\boldsymbol{\Sigma}}_{\text{diag}}^{(M)} \right\} = \left\{ \boldsymbol{\Sigma}_{\text{diag}}^{(1)}, \ldots, \boldsymbol{\Sigma}_{\text{diag}}^{(M)} \right\}$. This is a standard problem and will only be mentioned briefly.
3) Estimate the new transform $\mathbf{A}^{(r)}$ using the current set $\left\{ \tilde{\boldsymbol{\Sigma}}_{\text{diag}}^{(M)} \right\}$.
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 optimization is preferable to the direct nonlinear optimization as it is computationally efficient. Typically the algorithm converges after four or five iterations, and is guaranteed to be stable [8]. Step (3) is itself an iterative estimation scheme. Rewriting (17) using the fixed set $\left\{ \tilde{\boldsymbol{\Sigma}}_{\text{diag}}^{(M)} \right\}$ (all terms independent of $\mathbf{A}^{(r)}$ are ignored) yields[8]

$$\mathcal{Q}\left(\mathcal{M}, \hat{\mathcal{M}}; \left\{ \tilde{\boldsymbol{\Sigma}}_{\text{diag}}^{(M)} \right\} \right) = \sum_r \left\{ \beta^{(r)} \log \left(\left(\mathbf{c}_i^{(r)} \mathbf{a}_i^{(r)T} \right)^2 \right) - \sum_j \left(\mathbf{a}_j^{(r)} \mathbf{G}^{(rj)} \mathbf{a}_j^{(r)T} \right) \right\} \quad (19)$$

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

$$\mathbf{G}^{(rj)} = \sum_{m \in M^{(r)}} \frac{1}{\sigma_{\text{diag}j}^{(m)2}} \check{\mathbf{W}}^{(m)} \sum_\tau \gamma_m(\tau) \quad (20)$$

where $\sigma_{\text{diag}j}^{(m)2}$ is the jth leading diagonal element of $\boldsymbol{\Sigma}_{\text{diag}}^{(m)}$. It is possible to show that the ML solution for row i is[9]

$$\mathbf{a}_i^{(r)} = \mathbf{c}_i^{(r)} \mathbf{G}^{(ri)-1} \sqrt{\left(\frac{\beta^{(r)}}{\mathbf{c}_i^{(r)} \mathbf{G}^{(ri)-1} \mathbf{c}_i^{(r)T}} \right)}. \quad (21)$$

The optimization scheme is iterative since the estimation of each matrix row is influenced by the cofactors of the complete matrix. 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

[7]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 [4].

[8]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].

[9]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

[17]. In [8], a comparison of STC matrices with state-specific rotations was performed. It was found that STC modeling out-performed the state-specific rotation scheme.

IV. Multiple Subspace Projections

Sections I–III have described the use of LDA and related schemes to perform projections to a single, "optimal," feature space. Then STC systems were described. STC schemes may be viewed as using multiple linear transformations of the feature-space without performing any projections. This section describes how the two forms of modeling may be combined to give multiple subspace projections that can be trained in an ML fashion. The multiple subspace projection schemes will be described in terms of tying the parameters of an STC system.

A. Multiple HLDA

HLDA [3] was described in Section II. Multiple HLDA may be viewed as an extension to HLDA where the classes are partitioned into distinct transform classes. Then a separate HLDA transform is estimated for each of these transform classes. Multiple HLDA is illustrated for a simple case in Fig. 3.

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-D data is projected down to a single useful dimension for each transform class, shown on the diagram as *Dim 1—Transform 1* and *Dim 1—Transform 2*. There are two important things to notice. First, both transforms span the same, original, feature space,[10] thus, when suitably normalized the likelihoods of the transform classes may be directly compared. Second, though the nuisance dimensions for each transform are modeled using a simple single Gaussian distribution, they still contain some transform class-specific information. HLDA is a restricted version of multiple HLDA where R, the number of transformation classes, is restricted to be one.

For multiple HLDA the feature space is split into two subspaces for each transform class; the useful p-dimensional subspace, and an $(n-p)$-dimensional nuisance subspace where a simple single Gaussian component nuisance model is used for that transform class.[11] Thus

$$\mathbf{A}^{(r)} = \begin{bmatrix} \mathbf{A}^{(r)}_{[p]} \\ \mathbf{A}^{(r)}_{[n-p]} \end{bmatrix} \tag{22}$$

and

$$\boldsymbol{\mu}^{(m)} = \begin{bmatrix} \boldsymbol{\mu}^{(m)}_{[p]} \\ \boldsymbol{\mu}^{(r_m)}_{[n-p]} \end{bmatrix}, \quad \boldsymbol{\Sigma}^{(m)}_{\text{diag}} = \begin{bmatrix} \boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}^{(r_m)}_{\text{diag}[n-p]} \end{bmatrix} \tag{23}$$

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

[11]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 [18]. Furthermore, 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. More complicated multiple component noise models may be simply introduced. The nuisance dimensions components are treated as a separate stream [19] to the useful parameters. In terms of the optimization, the change is that $\mathbf{K}^{(rj)}$ in (32) is now a sum over all the nuisance components instead of the single noise Gaussian. This is described in more detail in [18].

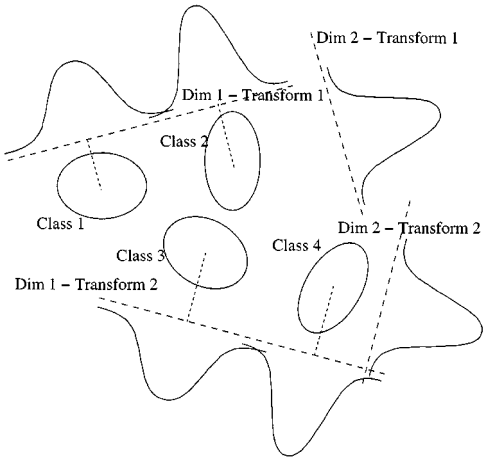

Fig. 3. Multiple HLDA.

where again r_m indicates the transform class that component m belongs to (i.e., $m \in M^{(r_m)}$)

$$\boldsymbol{\mu}^{(r)}_{[n-p]} = \mathbf{A}^{(r)}_{[n-p]} \check{\boldsymbol{\mu}}^{(r)} \tag{24}$$

$$\boldsymbol{\Sigma}^{(r)}_{\text{diag}[n-p]} = \text{diag}\left(\mathbf{A}^{(r)}_{[n-p]} \check{\boldsymbol{\Sigma}}^{(r)} \mathbf{A}^{(r)T}_{[n-p]} \right) \tag{25}$$

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

and

$$\check{\boldsymbol{\mu}}^{(r)} = \frac{\sum_{m \in M^{(r)},\tau} \gamma_m(\tau) \mathbf{o}(\tau)}{\sum_{m \in M^{(r)},\tau} \gamma_m(\tau)}. \tag{27}$$

The likelihood calculation for multiple HLDA is less expensive than that of STC, since certain dimensions of the feature-vector are modeled with simple transform class dependent distributions. The likelihood may be computed as

$$\mathcal{L}\left(\mathbf{o}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}_{\text{diag}}, \mathbf{A}^{(r_m)} \right) = l^{(r_m)} \left| \det\left(\mathbf{A}^{(r_m)} \right) \right| \mathcal{N} \times \left(\mathbf{A}^{(r_m)}_{[p]} \mathbf{o}; \boldsymbol{\mu}^{(m)}_{[p]}, \boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]} \right) \tag{28}$$

where

$$l^{(r)} = \mathcal{N}\left(\mathbf{A}^{(r)}_{[n-p]} \mathbf{o}; \boldsymbol{\mu}^{(r)}_{[n-p]}, \boldsymbol{\Sigma}^{(r)}_{\text{diag}[n-p]} \right). \tag{29}$$

$l^{(r)}$ need only be calculated once for each transform class. Thus the likelihood calculation cost is $\mathcal{O}(Mp + Rn^2 + R(n-p))$. Note that multiple HLDA is not a true projection scheme, since all the dimensions contain some class information. Therefore the likelihoods used during recognition must be computed, though efficiently, in the full n-dimensional space. When $R = 1$, HLDA, the computational cost is even lower. The final $n-p$ dimensions need not be computed since they do not discriminate between the classes (single HLDA is a true projection scheme). Thus,

the "likelihood" may be computed as[12]

$$\mathcal{L}\left(\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}_{\text{diag}},\mathbf{A}\right) \propto |\det(\mathbf{A})|\mathcal{N}\left(\mathbf{A}_{[p]}\mathbf{o};\boldsymbol{\mu}^{(m)}_{[p]},\boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]}\right). \quad (30)$$

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

Optimizing the parameters of the multiple HLDA transform case is a simple modification to the semi-tied transform case. It is possible to rewrite (17) as (ignoring all expressions that are independent of $\mathbf{A}^{(r)}$)

$$\mathcal{Q}\left(\mathcal{M},\hat{\mathcal{M}};\left\{\hat{\boldsymbol{\Sigma}}^{(M)}_{\text{diag}}\right\}\right)=\sum_r\left\{\beta^{(r)}\log\left(\left(\mathbf{c}^{(r)}_i\mathbf{a}^{(r)T}_i\right)^2\right) -\sum_{j\leq p}\left(\mathbf{a}^{(r)}_j\mathbf{G}^{(rj)}\mathbf{a}^{(r)T}_j\right)-\sum_{j>p}\mathbf{a}^{(r)}_j\mathbf{K}^{(rj)}\mathbf{a}^{(r)T}_j\right\} \quad (31)$$

where

$$\mathbf{K}^{(rj)}=\frac{1}{\sigma^{(r)2}_{\text{diag}j}}\check{\boldsymbol{\Sigma}}^{(r)}\sum_{m\in M^{(r)},\tau}\gamma_m(\tau). \quad (32)$$

For rows $i\leq p$, differentiating (31) with respect to $\mathbf{a}^{(r)}_i$ and equating to zero yields the standard STC re-estimation formula given in (21). For rows $i>p$ following the same procedure gives

$$\mathbf{a}^{(r)}_i=\mathbf{c}^{(r)}_i\mathbf{K}^{(ri)-1}\sqrt{\left(\frac{\beta^{(r)}}{\mathbf{c}^{(r)}_i\mathbf{K}^{(ri)-1}\mathbf{c}^{(r)T}_i}\right)}. \quad (33)$$

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

B. Multiple LDA

In Section IV-A, all the dimensions obtained using multiple HLDA contain some class information. This section describes the additional tying of the STC system required to generate a global nuisance subspace, while using multiple transformations. To generate the global nuisance subspace all parameters associated with the the final $n-p$, the nuisance dimensions, are tied. Thus

$$\mathbf{A}^{(r)}=\begin{bmatrix}\mathbf{A}^{(r)}_{[p]}\\ \mathbf{A}_{[n-p]}\end{bmatrix} \quad (34)$$

and

$$\boldsymbol{\mu}^{(m)}=\begin{bmatrix}\boldsymbol{\mu}^{(m)}_{[p]}\\ \mathbf{A}_{[n-p]}\check{\boldsymbol{\mu}}\end{bmatrix},$$
$$\boldsymbol{\Sigma}^{(m)}_{\text{diag}}=\begin{bmatrix}\boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]} & \mathbf{0}\\ \mathbf{0} & \text{diag}\left(\mathbf{A}_{[n-p]}\check{\boldsymbol{\Sigma}}\mathbf{A}^T_{[n-p]}\right)\end{bmatrix}. \quad (35)$$

This form of transform will be called multiple LDA (MLDA).[13]

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

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

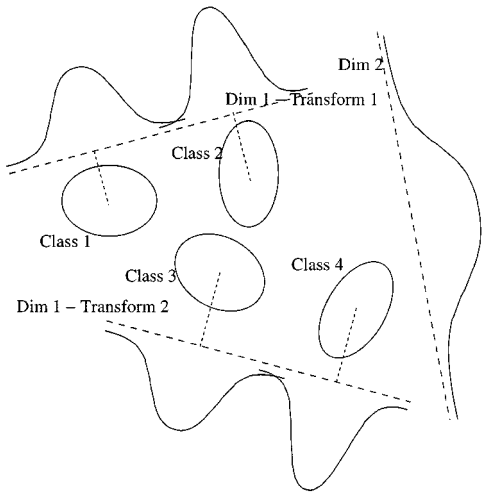

Fig. 4. Multiple linear discriminant analysis.

A simple example of MLDA is shown in Fig. 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 nuisance dimension, common to both transform classes, and all the data in *Dim 2* is modeled by a single Gaussian distribution. There are two different projections for the useful dimension, labeled *Dim 1—Transform 1* and *Dim 1—Transform 2*. In both cases, the nuisance and useful dimensions span the original feature space.[14] Thus, when appropriately normalized, likelihoods in both transformed spaces may be directly compared. However now the nuisance parameters, the distribution in *Dim 2* no longer achieve any class discrimination. As the nuisance dimensions do not discriminate between the classes it is unnecessary to use them during recognition. The nuisance dimensions have been projected out. Hence, the "likelihood" may be calculated as

$$\mathcal{L}\left(\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}^{(m)}_{\text{diag}},\mathbf{A}^{(r_m)}\right) \propto \left|\det\left(\mathbf{A}^{(r_m)}\right)\right|\mathcal{N}\left(\mathbf{A}^{(r_m)}_{[p]}\mathbf{o};\boldsymbol{\mu}^{(m)}_{[p]},\boldsymbol{\Sigma}^{(m)}_{\text{diag}[p]}\right). \quad (36)$$

For example, in Fig. 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.

Optimizing 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 optimize the STC parameters is used. The additional complication 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.

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

The auxiliary function in this case may be written as

$$\mathcal{Q}\left(\mathcal{M},\hat{\mathcal{M}};\left\{\tilde{\boldsymbol{\Sigma}}_{\text{diag}}^{(M)}\right\}\right)=\sum_r\left\{\beta^{(r)}\log\left(\left(\mathbf{c}_i^{(r)}\mathbf{a}_i^{(r)T}\right)^2\right)-\sum_{j\leq p}\left(\mathbf{a}_j^{(r)}\mathbf{G}^{(rj)}\mathbf{a}_j^{(r)T}\right)\right\}-\sum_{j>p}\mathbf{a}_j\mathbf{K}^{(j)}\mathbf{a}_j^T \quad (37)$$

where $\mathbf{G}^{(rj)}$ is defined in (20) and

$$\mathbf{K}^{(j)}=\frac{1}{\sigma_{\text{diag}j}^2}\check{\boldsymbol{\Sigma}}\sum_{m,\tau}\gamma_m(\tau). \quad (38)$$

Differentiating this with respect to $\mathbf{a}_i^{(r)}$ and equating to zero yields the standard STC estimation formula given in (21) for rows $i\leq p$. However, 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 (37) may be rewritten as

$$\mathcal{Q}\left(\mathcal{M},\hat{\mathcal{M}};\left\{\tilde{\boldsymbol{\Sigma}}_{\text{diag}}^{(M)}\right\}\right)=\sum_r\left\{\beta^{(r)}\log\left(\left(\mathbf{c}_i^{(r)}\mathbf{a}_i^{T}\right)^2\right)-\sum_{j\leq p}\left(\mathbf{a}_j^{(r)}\mathbf{G}^{(rj)}\mathbf{a}_j^{(r)T}\right)\right\}-\sum_{j>p}\mathbf{a}_j\mathbf{K}^{(j)}\mathbf{a}_j^T. \quad (39)$$

Now, differentiating with respect to $\mathbf{a}_i$ (note $i>p$) yields

$$\mathbf{f}^{(i)}=\frac{\partial\mathcal{Q}\left(\mathcal{M},\hat{\mathcal{M}};\left\{\tilde{\boldsymbol{\Sigma}}_{\text{diag}}^{(M)}\right\}\right)}{\partial\mathbf{a}_i}=2\sum_r\frac{\beta^{(r)}\mathbf{c}_i^{(r)}}{\mathbf{c}_i^{(r)}\mathbf{a}_i^T}-2\mathbf{a}_i\mathbf{K}^{(i)}. \quad (40)$$

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 optimization. Assuming that $\mathbf{K}^{(i)}$ is of full rank then it is clear that at the ML solution (i.e., $\mathbf{f}^{(i)}=\mathbf{0}$)

$$\mathbf{a}_i=\sum_r\lambda_r^{(i)}\mathbf{c}_i^{(r)}\mathbf{K}^{(i)-1}. \quad (41)$$

Now, the problem is to optimize the likelihood with respect to $\boldsymbol{\lambda}^{(i)}$, an R-dimensional vector. An n-dimensional optimization problem, finding $\mathbf{a}_i$ directly, has been transformed into an R-dimensional problem of finding $\boldsymbol{\lambda}^{(i)}$. Optimization for this particular version follows the same general scheme as that described below. For the special case when $R=1$ equating (40) to zero gives the standard HLDA re-estimation formulae.

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

$$\mathbf{H}^{(i)}=\frac{\partial^2\mathcal{Q}\left(\mathcal{M},\hat{\mathcal{M}};\left\{\tilde{\boldsymbol{\Sigma}}_{\text{diag}}^{(M)}\right\}\right)}{\partial\mathbf{a}_i^2}=-2\sum_r\frac{\beta^{(r)}\mathbf{c}_i^{(r)T}\mathbf{c}_i^{(r)}}{\left(\mathbf{c}_i^{(r)}\mathbf{a}_i^T\right)^2}-2\mathbf{K}^{(i)}. \quad (42)$$

The calculation of the Hessian is efficient. The cost is dominated by the calculation of the cofactors which must be calculated for the gradient, (40). As expected this expression is semi-negative definite indicating a concave error surface.[15] The update formula becomes

$$\mathbf{a}_i=\hat{\mathbf{a}}_i-\Delta\mathbf{f}^{(i)}\mathbf{H}^{(i)-1} \quad (43)$$

where Δ may be 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. However, this scheme was found to be stable with a fixed value of Δ, $\Delta=1$. The total number of outer iterations, model updates, was about the same 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. The overall transform for transform class r may be written as

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

and

$$\mathbf{A}_{[n-p]}=\mathbf{A}_{\mathbf{hlda}[n-p]} \quad (45)$$

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 has two distinct stages, whereas MLDA does a simultaneous optimization. This two-stage MLDA computational will be referred to as MLDA(2). The likelihood of this scheme may be efficiently computed as

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

where

$$\tilde{\mathbf{o}}_{[p]}=\mathbf{A}_{\mathbf{hlda}[p]}\mathbf{o}. \quad (47)$$

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 (46) may be used for MLDA and the effective number of free parameters reduced accordingly. However, the optimization 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.

V. Linear Gaussian Models

The schemes described up to this point should be compared to the use of subspace modeling of signals in other areas of machine learning. Based on linear Gaussian models, they have been used for low-dimensional visualization [20] and determination of independent hidden sources [21]. Since the speed of computing the likelihood is not as important in these tasks as for the density modeling described in this paper, different forms of

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

generative model have been used. These differences have considerable impact in both training and computing the likelihoods of such schemes.

Low dimensional modeling schemes based on ML linear subspaces have previously been proposed. These include schemes such as factor analysis (FA) [22] and probabilistic PCA (PPCA) [20]. In [7], a general framework for linear Gaussian models is given. For the case of static data modeling

$$\mathbf{z}(\tau) = \mathbf{w} \tag{48}$$

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

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

LDA and the other schemes described in Sections I–IV 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) [21]. Second there are restrictions on the form of the sensor noise. The generative model for LDA requires a modification to (49), so that

$$\mathbf{o}(\tau) = \mathbf{E}\begin{bmatrix}\mathbf{z}(\tau)\\ \mathbf{v}\end{bmatrix} \tag{50}$$

where $\mathbf{v}$ 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 [7] and IFA [21][16] may be trained using EM. None of the subspace projection schemes discussed in this paper may be trained in this fashion, they require nonlinear optimization 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 $\mathcal{O}(M(n+p^2)+np+n^2)$ [18]. 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, M, can be tens of thousands and the number of dimensions, n, typically around 40 the difference for any reasonable number of factors is large.

VI. 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 quiet environment with the same microphone. The training data consists of around 300 000 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 41 268 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.[17] 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 the performance using a standard LDA derived frontend was around 10% worse than the use of a single STC transform [8], [12]. Thus, the baseline system considered was the single global STC system. For the experiments where a projection scheme was used, the nuisance dimension size was set to be ten. No attempt was made to optimize this value, though various complexity schemes, such as BIC [23] could be used to determine the optimal size. In addition the number of transform classes was fixed at the number of states, 3430. 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 64 000 word vocabulary.

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

Table II shows the recognition performance of various systems using either global or state level tying of the transformation matrices. For HLDA the last ten elements of the feature-vector, the nuisance dimensions, were modeled with a single Gaussian component per transform class. For MLDA the nuisance dimensions were modeled using a single Gaussian component for all transform classes. For the global case the STC system outperformed 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, multiple HLDA, 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 ten of the dimensions had no effect on the performance. Performing

[16]This is not true when noiseless IFA is trained. In this case similar optimization schemes to STC may be used.

[17]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.

TABLE I
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

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 II
PERFORMANCE OF A STC, HLDA, MLDA, AND MLDA(2) SYSTEMS USING EITHER A GLOBAL OR STATE LEVEL TRANSFORMATION TYING

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

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 II 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 optimization 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 II 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 four. 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.

VII. 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 there was little difference in performance compared to the full system. This indicates that there are possible gains, in terms of both speed and model size, in having flexible model complexities over the dimensions. It was also found that simultaneously optimizing 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 [5], [6]. 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 [24]. All these aspects will be addressed in future work.

REFERENCES

[1] M. J. Hunt and C. Lefèbre, "A comparison of several acoustic representations for speech recognition with degraded and undegraded speech," in *Proc. ICASSP*, 1989, pp. 262–265.
[2] P. Brown, "The acoustic-modeling problem in automatic speech recognition," Ph.D. dissertation, IBM T. J. Watson Res. Center, Yorktown Heights, NY, 1987.
[3] N. Kumar, "Investigation of silicon-auditory models and generalization of linear discriminant analysis for improved speech recognition," Ph.D. dissertation, Johns Hopkins Univ., Baltimore, MD, 1997.
[4] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, Feb. 1989.
[5] C. Rathinavalu and L. Deng, "HMM-based speech recognition using state-dependent discriminatively derived transforms on Mel-warped DFT features," *IEEE Trans. Speech Audio Processing*, pp. 243–256, May 1997.
[6] L. K. Saul and M. G. Rahim, "Maximum likelihood and minimum classification error factor analysis for automatic speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 115–125, Mar. 2000.
[7] S. Roweiss and Z. Ghahramani, "A unifying review of linear Gaussian models," *Neural Comput.*, vol. 11, pp. 305–345, 1999.
[8] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 272–281, 1999.

[9] R. O. Duda and P. B. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[10] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1972.
[11] N. Campbell, "Canonical variate analysis—A general formulation," *Aus. J. Statist.*, vol. 26, pp. 86–96, 1984.
[12] R. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *Proc. ICASSP*, vol. II, 1998, pp. II-661–II-664.
[13] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximimum likelihood discriminant feature spaces," in *Proc. ICASSP*, 2000.
[14] G. Saon and M. Padmanabhan, "Minimum Bayes error feature selection for continuous speech recognition," in *Proc. NIPS*, 2000.
[15] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.
[16] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc.*, vol. 39, pp. 1–38, 1977.
[17] A. Ljolje, "The importance of cepstral parameter correlations in speech recognition," *Comput. Speech Lang.*, vol. 8, pp. 223–232, 1994.
[18] M. J. F. Gales, "Maximum likelihood projection schemes for hidden Markov models," Cambridge Univ., Cambridge, U.K., CUED/F-INFENG/TR365; http://svr-www.eng.cam.ac.uk/~mjfg, 1999.
[19] S. J. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland, *The HTK Book (for HTK Version 2.0)*. Cambridge, U.K.: Cambridge Univ., 1996.
[20] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, pp. 435–474, 1999.
[21] H. Attias, "Independent factor analysis," *Neural Comput.*, vol. 11, pp. 803–851, 1999.
[22] D. B. Rubin and D. T. Thayer, "EM algorithms for ML factor analysis," *Psychometrika*, vol. 47, pp. 69–76, 1982.
[23] G. Schwarz, "Estimating the dimension of a model," in *Ann. Statist.*, 1973, vol. 6, pp. 461–464.
[24] P. C. Woodland and D. Povey, "Very large scale MMIE training for conversational telephone speech recognition," in *Proc. 2000 Speech Transcription Workshop*, June 2000.

Mark J. F. Gales received the B.A. and Ph.D. degrees in electrical and information sciences from the University of Cambridge, Cambridge, U.K., in 1988 and 1996, respectively.

He was a Consultant with Roke Manor Research, Ltd., Hampshire, U.K. In 1991, he became a Research Assistant with the Speech, Vision, and Robotics Group of the Engineering Department, Cambridge University. From 1995 to 1997, he was a Research Fellow with Emmanuel College, Cambridge. He was then a Research Staff Member in the Speech Group, IBM T. J. Watson Research Center, Yorktown Heights, NY, until 1999. He is currently a University Lecturer with the Engineering Department, Cambridge University, and a Fellow of Emmanuel College.

Dr. Gales received the 1997 IEEE Young Author Paper Award in the Speech Processing Area.