Minimum Phone Error Training of Precision Matrix Models

Khe Chai Sim and M.J.F. Gales

***Abstract*— Gaussian Mixture Models (GMMs) are commonly used as the output density function for large vocabulary continuous speech recognition (LVCSR) systems. A standard problem when using multivariate GMMs to classify data is how to accurately represent the correlation in the feature vector. Full covariance matrices yield a good model, but dramatically increase the number of model parameters. Hence diagonal covariance matrices are commonly used. Structured precision matrix approximations provide an alternative, flexible and compact representation. Schemes in this category include the extended maximum likelihood linear transform and subspace for precision and mean models. This paper examines how these precision matrix models can be discriminatively trained and used on state-of-the-art speech recognition tasks. In particular the use of the minimum phone error criterion is investigated. Implementation issues associated with building LVCSR systems are also addressed. These models are evaluated and compared using large vocabulary continuous telephone speech (CTS) and broadcast news (BN) English tasks.**

***Index Terms*— precision matrix modelling, minimum phone error, discriminative training, speech recognition.**

I. Introduction

STATE-OF-THE-ART speech recognition systems are typically based on continuous density hidden Markov models [1] with Gaussian mixture models (GMMs) representing the output distribution associated with each state. A standard problem when using multivariate GMMs to classify data is how to accurately model the correlations in the feature vector. The use of a full covariance matrix for each Gaussian component dominates the total number of model parameters and dramatically increases the computational cost to train and perform recognition with these models. Furthermore, a large amount of training data is required to ensure robust model estimation. For these reasons, more compact and efficient correlation modelling techniques are required, particularly for a large vocabulary continuous speech recognition (LVCSR) [2] system, which comprises many Gaussian components (typically greater than 100,000) and high dimensional data (typically 39 or 52). The conventional approach to addressing these problems is to use a diagonal covariance matrix approximation. The feature dimensions are assumed to be uncorrelated given a particular component. Several methods have been employed to improve the validity of this assumption. For example, the use of Mel frequency Cepstral coefficients (MFCC) [3] and perceptual linear prediction (PLP) [4] coefficients provide data with low correlation. Further decorrelation can be achieved using feature transformation techniques such as linear discriminant analysis (LDA) [5], heteroscedastic LDA (HLDA) [6] and heteroscedastic discriminant analysis (HDA) [7].

Recently, more advanced covariance modelling techniques have been found to give improvements over the feature decorrleating schemes above. Techniques that approximate the inverse covariance (precision) matrices are commonly used. This is more efficient than modelling the covariance matrix, as it eliminates the need to invert the covariance matrices as required by schemes such as the factor-analysed HMMs (FAHMMs) [8]. This yields efficient likelihood computation for precision matrix models. Examples of these models are the semi-tied covariance (STC) [9], extended MLLT (EMLLT) [10] and subspace for precision and mean (SPAM) [11] models. These models have been successfully applied to LVCSR systems using the Maximum Likelihood (ML) training scheme [9], [12], [13].

For many years, ML estimation has been the standard approach to train the HMMs for speech recognition. However, discriminative training has been found to yield promising gain over the ML training on diagonal covariance matrix systems [14], [15]. This has motivated the use of discriminative training for many state-of-the-art LVCSR systems [16]. The STC [17] and SPAM [18] models have previously been discriminatively trained using the Maximum Mutual Information (MMI) criterion on small and medium vocabulary systems. An alternative discriminative training criterion, Minimum Phone Error (MPE), has been found to consistently outperform MMI training on large vocabulary diagonal covariance matrix systems [15]. This paper investigates the use of MPE trained precision matrix models for LVCSR systems. The MPE training approach adopted in this paper is based on the optimisation of the *weak-sense* auxiliary function with I-smoothing, as presented in [15]. Implementation issues regarding building LVCSR systems with precision matrix models will also be discussed.

This paper is organised as follows: Section II describes a generic framework of basis superposition [19], [20] which subsumes various forms of precision matrix modelling techniques. Next, discriminative training of precision matrix models based on the MPE criterion will be discussed in Section III. Section V then addresses the implementation issues of these precision matrix models for LVCSR systems. Experimental results on CTS and BN English tasks are presented in Section VI.

II. Precision Matrix Modelling

Compact precision matrix modelling has been found to yield good gains over the diagonal covariance matrix approximation for GMM covariance modelling. The generic framework of basis superposition [20] may be used as a convenient way of analysing various forms of precision matrix models, such as

the STC, EMLLT and SPAM models. Within this framework, the precision matrix, $\boldsymbol{P}_m$, is given by the following general expression

$$\boldsymbol{P}_m = \sum_{i=1}^{n} \lambda_{ii}^{(m)} \boldsymbol{S}_i = \sum_{i=1}^{n} \lambda_{ii}^{(m)} \sum_{r=1}^{R} \lambda_{rr} \boldsymbol{a}_{ir}' \boldsymbol{a}_{ir} \quad (1)$$

where n is the number of basis (basis order), $\boldsymbol{S}_i$ are a set of symmetric *basis matrices* and $\lambda_{ii}^{(m)}$ are the corresponding superposition coefficients for component m. The basis matrices, $\boldsymbol{S}_i$, can be further decomposed into a linear combination of R *basis row vectors*, $\boldsymbol{a}_{ir}$ weighted by λ_{rr} and R denotes the rank of $\boldsymbol{S}_i$. If $R = 1$, the precision matrix in equation (1) becomes a STC [9] when $n = d$ and an EMLLT [10] model when $d < n \leq \frac{d}{2}(d+1)$, where d is the feature dimensionality. Alternatively, a SPAM [18] model may be modelled with $R = d$. In this case, provided one of the $\boldsymbol{S}_i$ is positive-definite, n is allowed to be as small as 1. Furthermore, setting $R < d$ yields the Hybrid-EMLLT model [21]. Due to the parameterisation of basis superposition into the *global* (basis vectors/matrices) and *component* (basis coefficients) parameters, compact model representation may be achieved via sharing of the basis vectors or matrices. The PMM-HLDA model [19] employs tying of the basis coefficients, which further reduces the number of model parameters.

One of the attractive attributes of precision matrix modelling is its efficiency during decoding. This can be seen clearly from the likelihood expression given by

$$\mathcal{L}(\boldsymbol{\mu}_m, \boldsymbol{P}_m|\mathcal{O}) = K + \frac{1}{2}\sum_{t=1}^{T}\Big\{\log|\boldsymbol{P}_m| - \boldsymbol{x}_{mt}'\boldsymbol{P}_m\boldsymbol{x}_{mt}\Big\} \quad (2)$$

where $\mathcal{L}(\boldsymbol{\mu}_m, \boldsymbol{P}_m|\mathcal{O})$ is the likelihood of the model parameters given the complete set of observations, $\mathcal{O}$ and $\boldsymbol{x}_{mt} = (\boldsymbol{o}_t - \boldsymbol{\mu}_m)$. Modelling the precision matrix, $\boldsymbol{P}_m$, as a superposition of basis eliminates the need to invert the covariance matrix when computing the likelihood. Furthermore, it is shown in [20] that the terms in equation (2) can be divided into *model* and *observation* dependent. The former can be precomputed and cached once the model parameters are loaded. The latter can be cached for each observation and then reused for all the Gaussian components. This yields a significantly cheaper computational cost, which is linearly proportional to the basis order, n.

Maximum likelihood estimation (MLE) is a standard approach to finding model parameters. Within the HMM framework, this is commonly optimised using the well-known Baum-Welch (or more generally Expectation Maximisation) [22] algorithm. The auxiliary function to be maximised in the M-step is given by

$$\mathcal{Q}^{\mathtt{ml}}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = K + \frac{1}{2}\sum_{m=1}^{M}\beta_m\Big\{\log|\boldsymbol{P}_m| - \mathrm{Tr}(\boldsymbol{P}_m\mathbf{W}_m^{\mathtt{ml}})\Big\} \quad (3)$$

where

$$\mathrm{Tr}(\boldsymbol{P}_m\mathbf{W}_m^{\mathtt{ml}}) = \frac{\sum_{t=1}^{T}\gamma_m^{\mathtt{ml}}(t)(\boldsymbol{o}_t - \boldsymbol{\mu}_m)'\boldsymbol{P}_m(\boldsymbol{o}_t - \boldsymbol{\mu}_m)}{\beta_m} \quad (4)$$

T is the total number of frames, $\gamma_m^{\mathtt{ml}}(t)$ is the probability of component m at time t given the current parameter set, $\hat{\boldsymbol{\theta}}$, and K subsumes terms independent of the model parameters. $\boldsymbol{\theta}$ denotes the set of new parameter. The required statistics for the estimation of precision matrix parameters are given by

$$\mathbf{W}_m^{\mathtt{ml}} = \frac{\sum_{t=1}^{T}\gamma_m^{\mathtt{ml}}(t)(\boldsymbol{o}_t - \boldsymbol{\mu}_m)(\boldsymbol{o}_t - \boldsymbol{\mu}_m)'}{\beta_m^{\mathtt{ml}}} \quad (5)$$

$$\beta_m^{\mathtt{ml}} = \sum_{t=1}^{T}\gamma_m^{\mathtt{ml}}(t) \quad (6)$$

$\boldsymbol{W}_m^{\mathtt{ml}}$ and $\beta_m^{\mathtt{ml}}$ are the ML full covariance statistics and the component occupancy counts respectively. For all the forms of precision matrix modelling, the mean vectors are unconstrained. Thus, the following standard update formula may be used

$$\boldsymbol{\mu}_m = \frac{1}{\beta_m^{\mathtt{ml}}}\sum_{t=1}^{T}\gamma_m^{\mathtt{ml}}(t)\boldsymbol{o}_t \quad (7)$$

The ML update formulae for various precision matrix models are summarised in [20]. Further details regarding these models may also be obtained from the corresponding literatures ([9], [10], [18], [19]).

III. Minimum Phone Error (MPE) Training

Recently, discriminative training has been found to yield improved performance in LVCSR compared to the conventional ML training [15]. Various forms of discriminative objective functions have been described in these literatures, for example Maximum Mutual Information (MMI), Minimum Phone Error (MPE) and Minimum Word Error (MWE) criteria [14], [15]. Several forms of MMI trained precision matrix models have recently been published. *Goel et al.*, 2003 [18] presented the MMI estimation of the SPAM models with small vocabulary system. *McDonough et al.* [17] also employed MMI trained STC models in speaker-adapted training (SAT). *Tsakalidis et al.* [23] also introduced Discriminative Likelihood Linear Transform (DLLT), a variant of MLLT whose parameters estimation is also based on the MMI criterion. The consistent improvement of MPE training on large scale diagonal covariance matrix systems compared to the MMI discriminative criterion [15] motivates the investigation of MPE training of precision matrix models on LVCSR systems.

A. Maximising the MPE Objective Function

MPE training aims to minimise the phone classification error (or maximising the phone accuracy). The objective function to be maximised by the MPE training, $\mathcal{R}_{\mathrm{MPE}}(\boldsymbol{\theta})$, may be expressed as

$$\mathcal{R}_{\mathrm{MPE}}(\boldsymbol{\theta}) = \sum_{r}\frac{\sum_s p_\theta(\mathcal{O}_r|s)^\kappa P(s)\mathrm{PhoneAcc}(s, s_r)}{\sum_u p_\theta(\mathcal{O}_r|u)^\kappa P(u)} \quad (8)$$

where $\mathcal{O}_r$ is the rth training sentence and $P(s)$ is the language model probability for sentence s. κ is an acoustic de-weighting factor, which can be adjusted to improve the test-set performance. $\mathrm{PhoneAcc}(s, s_r)$ represent the raw *phone* accuracies of the sentence s given the correct sentence s_r.

As with the ML objective function, the MPE objective function is difficult to optimise directly. In this paper, MPE

training of the precision matrix models is based on the approach presented by *Povey et al.* [15]. The MPE objective function (8) is using an auxiliary function of the form[1]

$$\mathcal{Q}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) = \mathcal{Q}^{\mathtt{n}}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) - \mathcal{Q}^{\mathtt{d}}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) + \mathcal{F}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) \quad (9)$$

where $\mathcal{Q}^{\mathtt{n}}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}})$ and $\mathcal{Q}^{\mathtt{d}}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}})$ are the auxiliary functions for the numerator and denominator terms respectively in the objective function. They differ from the ML auxiliary frunction in that the "posterior" is no longer based on $\gamma_m^{\mathtt{ml}}(t)$, but the numerator and denominator counts, $\gamma_m^{\mathtt{n}}(t)$ and $\gamma_m^{\mathtt{d}}(t)$ respectively. $\mathcal{F}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}})$ is a smoothing function which, as suggested in [15], takes the form

$$\mathcal{F}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) = K + \frac{1}{2}\sum_{m=1}^{M} D_m\Big\{\log|\boldsymbol{P}_m| - \mathrm{Tr}(\boldsymbol{P}_m\hat{\boldsymbol{\Sigma}}_m)\Big\} \quad (10)$$

where $\hat{\boldsymbol{\Sigma}}_m$ is the current estimate of the full covariance matrix and D_m is a component-dependent constant that controls the amount of $\hat{\boldsymbol{\Sigma}}_m$ to be smoothed onto the covariance statistics. Equation (9) is referred to as the *weak-sense* auxiliary function in [15] because an increase in this function does not guarantee an increase in the objective function. In the following, the sufficient statistics required to optimise this weak-sense auxiliary function will be discussed and model parameter update formulae for the EMLLT and SPAM models will be given.

B. Sufficient Statistics for MPE Training

The full ML covariance statistics, $\mathbf{W}_m^{\mathtt{ml}}$, can be rewritten in terms of the sufficient statistics such that

$$\mathbf{W}_m^{\mathtt{ml}} = \frac{\left(\mathbf{Y}_m^{\mathtt{ml}} - \mathbf{x}_m^{\mathtt{ml}}\boldsymbol{\mu}_m^{\mathtt{ml}'} - \boldsymbol{\mu}_m^{\mathtt{ml}}\mathbf{x}_m^{\mathtt{ml}'} + \beta_m^{\mathtt{ml}}\boldsymbol{\mu}_m^{\mathtt{ml}}\boldsymbol{\mu}_m^{\mathtt{ml}'}\right)}{\beta_m^{\mathtt{ml}}} \quad (11)$$

where the sufficient statistics, $\boldsymbol{\Theta}^{\mathtt{ml}} = \{\beta_m^{\mathtt{ml}}, \mathbf{x}_m^{\mathtt{ml}}, \mathbf{Y}_m^{\mathtt{ml}}\}$ for all components m, are given by equation 6,

$$\mathbf{x}_m^{\mathtt{ml}} = \sum_{t=1}^{T}\gamma_m^{\mathtt{ml}}(t)\boldsymbol{o}_t \quad (12)$$

$$\mathbf{Y}_m^{\mathtt{ml}} = \sum_{t=1}^{T}\gamma_m^{\mathtt{ml}}(t)\boldsymbol{o}_t\boldsymbol{o}_t' \quad (13)$$

Given the set of parameters, $\boldsymbol{\theta}$, the ML auxiliary frunction 3 can be rewritten in terms of the ML statistics $\boldsymbol{\Theta}^{\mathtt{ml}}$

$$\mathcal{Q}^{\mathtt{ml}}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) = \mathcal{G}(\boldsymbol{\Theta}^{\mathtt{ml}}) \quad (14)$$

where

$$\mathcal{G}(\boldsymbol{\Theta}^{\mathtt{ml}}) = K + \frac{1}{2}\sum_{m=1}^{M}\beta_m\left\{\log|\boldsymbol{P}_m| - \mathrm{Tr}\left(\boldsymbol{P}_m\boldsymbol{W}_m^{\mathtt{ml}}\right)\right\} \quad (15)$$

Equation (9) can also be expressed in terms of sufficient statistics

$$\mathcal{G}(\boldsymbol{\Theta}^{\mathtt{mpe}}) = \mathcal{G}(\boldsymbol{\Theta}^{\mathtt{n}}) - \mathcal{G}(\boldsymbol{\Theta}^{\mathtt{d}}) + \mathcal{G}(\boldsymbol{\Theta}^{\mathtt{sm}}) \quad (16)$$

where $\boldsymbol{\Theta}^{\mathtt{n}}$ and $\boldsymbol{\Theta}^{\mathtt{d}}$ denote the sufficient statistics for numerator and denominator respectively. The set of parameters, $\boldsymbol{\Theta}^{(\mathtt{sm})}$, which correspond to the smoothing function (10), $\mathcal{F}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) = \mathcal{G}(\boldsymbol{\Theta}^{(\mathtt{sm})})$, are given by

$$\beta_m^{\mathtt{sm}} = D_m \quad (17)$$

$$\mathbf{x}_m^{\mathtt{sm}} = D_m\hat{\boldsymbol{\mu}}_m \quad (18)$$

$$\mathbf{Y}_m^{\mathtt{sm}} = D_m(\hat{\boldsymbol{\Sigma}}_m + \hat{\boldsymbol{\mu}}_m\hat{\boldsymbol{\mu}}_m') \quad (19)$$

Maximising this auxiliary function with respect to the mean vector and covariance matrix parameters yields the following update formulae

$$\boldsymbol{\mu}_m = \frac{\mathbf{x}_m^{\mathtt{n}} - \mathbf{x}_m^{\mathtt{d}} + D\hat{\boldsymbol{\mu}}_m}{\beta_m^{\mathtt{n}} - \beta_m^{\mathtt{d}} + D_m} \quad (20)$$

$$\mathbf{W}_m^{\mathtt{mpe}} = \frac{\mathbf{Y}_m^{\mathtt{n}} - \mathbf{Y}_m^{\mathtt{d}} + D_m(\hat{\boldsymbol{\Sigma}}_m + \hat{\boldsymbol{\mu}}_m\hat{\boldsymbol{\mu}}_m')}{\beta_m^{\mathtt{n}} - \beta_m^{\mathtt{d}} + D_m} - \boldsymbol{\mu}_m\boldsymbol{\mu}_m' \quad (21)$$

It is also possible to consider a set of combined statistics where

$$\boldsymbol{\Theta}^{\mathtt{c}} = \boldsymbol{\Theta}^{\mathtt{n}} - \boldsymbol{\Theta}^{\mathtt{d}} \quad (22)$$

where this set "-" operator yields $\beta_m^{\mathtt{c}} = \beta_m^{\mathtt{n}} - \beta_m^{\mathtt{d}}$ and similarly for $\mathbf{Y}_m^{\mathtt{c}}$ and $\mathbf{x}_m^{\mathtt{c}}$. Using this concept of functions over statistics it is simple to incorporate smoothing techniques such as I-smoothing [15] and Maximum a-Posteriori (MAP) [26] smoothing. To ensure that the auxiliary function is valid, $\mathbf{W}_m$ is required to be positive-definite. Combining equations (20) and (21) gives the full covariance statistics in terms of D_m,

$$\mathbf{W}_m^{\mathtt{mpe}} = \frac{\boldsymbol{B}_2 D_m^2 + \boldsymbol{B}_1 D_m + \boldsymbol{B}_0}{\beta_m^{(c)} + D_m} \quad (23)$$

where

$$\boldsymbol{B}_2 = \hat{\boldsymbol{\Sigma}}_m \quad (24)$$

$$\boldsymbol{B}_1 = \mathbf{Y}_m^{\mathtt{c}} + \beta_m^{\mathtt{c}}\left(\hat{\boldsymbol{\Sigma}}_m + \hat{\boldsymbol{\mu}}_m\hat{\boldsymbol{\mu}}_m'\right) - \left(\hat{\boldsymbol{\mu}}_m\mathbf{x}_m^{\mathtt{c}'} + \mathbf{x}_m^{\mathtt{c}}\hat{\boldsymbol{\mu}}_m'\right) \quad (25)$$

$$\boldsymbol{B}_0 = \beta_m^{\mathtt{c}}\mathbf{Y}_m^{\mathtt{c}} - \mathbf{x}_m^{\mathtt{c}}\mathbf{x}_m^{\mathtt{c}'} \quad (26)$$

The constant, D_m, is given by the largest positive eigenvalues of the Quadratic Eigenvalue Problem (QEP) of equation (23) [18]. In practice, a lower bound is applied to the smoothing constant value such that the actual smoothing constant value, $\hat{D}$, is given by

$$\hat{D} = \max\left(2D, E\beta_m^{\mathtt{d}}\right) \quad (27)$$

where the lower bound, $E\beta_m^{\mathtt{d}}$ is applied to ensure that the combined occupancy count, $\beta_m^{\mathtt{c}}$, is greater than zero. $E = 2$ is empirically found to lead to good test-set performance [15].

C. I-Smoothing

I-smoothing is an interpolation technique proposed by *Povey .et .al* [15] that incorporates prior information over each Gaussian parameters to control the convergence of the MPE training process. The prior is based on the ML statistics. Using I-smoothing requires the redefinition of the weak-sense auxiliary function (9) as

$$\mathcal{Q}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) = \mathcal{Q}^{\mathtt{n}}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) - \mathcal{Q}^{\mathtt{d}}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) + \mathcal{F}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) + p(\boldsymbol{\theta})$$

[1]Using this form of auxiliary function yields the same update formulae as using the extended Baum-Welch (EBW) algorithm [24], [25]

where

$$p(\boldsymbol{\theta}) = K + \frac{\tau^I}{2}\sum_{m=1}^{M}\left\{\log|\boldsymbol{P}_m| - \mathrm{Tr}\left(\boldsymbol{P}_m\mathbf{W}_m^{\mathtt{ml}}\right)\right\}$$

$$\mathbf{W}_m^{\mathtt{ml}} = \left(\frac{\mathbf{Y}_m^{\mathtt{ml}} - \mathbf{x}_m^{\mathtt{ml}}\mathbf{x}_m^{\mathtt{ml}\prime}/\beta_m^{\mathtt{ml}}}{\beta_m^{\mathtt{ml}}}\right)$$

τ^I is the I-smoothing constant. The prior can be regarded as the log likelihood of τ^I data points with the mean and variance of the ML estimate. Incorporating I-smoothing is easy by rewriting the combined statistics as

$$\mathbf{x}_m^{\mathtt{c}} = \mathbf{x}_m^{\mathtt{n}} - \mathbf{x}_m^{\mathtt{d}} + \frac{\tau^I}{\beta_m^{\mathtt{ml}}}\mathbf{x}_m^{\mathtt{ml}} \quad (28)$$

$$\mathbf{Y}_m^{\mathtt{c}} = \mathbf{Y}_m^{\mathtt{n}} - \mathbf{Y}_m^{\mathtt{d}} + \frac{\tau^I}{\beta_m^{\mathtt{ml}}}\mathbf{Y}_m^{\mathtt{ml}} \quad (29)$$

$$\beta_m^{\mathtt{c}} = \beta_m^{\mathtt{n}} - \beta_m^{\mathtt{d}} + \tau^I \quad (30)$$

It is simple to see that as the I-smoothing constant, τ^I, tends to infinity, the resulting estimation formulae tend to those of the ML training.

IV. MPE Training of Precision Matrix Models

Once the overall statistics in equations (20) and (21) are found, the auxiliary function in equation (9) can be maximised to discriminatively train the precision matrix model parameters. Since the MPE optimisation has been re-expressed in terms of a set of combined statistics and a function over those stastics that have exactly the same form as ML training all the standard ML optimisation formulae may be used. This is described in more detail in [20]. In this section, the basis coefficient updates for EMLLT and SPAM models are given as examples.

The basis coefficients of the EMLLT model, $\lambda_{ii}^{(m)}$, may be updated using the formula given in [10], modified to refelct the MPE statistics

$$\lambda_{ii}^{(m)} = \hat{\lambda}_{ii}^{(m)} + \left(\frac{1}{\hat{\boldsymbol{a}}_i\mathbf{W}_m^{\mathtt{mpe}}\hat{\boldsymbol{a}}_i'} - \frac{1}{\hat{\boldsymbol{a}}_i\hat{\boldsymbol{\Sigma}}_m\hat{\boldsymbol{a}}_i'}\right) \quad (31)$$

where $\hat{\lambda}_{ii}^{(m)}$ and $\hat{\boldsymbol{\Sigma}}_m$ are the current estimates of the basis coefficient and full covariance matrix respectively.

Similarly, the basis coefficients of SPAM models may be updated iteratively using the Polak-Ribeire conjugate-gradient method [27], as presented in [11]. The change in the auxiliary function, $\mathcal{Q}_\Delta^{(m)}$ for a corresponding change in the basis coefficient, Δ_m, is given by

$$\mathcal{Q}_\Delta^{(m)} = \frac{\beta_m^{\mathtt{mpe}}}{2}\left\{\sum_{j=1}^{d}\log\left(1+\Delta_m z_j^{(m)}\right) - \Delta_m\sum_{i=1}^{n}d_i^{(m)}\tilde{w}_i^{(m)}\right\}$$

where $z_j^{(m)}$ is the jth eigenvalue of $\hat{\boldsymbol{P}}_m^{-\frac{1}{2}}\left(\sum_i^n d_i^{(m)}\boldsymbol{S}_i\right)\hat{\boldsymbol{P}}_m^{-\frac{1}{2}}$, $d_i^{(m)}$ is the ith element of the direction vector calculated from the Polak-Ribeire conjugate-gradient method and $\tilde{w}_i^{(m)} = \mathrm{Tr}(\mathbf{W}_m^{\mathtt{mpe}}\boldsymbol{S}_i)$ is the *projected* statistics which will be discussed in Section V-A. As before, the standard ML full covariance statistics is replaced by equation (21) for MPE training. The formulation of determining $d_i^{(m)}$ is provided in [20].

V. Implementation Issues

This section addresses the implementation issues of various precision matrix models, paying particular attention to building LVCSR systems. Many of these models have been successfully applied to LVCSR systems [9], [12], [13]. This paper emphasises issues such as memory requirement, computational feasibility and training robustness in LVCSR systems. System efficiency may be adversely affected if these issues are not addressed properly. Here, various implementation issues for LVCSR systems will be considered.

A. Memory Issues

The major issue with implementing precision matrix models on LVCSR systems is the large amount of memory requirement for full covariance statistics accumulation. The update of the *tied* parameters is highly inefficient, especially for the SPAM models where the basis matrices are not rank-1. In general, it is more practical to get a good initial set of basis matrices and concentrate on updating the basis coefficients which is efficient. Moreover, updating basis coefficients does not require the full covariance statistics. For models with rank-1 basis (STC and EMLLT), the required statistics, $\mathbf{W}_m^{\mathtt{mpe}}$ is reduced to the so-called *projected* statistics, $\tilde{w}_i$

$$\tilde{w}_i = \hat{\boldsymbol{a}}\mathbf{W}_m^{\mathtt{mpe}}\hat{\boldsymbol{a}}' = \frac{\sum_{t=1}^{T}\gamma_m^{\mathtt{mpe}}(t)(\tilde{o}_{ti}-\tilde{\mu}_{mi})^2}{\beta_m^{\mathtt{mpe}}} \quad (32)$$

for $i = 1, 2, \ldots, n$. $\tilde{w}_i$ is a scalar term. $\tilde{o}_{ti} = \boldsymbol{a}_i\boldsymbol{o}_t$ and $\tilde{\mu}_{mi} = \boldsymbol{a}_i\boldsymbol{\mu}_m$ are the projected observation and mean vectors associated with the projection vector, $\boldsymbol{a}_i$. Hence, the total amount of memory required is proportional to n rather than $\frac{d}{2}(d+1)$ for the full covariance statistics, $\boldsymbol{W}_m$. This dramatically reduces the total memory requirement. The values of $\tilde{o}_{ti}$ and $\tilde{\mu}_{mi}$ will have been pre-computed and cached for efficient likelihood computation [20]. Thus, no extra cost is incurred in computing the projected statistics for STC and EMLLT models.

Likewise, the sufficient statistics required to update the basis coefficients for SPAM models can also be expressed in terms of the projected statistics, $\tilde{w}_i$, which is given by

$$\tilde{w}_i = \mathrm{Tr}\left(\boldsymbol{S}_i\boldsymbol{W}_m\right) = \frac{\sum_{t=1}^{T}\gamma_m(t)(\boldsymbol{o}_t'\boldsymbol{S}_i\boldsymbol{o}_t - 2\boldsymbol{\mu}_m'\boldsymbol{S}_i\boldsymbol{o}_t + \boldsymbol{\mu}_m'\boldsymbol{S}_i\boldsymbol{\mu}_m)}{\beta_m}$$

As before, the required memory is proportional to the basis order, n and the terms $\boldsymbol{o}_t'\boldsymbol{S}_i\boldsymbol{o}_t$ and $\boldsymbol{S}_i\boldsymbol{o}_t$ have already been computed and cached when calculating the likelihood.

B. Basis Initialisations

In the basis superposition framework, the basis vectors or matrices extract the common structure of the precision matrices of all Gaussian components. The update of the basis vectors for EMLLT models and basis matrices for SPAM models does not have a closed form solution and generic optimisation routines such as the conjugate gradient decent method [27] have to be used. Thus, it is important to obtain a

good initial set of basis to allow fast convergence and avoid hitting a poor local maximum during parameters estimation process. This is especially true for the EMLLT and SPAM models, where the update of basis vectors/matrices is slow. For STC, a trivial identity initialisation leads to a diagonal covariance matrix system. Several basis initialisation schemes are available for the EMLLT models [20]. The STC-HLDA initialisation scheme was found to be the best in terms of WER performance and is more flexible than simply stacking multiple STC transforms [10], which constrains n to be a multiple of d.

According to [11], it is useful to initialise the set of basis matrices $\{\boldsymbol{S}_i\}$ for SPAM models as the symmetric matrices associated to the top $n-1$ singular vectors of the matrix

$$\boldsymbol{V} = \frac{\sum_{m=1}^{M} c_m \boldsymbol{v}_m \boldsymbol{v}_m'}{\sum_{m=1}^{M} c_m} \tag{33}$$

where $\boldsymbol{v}_m = \mathrm{vec}(\boldsymbol{W}_m^{-1})$. On large systems, it was found that the basis matrix initialisation given by equation (33) is not robust due to the robustness of full covariance statistics for each component. Instead, the inverse of the state-level covariance statistics is used to produce a more reliable set of basis matrices [20].

C. Variance Flooring

In situations of data sparseness, which is common in LVCSR systems, a variance floor is required to prevent over-fitting. It imposes a lower bound to the variances (diagonal elements of the covariance matrix). The standard form, for example in HTK [28], of the variance floor, $\sigma_{ii}^{(vf)2}$, is

$$\sigma_{ii}^{(vf)2} = \frac{\alpha \sum_{s=1}^{S} \beta_s \sigma_{ii}^{(s)2}}{\sum_{s=1}^{S} \beta_s} \tag{34}$$

where $\mu_i^{(s)}$, $\sigma_{ii}^{(s)2}$ and β_s are the within state mean, variance and occupancy count respectively. α is a scaling factor which is typically set as 0.1 (10%). This method is readily applicable to the basis coefficients of the STC models due to the independent basis vectors [20].

However, the above method is not applicable to EMLLT models due to the existence of negative basis coefficients. Instead, in this work, variance floor is applied to the full covariance or projected statistics used to update the model parameters [19], [20]. Unfortunately, it is not possible to apply variance floor onto the projected statistics, $\mathrm{Tr}\,(\boldsymbol{W}_m \boldsymbol{S}_i)$, for SPAM models. However, if one of the basis matrices is initialised to be positive-definite ($\boldsymbol{S}_1$) [11], the coefficient corresponding to $\boldsymbol{S}_1$ can be gradually increased until the final precision matrices satisfy the variance floor condition. However, this approach is computationally inefficient.

D. Multiple Transformations Scheme

The basis superposition framework introduced earlier has an extreme basis tying scheme. A single set of basis matrices is shared by all the Gaussian components. This requires a large set of basis matrices to yield good representation. Alternatively, the components can be partitioned into clusters. Each cluster will then contain a smaller number of components. Extracting basis for each cluster of Gaussian components yields more accurate basis information. The basis matrices are now tied at the cluster level. This leads to a multiple projections scheme where each projection is associated with the set of basis matrices. A good summarisation of multiple projections schemes is given in [29]. Multiple HLDA projections models have been found to lead to good recognition performance [30]. For multiple projections basis superposition models, equation (1) can be rewritten as

$$\boldsymbol{P}_m = \sum_{i=1}^{n} \lambda_{ii}^{(m)} \boldsymbol{S}_i^{g(m)} \tag{35}$$

where $m \in g(m)$ and $g(m)$ denotes the cluster to which component m belongs to. There are many ways to perform Gaussian clustering. One way is to use a regression class tree [31] and the terminal nodes of the tree corresponds to the clusters of Gaussian components.

E. Approximating the Smoothing Constant, D_m

Determination of the smoothing constant value as described earlier is memory inefficient because solving the quadratic eigenvalue problem for equation (23) requires storing of the full covariance statistics. Storing the full covariance statistics for large systems results in intensive memory requirement.

For STC and EMLLT models, the smoothing constant can be determined more efficiently by considering the transformed mean and variance vectors. By applying the ith projection vector to equation (23), the transformed variance statistics (projected statistics [20]) are thus given by

$$\boldsymbol{a}_i \boldsymbol{W}_m \boldsymbol{a}_i' = \frac{b_2^{(i)} D_m^2 + b_1^{(i)} D_m + b_0^{(i)}}{\beta_m^{(c)} + D_m} \tag{36}$$

where $b_2^{(i)} = \boldsymbol{a}_i \boldsymbol{B}_2 \boldsymbol{a}_i'$, $b_1^{(i)} = \boldsymbol{a}_i \boldsymbol{B}_1 \boldsymbol{a}_i'$ and $b_0^{(i)} = \boldsymbol{a}_i \boldsymbol{B}_0 \boldsymbol{a}_i'$. More details on projected statistics will be given in Section V-A. Hence, the QEP is simplified to solving n independent quadratic equations given by equation (36) using only the transformed statistics, $\boldsymbol{a}_i \mathbf{x}_m^{(c)}$ and $\boldsymbol{a}_i \mathbf{Y}_m^{(c)} \boldsymbol{a}_i'$. Thus, the same set of statistics is used to determine the smoothing constant, D_m, and to estimate the model parameters.

Unlike STC and EMLLT models where the basis matrices are rank-1, the 'projected' statistics, $\mathrm{Tr}(\boldsymbol{W}_m \boldsymbol{S}_i)$, associated with the basis matrices, $\boldsymbol{S}_i$ of the SPAM model can not be used to determine the smoothing constant. There is no way to infer the positive-definiteness of the full covariance statistics, $\boldsymbol{W}_m$, from these 'projected' statistics. Instead of obtaining the exact smoothing constant value by solving the QEP for equation (23), this value can be approximated by using a *pseudo* transformation matrix, $\boldsymbol{A}^*$. The transformed space is assumed to have negligible correlation such that the QEP is once again broken down into n independent quadratic equations as for the STC and EMLLT models. Thus, two sets of statistics are required: one for determining the smoothing constant, D_m ($\boldsymbol{a}_i^* \boldsymbol{W}_m \boldsymbol{a}_i^{*'}$) and the other one for estimating the model parameters ($\mathrm{Tr}(\boldsymbol{W}_m \boldsymbol{S}_i)$). To yield a good approximation for the smoothing constant, $\boldsymbol{A}^*$ should

be chosen such that the transformed space is as uncorrelated as possible. So, it is intuitive to select the STC transform as the *pseudo* transformation matrix. In the case where STC transform is unavailable, an identity matrix may be used. This was found to be a good approximation [20].

VI. Experimental Results

Discriminative training of precision matrices was evaluated on an English conversational telephone speech (CTS) task, which consists of multi-speaker spontaneous telephone conversational speech, and an English broadcast news (BN) task, both provided by the Linguistic Data Consortium (LDC). Data was coded into 12 PLP coefficients at a frame rate of 10ms with a frame size of 25ms, together with the log energy term, first, second and third derivatives to form 52-dimensional feature vectors. Acoustic models are represented by decision tree state-clustered triphone models with 6189 distinct states. Side-based Ceptral Mean Normalisation (CMN), Cepstral Variance Normalisation (CVN) and Vocal Tract Length Normalisation (VTLN) are used in all the systems.

The models used in all the experiments were built using the HTK [28]. ML and MPE training were conducted with 4 and 8 iterations respectively. All HLDA systems used a 39×52 transformation matrix trained once at 16-component models and fixed for subsequent training. Basis matrices for EMLLT and SPAM models were initialised as described in Section V-B, where the STC-HLDA method was used for EMLLT models. For memory tractability, only basis coefficients were updated in MPE training. A multi-pass decoding strategy was employed where word lattices were first generated using a bigram language model and a dictionary comprising 58231 words with multiple pronunciation probabilities. These lattices were then rescored using a trigram language model to produce the final 1-best hypotheses.

Initial experiments were conducted based on the `h5etrain03` (296 hours) training set and the `dev01sub` (3 hours) test set of the CTS English task to evaluate the performance of various precision matrix models. The performance of multiple transforms systems was also compared using the same training and test sets. Finally, selected systems were tested on the full CTS (6 hours `eval03`) and BN (3 hours each for `dev03` and `eval03`) English tasks.

A. Development Results

This section evaluates the performance of various precision matrix models. 16-component models were trained because of rapid training to serve as an initial comparison. The word error rate (WER) numbers are summarised in Table I. The second and third columns show the dimensions for the mean vector and basis coefficients respectively. The HLDA ML model has a WER of 33.5% on `dev01sub`. If the nuisance dimensions are retained, the equivalent 52 dimensional STC model yields a further 0.2% absolute reduction in WER. By tying the 13 basis coefficients corresponding to the HLDA nuisance dimensions using a HLDA-PMM model, another 0.1% absolute improvement was obtained. With 78 basis coefficients, the EMLLT model is 0.9% absolute better than the HLDA model.The SPAM model, with half the number of basis coefficients, is only 0.2% absolute behind the EMLLT model. By applying SPAM within the HLDA subspace, the HLDA+SPAM model gave the best performance of 32.0% WER, which is 1.5% absolute better than the baseline HLDA. This illustrate the importance of compact model representation to yield robust and improved performance.

System	Dimensions		WER (%)	
	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$	ML	MPE
HLDA	39	39	33.5	29.8
HLDA+SPAM			32.0	28.5
HLDA-PMM	52	39	33.2	29.4
STC		52	33.3	29.7
EMLLT		78	32.6	29.2
SPAM		39	32.8	29.2

TABLE I
Comparison of WER (%) performance of ML and MPE trained 16-component precision matrix models on `dev01sub` CTS English task

The final column of Table I depicts the performance of the MPE models. The gain from MPE training is about 3.4–3.8% absolute. The gains form various precision matrix models were retained after MPE training. The WER of the HLDA and HLDA+SPAM MPE models were lowered to 29.8% and 28.5% respectively. This translates to an absolute improvement of 1.3% absolute.

As described in Section V-D, multiple transformations models provide a simple and powerful way of improving modelling accuracies without severely increasing the total number of model parameters. Gaussian clustering is performed in two different ways. For HLDA and STC models, a regression class tree is used to cluster the Gaussian components with an initial speech-silence split. Splitting criterion is based on the Euclidean distance between Gaussian components. This yields the 65-transform (64 speech, 1 silence) HLDA and STC models[2]. Gaussian components for the EMLLT models were clustered into 64 groups without an initial speech-silence split and the splitting is based on the Euclidean distance of the vectors of basis coefficients.

Table II summarises the WER results for multiple projections HLDA, STC and EMLLT models. These models are 0.8%, 1.0% and 0.6% absolute better than their corresponding single transform models. After 4 MPE iterations, the WER for the HLDA and EMLLT models were both reduced by 3.0% absolute while the STC model achieved a 2.6% absolute WER reduction. After 4 additional MPE iterations, the WER of the 64-transform EMLLT model was 28.3%, 0.9% absolute better than its single-transform model. The slow convergence of the basis update for SPAM models hinders the build of multiple transformation SPAM models. Although it is possible to initialise multiple sets of basis matrices for different cluster

[2]The multiple transforms HLDA and STC models were obtained from X. Liu. These models have been trained and decoded using the same setup as described earlier.

System	# of transforms	ML	MPE Iter.	
			4	8
HLDA	1	33.5	30.8	29.8
	65	32.7	29.7	–
STC	1	33.3	30.3	29.7
	65	32.3	29.7	–
EMLLT	1	32.6	29.8	29.2
	64	32.0	29.0	28.3

TABLE II
COMPARING WER (%) PERFORMANCE OF 16-COMPONENT PRECISION MATRIX MODELS WITH MULTIPLE TRANSFORMATIONS

of Gaussian components using the method described in Section V-B, the resulting basis matrices gave a poorer performance than the single transform SPAM models.

B. State-of-the-art Results

So far, the performance of various precision matrix models was presented based on the dev01sub test set for the CTS task. This section compares selected precision matrix models with the full CU-HTK LVCSR systems [16], [32] used in the 2003 Rich Transcription (RT03) evaluation[3]. The unadapted 28-component HLDA system was chosen as the baseline for comparison. The models were trained on h5etrain03 and evaluated on both dev01sub and eval03. Due to memory constraint, the basis matrices for EMLLT and SPAM models were initialised using the 16-component systems.

System	dev01sub		eval03	
	ML	MPE	ML	MPE
HLDA	32.3	29.1	31.7	28.4
HLDA+SPAM	31.1	27.9	30.4	27.3

TABLE III
WER PERFORMANCE OF 28-COMPONENT PRECISION MATRIX MODELS ON dev01sub AND eval03 FOR CTS TASK

The results are tabulated in Table III. The WERs of the ML HLDA model were 32.3% and 31.7% respectively. The gains from MPE are similar on both test sets, 3.2% and 3.3% respectively. The best single-transform system from before, HLDA+SPAM, was built with 28 Gaussian components per state. Both ML and MPE models consistently outperform the baseline by 1.2% absolute on dev01sub. On eval03, the gains after ML and MPE training were 1.3% and 1.1% absolute, giving the final WER of 27.3% for MPE HLDA+SPAM model. The gains from HLDA+SPAM in Table III were found to be statistically *significant* [4]. Although the 64-transform 16-component MPE EMLLT model is 0.8% absolute better than the 28-component HLDA model on dev01sub, this gain does not generalise to eval03. Only 0.3% improvement was obtained on this test set.

[3]See http://htk.eng.cam.ac.uk/docs/cuhtk.shtml

[4]Significance tests were carried out using the NIST Scoring Toolkit

A 16-component HLDA+SPAM model was also built to compare with the unadapted HLDA BN HLDA system trained on the bnac+TDT4 (375 hours) date set. These systems were evaluated on the dev03 and eval03 test sets, each consisting of 3 hours data. The results are tabulated in Table IV. The

System	dev03			eval03		
	ML	MPE	GD	ML	MPE	GD
HLDA	16.3	13.6	13.5	14.6	12.5	12.3
HLDA+SPAM	15.7	13.5	13.2	14.3	12.0	12.0

TABLE IV
WER PERFORMANCE OF 28-COMPONENT PRECISION MATRIX MODELS ON dev03 AND eval03 FOR BN TASKS

ML baseline WERs are 16.3%(dev03) and 14.6% (eval03). After MPE training, the WERs reduced to 13.6% and 12.5% respectively. An absolute gain of 0.6% was observed from HLDA+SPAM ML model on dev01sub. The corresponding gain on eval03 was only 0.3%. After MPE training, the gain from HLDA+SPAM was reduced to 0.1% on dev03 but was increased to 0.5% on eval03. Similar to the RT03 setup, gender dependent (GD) models were also built. Starting from the gender-independent (GI) MPE model, GD models were built with 3 MPE+MAP[26] iterations, using the corresponding GI MPE models as the prior. The baseline system gave a further 0.1% and 0.2% WER reductions on dev03 and eval03 respectively. Meanwhile, the HLDA+SPAM model yielded 0.3% improvement on dev03 but no further improvement was obtained on eval03. The final absolute gains of 0.3% on both test sets were found to be statistically significant.

VII. CONCLUSIONS

This paper has presented the large vocabulary discriminative training of various precision matrix models based on the minimum phone error criterion. The structured approximation of precision matrices was illustrated using a generic framework of basis superposition, which subsumes many existing models including the semi-tied covariance (STC), extended MLLT (EMLLT) and subspace for precision and mean (SPAM) models. These models have efficient likelihood calculation which leads to efficient decoding.

Various issues concerning large system implementation were addressed. In particular, computational tractability and memory requirement are two important factors that determine the efficiency of the systems. Issue with high computational cost and slow convergence of the basis matrix update was overcome with good initialisation schemes. This also allows the models to be trained by updating only the basis coefficients, which is more efficient and requires significantly less memory. The inefficiency in solving the QEP to find the smoothing constant for the SPAM models was alleviated by using a *pseudo* transformation matrix to mimic the smoothing constant determination process for STC or EMLLT models.

Experimental results reveal that various precision matrix models outperform the standard HLDA diagonal covariance matrix system on the CTS English Task. Without dramatical

increase in the total number of model parameters, multiple transformations models were found to yield between 2% to 5% relative reduction in word error rate compared to single-transform models. The best performance was achieved by modelling the precision matrices using the SPAM model within a HLDA subspace (HLDA+SPAM). 1.1% and 0.2% absolute WER reductions were obtained on conversational telephone speech (CTS) and broadcast news (BN) tasks respectively over the unadapted HLDA model used in the 2003 Rich Transcription (RT03) evaluation.

In a nutshell, various precision matrix models have been successfully implemented in LVCSR discriminatively trained systems and several implementation issues were addressed to yield robust training and efficient decoding.

ACKNOWLEDGMENT

The authors would like to thank X. Liu, Cambridge University, for his multiple transforms HLDA and STC models. This work was supported by DARPA grant MDA972-02-1-0013. The paper does not necessarily reflect the position or the policy of the US Government and no official endorsement should be inferred.

REFERENCES

[1] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," *Bull. Amer. Math. Soc.*, vol. 73, pp. 360–363, 1967.

[2] S. J. Young, "Large vocabulary continuous speech recognition: A review," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Snowbird, Utah, December 1995, pp. 3–28.

[3] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Speech and Audio Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[4] H. Hermansky, "Perceptual Linear Predictive (PLP) analysis of speech," *Journal of the Acoustic Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[5] G. Saon, M. padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," in *Proc. ICASSP*, 2000.

[6] N. Kumar, "Investigation of silicon-auditory models and generalization of linear discriminant analysis for improved speech recognition," Ph.D. dissertation, Johns Hopkins University, 1997.

[7] N. K. Goel and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced-rank HMMs for improved speech recognition," *Speech Communications*, vol. 26, pp. 283–297, 1998.

[8] A.-V. I. Rosti and M. J. F. Gales, "Factor analysed hidden Markov models for speech recognition," Cambridge University, Tech. Rep. CUED/F-INFENG/TR453, 2003. [Online]. Available: (via anonymous) ftp://svr-www.eng.cam.ac.uk

[9] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[10] P. Olsen and R. A. Gopinath, "Modelling inverse covariance matrices by basis expansion," in *Proc. ICASSP*, 2002.

[11] S. Axelrod, R. Gopinath, and P. Olsen, "Modeling with a subspace constraint on inverse covariance matrices," in *Proc. ICSLP*, 2002.

[12] J. Huang, V. Goel, R. A. Gopinath, B. Kingsbury, P. Olsen, and K. Visweswariah, "Large vocabulary conversational speech recognition with the extended maximum likelihood linear transformation (EMLLT) model," in *Proc. ICSLP*, 2002.

[13] S. Axelrod, V. Goel, B. Kingsbury, K. Visweswariah, and R. A. Gopinath, "Large vocabulary conversational speech recognition with a subspace constraint on inverse covariance matrices," in *Proc. Eurospeech*, 2003.

[14] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 25–48, Jan 2002.

[15] D. Povey and P. C. Woodland, "Minimum Phone Error and I-smoothing for improved discriminative training," in *Proc.ICASSP*, 2002.

[16] G. Evermann, H. Y. Chan, M. J. F. Gales, T. Hain, X. Liu, D. Mrva, L. Wang, and P. C. Woodland, "Development of the 2003 CU-HTK conversational telephone speech transcription system," in *Proc. ICASSP*, 2004.

[17] J. McDonough and A. Waibel, "Maximum mutual information speaker adapted training with semi-tied covariance matrices," in *Proc. ICASSP*, 2003.

[18] V. Goel, S. Axelrod, R. Gopinath, P. Olsen, and K. Visweswariah, "Discriminative estimation of Subspace Precision and Mean (SPAM) models," in *EUROSPEECH*, 2003.

[19] K. C. Sim and M. J. F. Gales, "Basis superposition precision matrix modelling for large vocabulary continuous speech recognition," in *Proc. ICASSP*, 2004.

[20] K. C. Sim and M. J. F. Gales, "Precision matrix modelling for large vocabulary continuous speech recognition," Cambridge University, Tech. Rep. CUED/F-INFENG/TR485, 2004. [Online]. Available: (via anonymous) ftp://svr-www.eng.cam.ac.uk

[21] K. Visweswariah, P. Olsen, R. Gopinath, and S. Axelrod, "Maximum likelihood training of subspaces for inverse covariance modeling," in *Proc. ICASSP*, 2003.

[22] 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*, vol. 39, pp. 1–39, 1977.

[23] S. Tsakalidis, V. Doumpiotis, and W. Byrne, "Discriminative linear transforms for feature normalisation and speaker adaptation in hmm estimation," in *Proc. ICSLP*, 2002.

[24] P. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Trans. Info. Theory*, pp. 107–113, 1991.

[25] Y. Normandin, "Hidden Markov models, maximum mutual information estimation and the speech recognition problem," Ph.D. dissertation, McGill University, Montreal, 1991.

[26] D. Povey, P. C. Woodland, and M. J. F. Gales, "Discriminative MAP for acoustic model adaptation," in *ICASSP*, 2003.

[27] E. Polak, *Computational Methods in Optimization: A Unified Approach*. Academic Press, 1971.

[28] S. J. Young, D. Kershaw, J. J. Odell, D. Ollason, V. Valtchev, and P. C. Woodland, *The HTK Book (for HTK version 3.0)*. Cambridge University, 1997.

[29] M. J. F. Gales, "Maximum likelihood multiple projection schemes for hidden Markov models," Cambridge University, Tech. Rep. CUED/F-INFENG/TR365, 1999. [Online]. Available: (via anonymous) ftp://svr-www.eng.cam.ac.uk

[30] X. Liu and M. J. F. Gales, "Automatic model complexity control using marginalized discriminative growth functions," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2003.

[31] M. J. F. Gales, "The generation and the use of regression class trees for MLLR adaptation," Cambridge University, Tech. Rep. CUED/F-INFENG/TR263, 1996. [Online]. Available: (via anonymous) ftp://svr-www.eng.cam.ac.uk

[32] D. Y. Kim, G. Evermann, T. Hain, D. Mrva, S. E. Tranter, L. Wang, and P. C. Woodland, "Recent advances in broadcast news transcription," in *Proc. ASRU*, 2003, pp. 105–110.

Khe Chai Sim Biography text here.

Mark Gales Biography text here.