Subspace Gaussian Mixture Models for Large Vocabulary Speech Recognition

Liang Lu

May 14, 2010
Abstract

Subspace Gaussian mixture model (GMM) is an alternative approach to approximate the probabilistic density function (p.d.f) of a set of independent identical distributed (i.i.d) data with prior density estimates. In this approach, the prior density of GMM parameters is estimated from a development dataset, and when predict the new enrolled data, the prior knowledge can be utilised by criteria like Maximum a Posterior. Unlike the conventional prior estimate method for GMM, the correlations between parameters of different Gaussian components are considered in this approach. In order to handle the large size of parameter set and meanwhile to ensure the priors be informative, the prior density estimation is constraint to a low dimensional subspace of the whole model space which can capture the main model variations.

The subspace GMM has already been successfully applied in the task of speaker recognition, and achieved promising performance, but there is no much work of applying this approach to speech recognition. In this paper, we will present a new framework of HMM based speech recognition system based subspace GMM, in which, the parameters of state-dependent GMM are not estimated separately but been generated from the globally shared low dimensional model subspace. The approach can considerably reduce the model size and in addition, make the speech recognition system more scalable and adaptable. In this paper, we will first review the principles of subspace GMM approach based on its applications in speaker recognition and then discuss how to extend it to the task of speech recognition.
Contents

1 Introduction 2

2 Subspace GMM, An Overview 4
  2.1 Latent variable models 4
  2.2 Classic MAP 5
  2.3 Modelling priors in subspace 6

3 Subspace GMM for Speaker Recognition 7
  3.1 The likelihoods and posteriors 8
  3.2 Model estimation 10
  3.3 Model adaptation 11
  3.4 Mixtures of subspaces 11
  3.5 Model extension with multiple factors 13
  3.6 Point estimate vs. Uncertainty estimate 15

4 Subspace GMM for Speech Recognition 16
  4.1 Basic model description 16
  4.2 Modelling the speaker subspace 18
  4.3 Model extension with sub-states and sub-models 19
  4.4 Discriminative training 21

5 Experimental results 23
  5.1 System setup 23
  5.2 Results and discussion 24

6 Future works 27
  6.1 Short-term plan 28
  6.2 Long-term plan 29

7 Conclusion 29

A Proofs of propositions in section 3 33
  A.1 Proof of proposition 1 33
  A.2 Proof of proposition 2 34
  A.3 Proof of proposition 3 34
  A.4 Proof of proposition 4 35

B Update of the weight projection 37
1 Introduction

For the task of large vocabulary continuous speech recognition (LVCSR), the continuous density Hidden Markov Models (CDHMM) with Gaussian mixture models (GMM) for emission probability density is now the prevalent approach [1]. Although the success of such CDHMM approach for speech recognition has been well recognised by the community, it still suffers a major problem, namely, the large number of model parameters. In order to model coarticulation, a context dependent phone system normally has thousands of emitting states. Even though some parameter sharing approaches have been developed to reduce the model size, such as state tying [2] to reduce the total number of Gaussian mixtures and semi-tied covariance matrix (STC) [3] to seek a compromise between diagonal and full covariance system, the number of parameters for a typical settings of speech recognition system can still easily be several million.

The large number of parameters in the conventional HMM based speech recognition system can lead to two problems. Firstly, such systems normally require large amount of training data to fit the model. The second, which is also a very important problem, is the difficulty to adapt the model from one domain to another or from a speaker-independent to a speaker-dependent model because of the large model size. Since the data and system mismatch is one of the major causes for performance degradation, numerous approaches have been proposed to address the model adaptation difficulty, such as Maximum a Posterior (MAP) adaptation [4], maximum likelihood linear regression (MLLR) [5] and eigenvoices [6]. As the MAP approach adapts each model parameter independently, it normally requires a large amount of adaptation data to achieve good results; however, it’s advantage lies in that it can be asymptotic to the maximum likelihood (ML) estimate given sufficient adaptation data. MLLR based speaker adaptation, however, requires relatively little adaptation data to achieve good performance as the number of parameters in the transformation matrix is very small, but it does not converge to the ML estimate given large amount of adaptation data. Eigenvoice adaptation is also effective for relatively little adaptation data, but it needs a large amount of development data (normally from hundreds of speakers) to estimate the principle components of speaker variations in the whole model space, and, again, it can not guarantee to be equivalent to the ML estimate given sufficient adaptation data.

In this paper, we will address the large model size problem of conventional speech recognition system from another perspective. Unlike the conventional approach in which the Gaussian mixture models are estimated separately for each emitting state, we will consider all these Gaussian parameters for each state are generated from a global shared subspace. In this approach, which termed the subspace GMM [7], the subspace will model the main correlations of Gaussian mixture models from all the context dependent phone states, hence, only a small number of parameters are needed to well characterise each state specific GMM. Although the number of parameters in the subspace can be very large, we can reduce the number of state specific parameters leading to an overall model that is more compact compared to the conventional one. In addition, as all the Gaussian
mixture models vary in the same model subspace, model adaptation can be fairly simple to perform. Note however, as we have to put the GMMs from all the states into the same model space, it will result in the same number of Gaussian mixture components for all the states. Nevertheless, we can still scale the size of different states according to the state specific parameters. We will leave this question to later sections.

To explain this approach a little bit further, we can view the subspace GMM as a kind of extended MAP for GMMs as the subspace actually plays the role of priors to the GMM model parameters. However, unlike conventional MAP for GMMs [4], in which, the parameters of the prior density function are assumed independent between different Gaussian components for simplicity, the subspace GMM relaxes the constraint and model the correlations between parameters. Since the Gaussian mixture weights, means and covariances belong to different distribution families, it is difficult to put all the parameters into one model space in order to explore such correlations. To work around this problem, we only model the priors for a subset of parameters (specifically Gaussian mixture means and weights as required) and fix the others (mainly the covariance matrices which are difficult to handle in one model space) the same for all the target GMMs. Even though, we still face another problem, namely, the large size of the model space makes it impossible to capture all the correlations of all the parameters based on limited development data. To handle this problem, we just model correlations in a subspace which can capture the main model variation, hence the name \textit{subspace Gaussian mixture model}. Such an idea shares much similarity with the probabilistic version of principal component analysis (PCA) for dimension reduction [8], together with some extensions.

In the field of speech recognition, the subspace GMM is closely related to eigenvoice adaptation [6] and cluster adaptive training (CAT) [9]. In both approaches, a principal model subspace is also introduced by PCA estimate or maximum likelihood estimate for speaker adaptation. The subspace GMM differs from them mainly in that the subspace does not model the correlations of speaker models but the phone state models. Moreover, the idea of the subspace GMM has also been successfully applied in the task of speaker recognition, which is normally referred as joint factor analysis (JFA) [10]. In JFA, a speaker’s GMM mean parameters are decomposed into speaker-dependent and channel-dependent factors which lies in speaker subspace and channel subspace respectively. The speaker subspace is introduced for robust estimate of speaker models with spare training data [11] while the channel subspace, which can capture the major model variabilities caused by channel effect, environment noise, etc [12], is included for channel compensation which has proven effective for speaker recognition [13].

For speech recognition, some pilot work of applying subspace GMM to build speech recognition systems has already been carried out by Povey, Burget et.al., in the 2009 JHU summer workshop [14, 7]. They have shown that the subspace GMM based speech recognition system can achieve better performance with a smaller model size compared with conventional modelling approach. However, as the research on this direction is just beginning, much work is still required to shape this new speech recognition framework.
The purpose of this paper is to present a systematic review of the subspace GMM approach including its principles and applications in speaker recognition as well as the current work in speech recognition. Then we will focus on the discussion of possible extensions of the subspace GMM for speech recognition which will lay a foundation for our future work.

The rest of the paper is organised as follows, we will first discuss the related modelling approach and motivations for the subspace GMM approach in section 2. And then in section 3, we will present a detailed overview of the application of subspace GMM in speaker recognition, which include likelihood calculations, model training and adaptation, possible extensions etc. In section 4, we will discuss how to apply the idea of subspace GMMs into speech recognition together with some possible topics for further investigation. Some preliminary results as well as some discussions are given in section 5 and in section 6, some future works are given. Finally, conclusions are drawn in section 7.

2 Subspace GMM, An Overview

From the perspective of statistical modelling, the subspace GMM belongs to the family of latent variable models. In section 2.1, we will first discuss two examples of latent variable models, namely, factor analysis [15] and probabilistic PCA [8] which are closely related to the subspace GMM. We will also briefly review the classic MAP for GMM modelling [4] in section 2.2 as the subspace GMM can also be viewed as a kind of extended MAP approach, Finally, the motivations of subspace GMM as well as the basic formula are presented in section 2.3.

2.1 Latent variable models

Factor analysis [15, 16] is a common example of latent variable modelling which links a d-dimensional observable vector \( t \) with a q-dimensional latent variables \( x \) in the form of:

\[
t = Wx + \mu + \epsilon
\]

where \( \epsilon \) is the noise model which is assume to be Gaussian distributed with zero mean and diagonal covariance matrix as \( \mathcal{N}(0, \Psi) \). The \( d \times q \) rectangular matrix \( W \) is normally referred as factor loading matrix and the latent variables are standard Gaussian distributed, i.e., \( x \sim \mathcal{N}(0, \Psi) \). The parameter \( \mu \) permits the model to have a non-zero mean. Hence, from this formulation, the observation vectors are also Gaussian distributed as \( t \sim \mathcal{N}(\mu, C) \) where \( C = WW^* + \Psi \). Since \( \Psi \) is diagonal matrix, one important property of factor analysis is that the observable vector \( t \) is conditionally

\[1\]In this paper, we denote \((\cdot)^*\) as matrix or vector transpose. Later, we will denote \(tr(\cdot)\) as the trace of matrices
independent given the latent variables \( x \), and in addition, we commonly choose \( d < q \) for matrix \( W \) for factor analysis model which means we can represent high dimensional observations by low dimensional variables.

The probabilistic principle component analysis (PPCA) \([8]\) shares much similarity with factor analysis model except that, the noise term in expression (1) is not assumed diagonal but isotropic distributed, namely, \( \epsilon \sim \mathcal{N}(0, \sigma I) \). The minor change brings several differences in the behaviour of two methods. Firstly, the subspace defined by \( W \) does not generally correspond to the principal subspace of the data for factor analysis model while it does for PPCA. In addition, if we consider a non-singular transformation of the observable data \( t \to At \), then PPCA is covariant for orthogonal transformation matrix, namely, \( AA^* = I \) while for factor analysis model, the property is held for diagonal transformation matrix \( A \) \([16]\). Finally, we note that for both factory analysis and PPCA models, there is no closed-form analytic solutions of model parameters and we normally resort to EM algorithm to estimate the model.

2.2 Classic MAP

Compared with maximum likelihood estimation (MLE), MAP involves a prior distribution of the parameter set. The authors in \([4]\) formulated the MAP estimate procedure for GMMs and HMMs which we refer as the classic MAP estimator. If we denote the observations as \( X \) with each frame of dimensionality of \( F \) and the Gaussian mixture model is in the form of

\[
P(X|\theta) = \sum_{c} w_c \mathcal{N}(X; \mu_c, \Sigma_c)
\]

(2)

where \( w_c, \mu_c \) and \( \Sigma_c \) are the mixture weight, mean and covariance matrix for \( c_{th} \) Gaussian component, respectively, and \( \theta \) denotes the parameter set \( \theta = \{(w_c, \mu_c, \Sigma_c) | c = 1, \ldots, C\} \), then the MAP estimation for GMMs can be simply expressed as:

\[
\theta_{MAP} = \arg \max_{\theta} g(\theta|X)
= \arg \max_{\theta} P(X|\theta)g(\theta).
\]

(3)

where \( g(\theta) \) is prior distribution of the GMM parameter set. The important and some times difficult problem for MAP estimation is to determine the prior distribution \( g(\theta) \). In \([4]\), the prior distribution of Gaussian mixture weights and different component means and covariances are assume to be independent in order to apply the conjugate prior density to simplify the estimation process. In particular, for the Gaussian mixture weight, the prior density the author used is the conjugate density such as a Dirichlet density

\[
g(w_c, \ldots, w_C|\nu_1, \ldots, \nu_C) \propto \prod_{c=1}^{C} w_c^{\nu_k-1}
\]

(4)
where $\nu_c > 0$ are the parameters for the Dirichlet density. As for the mean and variance $(\mu_c, \Sigma_c)$ for $c$th Gaussian component, a normal-Wishart density is used as the joint conjugate prior density which is of the form

$$g(\mu_c, \Sigma_c | \tau_c, \eta_c, \alpha_c, v_c) \propto \left| \Sigma_c \right|^{(\alpha_c-p)/2} \exp \left[ -\frac{\tau_c}{2} (\mu_c - \eta_c)^* \Sigma_c (\mu_c - \eta_c) \right] \exp \left[ -\frac{1}{2} tr(v_c \Sigma_c) \right]$$

(5)

where $(\tau_c, \eta_c, \alpha_c, v_c)$ are the prior density parameters such that $\alpha_c > F - 1$, $\tau_c > 0$, $\eta$ is a vector of dimension $F$ and $v$ is a $F \times F$ positive definite matrix. To apply the above prior densities into the MAP estimation of GMMs, we need further assume the independence between the parameters of different mixture components and the set of mixture weights. Hence, in this case, the joint prior density $g(\theta)$ is the product of the p.d.f defined in equation (4) and (5), i.e.

$$g(\theta | \phi) = g(w_1, \ldots, w_C) \prod_{c=1}^{C} g(\mu_c, \Sigma_c)$$

(6)

where we denote $\phi = \{\nu, \tau, \eta, \alpha, v\}$ as the hyperparameter set governing the distribution of $g(\theta)$. To learn the prior distribution, namely, the hyperparameter set $\phi$, we can adopt the maximum likelihood of marginal density $P(X | \phi)$ as:

$$\phi = \arg \max \phi P(X | \phi)$$

$$= \arg \max \phi \int f(X | \theta) g(\theta | \phi) d\theta$$

(7)

However, learning the prior density of $g(\theta)$ in the classic MAP is not easy as the number of hyperparameters $\phi$ is not less than that of GMMs. In real applications, some constraints are applied to the prior density to reduce the number of parameters, e.g., the priors are set to be proportional to ML estimate of the GMMs parameters as in relevance MAP for speaker recognition [17]. In fact, such a MAP estimation approach seems very ineffective to explore the prior knowledge from the development dataset, mainly due to the large model size itself. In the next section, we will consider another prior modelling approach which reduces the model size by modelling the correlations of the parameters.

### 2.3 Modelling priors in subspace

Model the correlations of different parameters for GMM in the prior density function is not a simple task, and there is no an explicit solution in general. However, for the subspace GMM, we simplify the problem by fixing the mixture weights and covariance matrices to be the same for all the target models, and only explore the priors for a subset of the model parameters (specifically Gaussian component means). To make the expression clear, denote $m$ as a $CF \times 1$ supervector obtained by concatenating all the
means of the GMM \((\mu_1, \mu_2, \ldots, \mu_C)\). The basic assumption for the subspace GMM is that all the supervectors are Gaussian distributed with mean \(\bar{m}\) and covariance matrix \(T\). However, as the dimensionality of supervectors can be very large (they could be thousands or tens of thousands dimensional), it is impossible to learn the full rank covariance matrix \(T\) from limited development data. The subspace GMM handles this problem by constraining the variations of all the supervectors into a low dimensional but most informative subspace [11]. If we define \(U\) as a \(CF \times R\) matrix where \(R \ll CF\), then for the subspace GMM, we approximate the covariance matrix \(T\) by \(UU^*\). As \(U\) is a low dimensional matrix, it has many fewer parameters which makes it possible learn from limited data with tractable computation. Meanwhile, it only requires a little amount of data to adapt to the target GMMs. Let \(m(s)\) be the Gaussian mean supervector for our target model, then in the subspace GMM framework, it can be expressed as

\[
m(s) = \bar{m} + Uy(s)
\]

where \(y(s)\) is an \(R\) dimensional latent factor \(^2\) which is to be learned from the adaptation data. From the above discussion, we can see that \(y(s)\) is distributed according to standard normal distribution, i.e., \(y(s) \sim N(0, I)\). For the subspace GMM, we have to work out a probabilistic approach to estimate the subspace matrix \(U\) and the latent variables \(y(s)\) for model adaptation. In fact, it is apparent from expression (8) that the subspace GMM is quite similar to factor analysis and probabilistic PCA, discussed in section 2.1, hence, the model estimation approach and some properties of these two models are applicable to subspace GMM. However, expression (8) is just the basic representation for subspace GMM modelling method; later in this paper, we will also discuss various extensions include using multiple subspace matrix to approximate \(T\) and relaxing our unimodal assumption of supervectors to multimodal in the next section.

3 Subspace GMM for Speaker Recognition

In this section, we will review some principles of the subspace GMM based on its application in speaker recognition [11, 10, 18, 19, 20]. We will begin with some essential calculations such as likelihoods and posteriors and then go through model estimation, adaptation and some other extensions. Note however, although the discussions are based on speaker recognition scenario, general conclusions are applicable when they are applied in speech recognition.

Before getting into details, it will be helpful to define the Baum-Welch statistics in the usual way as in the field of speech recognition. Given the acoustic data \(\mathcal{X}(s) = \)

\(^2\)In this paper, we will use the term latent factor, latent variables and state-specific vector (when discussing speech recognition) interchangeably to denote \(y\)
for speaker $s$, its Baum-Welch statistics are defined as

\[
\gamma_{sc} = \sum_{t=1}^{T_s} \gamma_{sc}(t)
\]

\[
F_{sc} = \sum_{t=1}^{T_s} \gamma_{sc}(t)x(t)
\]

\[
S_{sc} = \sum_{t=1}^{T_s} \gamma_{sc}(t)x(t)x(t)^*\]

where $\gamma_{sc}(t)$ is the posterior probability of the event that the feature vector $x(t)$ is accounted for by the mixture component $c$, $\gamma_{sc}(t) = \frac{w_c \mathcal{N}(x(t)|\mu_1, \Sigma_1)}{\sum_{i=1}^{C} w_i \mathcal{N}(x(t)|\mu_i, \Sigma_i)}$, for speaker $s$. $F_{sc}, S_{sc}$ are the first and second order statistics collected from the acoustic data. As in equation (8), our model includes a centre point in the model space $\bar{m}$. It will be helpful to simplify expressions by further defining the centralised first and second order Baum-Welch statistics as follows:

\[
\bar{F}_{sc} = \sum_{t=1}^{T_s} \gamma_{sc}(t) (x(t) - \bar{m}_c)
\]

\[
\bar{S}_{sc} = \sum_{t=1}^{T_s} \gamma_{sc}(t) (x(t) - \bar{m}_c)(x(t) - \bar{m}_c)^*
\]

where $\bar{m}_c$ denotes the mean vector corresponding to $c_{th}$ Gaussian component.

### 3.1 The likelihoods and posteriors

Suppose we are given the acoustic data $X(s)$ for speaker $s$, and the latent variables $y(s)$ in the subspace, we will show in this section how to derive the conditional and marginal likelihood of speech data given by the latent factors, namely, $P(X(s)|y(s))$ and $P(X(s))$ as well as the posterior distribution of the latent variables $P(y(s)|X(s))$. Those calculations will play an important role in our model estimation, adaptation as well as decoding the subspace GMMs based speech recognition system. Hence, we will present a comparable detailed review here, mainly based on [11].

#### A. The conditional likelihood of the data

We give the conditional likelihood of $P_U(\Sigma(X(s)|y(s)))$ by the following proposition.

**Proposition 1**: Given the parameter set $(U, \Sigma)$ and current estimate of latent factor $y(s)$, the conditional log-likelihood of $X(s)$ can be expressed as follows:

\[
\log P_U(\Sigma(X(s)|y(s))) = \sum_{c=1}^{C} (G_{\Sigma}(s, c) + H_U(\Sigma(s, c))
\]
Where we denote $G(s, c)$ and $H_{U,s}(s, c)$ as follows:

$$G(s, c) = \gamma_{sc} \log w_c + \frac{1}{2} \left[ \gamma_{sc} \log (2\pi)^{-F} |\Sigma_c^{-1}| - \text{tr} \left( \Sigma_c^{-1} \bar{S}_{sc} \right) \right]$$  \hfill (15)

$$H_{U,s}(s, c) = -\frac{1}{2} y^*(s)U_c*\gamma_{sc}\Sigma_c^{-1}U_c y(s) + y^*(s)U_c*\Sigma_c^{-1}F_{sc}$$ \hfill (16)

And $U_c, \Sigma_c$ are the subspace matrix and covariance matrix corresponding to the $c_{th}$ Gaussian component. $\bar{F}_{sc}$ and $\bar{S}_{sc}$ are the centralised first and second order statistics defined by equation (12) and (13). It is necessary to mention that the terms $U_c*\Sigma_c^{-1}U_c$ and $U_c*\Sigma_c^{-1}$ are normally just computed once for efficiency. As we can see from (16) that they are speaker independent but expensive to compute. The proof of proposition 1 is given in appendix A.1.

**B. The posterior distribution of the hidden variables**

The posterior distribution of the latent variables $y(s)$ are useful for model estimation and model adaption. As discussed in section 1, the prior distribution of hidden variables $y(s)$ for every speaker is standard Gaussian. If we have some collection of speech data from speaker $s$, then we can calculate the posterior of $y(s)$ as proposition 2 given below.

**Proposition 2:** If we define $\alpha(s)$ and $\beta(s)$ as follows:

$$\alpha(s) = I + \sum_{c=1}^{C} U_c^*\gamma_{sc}\Sigma_c^{-1}U_c$$ \hfill (17)

$$\beta(s) = \sum_{c=1}^{C} U_c^*\Sigma_c^{-1}\bar{F}_{sc}$$ \hfill (18)

Then given the training data for speaker $s$, $X(s)$ and current estimation of the parameter set $(U, \Sigma)$, the posterior distribution of latent variables $P(y(s)|X(s))$ is Gaussian with mean $\alpha^{-1}(s)\beta(s)$ and covariance matrix $\alpha^{-1}(s)$. The proof is given in appendix A.2.

**C. The marginal likelihood of the training data**

In this paragraph, we will show how to evaluate the marginal likelihood given a set of training data $\{X(1), \ldots, X(S)\}$ from $S$ speakers which is defined as:

$$\sum_{s=1}^{S} \log P_{U,s}(X(s)) = \sum_{s=1}^{S} \int P_{U,s}(X(s)|y(s))N(y(s); 0, I)dy$$ \hfill (19)

This likelihood function will serve as a diagnostic for verifying the implementation of the EM algorithm for hyperparameter estimation that we will present later. And it also serves as the basis for uncertainty estimate of the model which we will discuss in detail in section 3.6. We present the likelihood function as another proposition.
Proposition 3: Given the parameter set of the model \((U, \Sigma)\), the complete likelihood of training data \(X(s)\) can be expressed as

\[
\log P_{U, \Sigma}(X(s)) = \sum_{c=1}^{C} G_{\Sigma}(s, c) - \frac{1}{2} \log |\alpha(s)| + \frac{1}{2} \|\alpha^{-1/2}(s) \beta(s)\|^2
\]

where \(\alpha(s)\) and \(\beta(s)\) is defined as equation (17) and (18), and \(G_{\Sigma}(s, c)\) as equation (15). See proof in appendix A.3.

3.2 Model estimation

In this section, we will discuss the estimation of the parameters of Gaussian mixture model with a latent subspace. The parameter set for update we consider here is \((m, U, \Sigma)\). We do not consider the update for mixture weights as on the one hand it’s very straightforward when necessary and on the other hand, we will relax the weight constraint to be state dependent in the case of speech recognition in section 4. The model estimation is composed of three parts, namely, the estimation of \(\Sigma, m\) and the hyperparameter \(U\). As the latent factor \(y(s)\) for each speaker can not be observed directly, a close form solution for the model estimation does not exist. However, we can adopt the EM algorithm [21] to update the model parameters iteratively, which mainly involves two steps as follows:

step 1: For each speaker \(s\), using the current estimate the parameter set \(\hat{\theta} = (U_0, \Sigma_0, \tilde{m}_0)\), evaluate the posterior probability of the latent factor \(y(s)\), \(P(y(s)|X(s), \hat{\theta})\) by proposition 2.

step 2: Update the parameters by maximising the follow auxiliary function

\[
\theta = \underset{\theta}{\text{argmax}} Q(\theta, \hat{\theta})
\]

Where

\[
Q(\theta, \hat{\theta}) = \sum_{s=1}^{S} \int P(y(s)|X(s), \hat{\theta}) \log P(X(s), y(s)|\theta) dy
\]

From the EM algorithm, we can get the parameter updating formula as follows:

Proposition 4:

\[
\tilde{U}_c = \left( \sum_s F_{sc} E[y^*(s)] \right) \left( \sum_s \gamma_{sc} E[y(s)y^*(s)] \right)^{-1}
\]

\[
\Sigma_c = \sum_{s} \left[ S_{sc} - \frac{1}{2} \left( F_{sc} E[y^*(s)] \tilde{U}_c^* + \tilde{U}_c E[y(s)] F_{sc}^* \right) \right]
\]

\[
\tilde{m}_c = \frac{1}{\sum_s \gamma_{sc}} \sum_s \left( F_{sc} - \gamma_{sc} \tilde{U}_c y(s) \right)
\]
where \( E[y(s)] \) and \( E[y(s)y^*(s)] \) is the first and second order expectation of the latent variable, and from proposition 2, we can obtain

\[
E[y(s)] = \alpha^{-1}(s)\beta(s) \tag{26}
\]

\[
E[y(s)y^*(s)] = E[y(s)]E[y(s)]^* + \alpha^{-1}(s) \tag{27}
\]

By \( \hat{\hat{\alpha}} \), we mean the updated or new parameters. Proof of proposition 4 is given in the appendix A.4.

### 3.3 Model adaptation

Given the parameter set of subspace Gaussian mixture models \( \theta = (\bar{m}, \Sigma, U) \), it is very straightforward to adapt the GMMs for a new enrolled speaker \( s \) given the adaptation data \( \mathcal{X}(s) \) by MAP criteria as

\[
y(s) = \arg \max_y P(\mathcal{X}(s)|\bar{m} + U_y, \Sigma)N(y|0, I) \tag{28}
\]

From proposition 2, we can obtain the point estimate of the model as

\[
m_c(s) = \bar{m}_c + U_c\alpha^{-1}(s)\beta(s) \tag{29}
\]

\[
B_c(s) = U_c\alpha^{-1}(s)U_c^* \tag{30}
\]

where \( \bar{m}_c(s) \) denotes the mean for \( c \)th Gaussian component and \( B_c(s) \) is the covariance for the mean vector for speaker \( s \). The total variance of the acoustic frames aligned to the Gaussian component \( c \) for speaker \( s \), \( x_c(s) \) can be expressed as

\[
\text{Cov}(x_c(s), x_c(s)) = \Sigma_c + B_c(s) \tag{31}
\]

As we can see, \( B_c(s) \) serves to capture the uncertainty of the mean estimate for speaker \( s \) for Gaussian component \( c \) and \( \Sigma_c \) is assumed to capture the residual uncertainty on this component. One way to get a feeling of our model is to compare the size of subspace defined by \( UU^* \) and \( \Sigma \) to see how the total uncertainties are distributed to the subspace and Gaussian mixture covariance matrix. As we can see from (17), as the adaptation data increases (or \( \gamma_{sc} \) is large), the uncertainty for the mean estimate \( m_c(s) \) will reduce, resulting in that \( \Sigma_c \) captures all the uncertainties.

### 3.4 Mixtures of subspaces

Up to now, we have addressed the major issues involved in the subspace Gaussian mixture models, in which, the prior density distribution is constrained into a subspace of the whole model space. The prior density function we discuss is Gaussian. Here, we consider whether it will be more effective to use a mixture of Gaussians to model the priors, as it’s more likely that for a big model set (large number of speaker or HMM states for speech
recognition), the distribution of GMM parameters is multimodal rather than unimodal, which makes the idea of mixtures of subspaces rational. Using Gaussian mixtures as the prior density function for subspace GMMs involve estimating multiple (joint or disjoint) subspaces in the whole model space, and the estimation of each subspace is a PCA style least reconstruction error dimension reduction problem. We will base our discussion in this subsection on mixtures of probabilistic principle component analysis (PPCA) proposed by Tipping and Bishop [16].

By mixtures of subspaces, we now suppose the hyperparameter set is \( \{ (\pi_p, \tilde{m}_p, U_p), p = 1, \ldots, P \} \), where \( P \) is the number of subspace mixtures, and \( \pi_p, \tilde{m}_p, U_p \) are mixture weight, mean and subspace variance matrix for \( p \)-th mixture, respectively. The conditional likelihood is now given as

\[
\sum_s \log P(\mathcal{X}(s)|\theta) = \sum_s \log \left( \sum_p \pi_p P(\mathcal{X}(s)|m_p(s)) \right)
\]

where \( P(\mathcal{X}(s)|m_p(s)) \) can be calculated according to proposition 1 and \( m_p(s) = \tilde{m}_p + U_p y_p(s) \)

Learning the mixtures of subspaces model is analogous to the learning of conventional Gaussian mixture models. Firstly, for EM update the posterior of the mixture weights for speaker \( s \) can be given explicitly as

\[
R_{sp} = \frac{\pi_p P(\mathcal{X}(s)|m_i(s))}{\sum_i \pi_i P(\mathcal{X}(s)|m_i(s))}
\]

And the complete log-likelihood is

\[
\sum_s \log P(\mathcal{X}(s)|\theta) = \sum_{s,p} R_{sp} \log P(\mathcal{X}(s)|m_p(s))
\]

To update the model, the posterior distribution of the latent variable \( y_p(s) \) should be first estimated for each subspace \( U_p \). The mean and covariance \( E[y_p(s)], E[y_p(s)y_p^*(s)] \) are obtained by proposition 2. The auxiliary function for EM update of parameters is now

\[
Q(\theta, \hat{\theta}) = \sum_{s,c,p} R_{sp} \left( G^p_{\Sigma}(s,c) + E_y[H^p_{U,\Sigma}(s,c)] \right)
\]

Following the similar procedures in section 3.2, we give the update formulas as below without the derivation:

\[
\tilde{U}_c^p = \left( \sum_s R_{sp} \tilde{F}_c(s) E[y_T^T(s)] \right) \left( \sum_s R_{sp} \gamma_{sc} E[y(s)y_T^T(s)] \right)^{-1}
\]

\[
\tilde{\Sigma}_c = \sum_{s,p} S_{sc} - \frac{1}{2} \left( \tilde{F}_{scp} E[y_p^*(s)] \tilde{U}_c^p + \tilde{U}_c^p E[y_p(s)] \tilde{F}_{scp}^* \right) \sum_{s,p} \gamma_{scp}
\]

\[
\tilde{m}_{pc} = \frac{1}{\sum_{s} \gamma_{scp}} \sum_s \left( F_{scp} - \gamma_{scp} \tilde{U}_{pc} y_p(s) \right)
\]
where in the above notations, the script $p$ is introduced to indicate the portion of statistics for the subspace defined by $U_p$.

Kenny et al. [20] discussed some preliminary work to apply the mixture of subspaces in speaker recognition, but did not achieve good results. Nevertheless, we still highlight this idea for speech recognition as the number of HMM states is very large which may lead to more complex distribution in model space. In real applications, however, to constrain the model complexity, an additional parameter set should be introduced to govern the number of subspaces as well as the size of each subspace to avoid overfitting. Ghahramani and Beal [22] have investigated a similar model using variational inference.

3.5 Model extension with multiple factors

We now consider another way to extend the model, in which the Gaussian mean supervector is composed of multiple factors which lie in its individual subspace. We can express this as:

$$m(s, h) = \bar{m} + Uy(s) + Vx(s, h) + dz(s)$$

Such model is commonly used in field of speaker recognition, referred to as Joint Factor Analysis (JFA) [10]. In that context, $m(s, h)$ denotes the speaker- and channel-dependent mean supervector, and $V$ of similar size to $U$ is a matrix characterising the model variation caused by channel effects. And $d$ is a $CF \times CF$ diagonal matrix which plays the role of relevance MAP. For the task of speaker recognition, channel factors are a major reason for performance degradation since they bring much more uncertainty to the speaker model. The subspace defined by $VV^*$ can be viewed as a nuisance subspace which we would like to compensate during both model training and testing to achieve performance gains. The diagonal matrix $d$ is introduced to be asymptotical to the ML estimate when given a sufficiently large amount of training data. Splitting the model space into a speaker subspace and a nuisance subspace brings much flexibility when compensating a noisy environment and makes the model more robust.

To estimate the matrix $V$ and $d$, firstly we need to collect a set of recordings under different channels for each speaker $s$, denoted as $X(s) = \{X(s, h), h = 1, \ldots, H\}$. There are two general estimation schemes to estimate the multiple set of hyperparameters, namely, joint estimate and iterative independent estimate. For the joint estimation method, the three matrices $U$, $V$ and $d$ are updated together during the EM iteration. To describe it a little bit, we could express the subspace matrices and latent variables
as:

\[
W = \begin{pmatrix}
V & \ldots & 0 & U & d \\
0 & \ddots & 0 & \vdots & \vdots \\
0 & \ldots & V & U & d
\end{pmatrix}; \quad \vartheta(s) = \begin{pmatrix}
x(s, 1) \\
\vdots \\
x(s, H) \\
y(s) \\
z(s)
\end{pmatrix}.
\]

Then the Gaussian mean supervector for speaker \(s\) under channel \(h\) can be represented by

\[
m(s, h) = \bar{m} + W_h \vartheta(s)
\]

where \(W_h\) means the \(h\)th row of the matrix. The conditional log-likelihood of the speech data given the latent variables can be re-written as:

\[
\log P(\mathcal{X}(s)|\vartheta(s)) = \sum_{h=1}^{H} \log P(\mathcal{X}(s, h)|\bar{m} + W_h \vartheta(s))
\]

The deduction follows the similar procedure with the case of only one factor, and the results are also similar with those obtained in the above subsections. We do not present the detailed derivation formulas in order to avoid overlap with the above discussion. The reader can refer to [10] for more information.

It should be noted that to perform the joint estimation of the model, we need to calculate the expectation and covariance of the latent variables \(E[\vartheta(s)]\) and \(E[\vartheta(s)\vartheta^*(s)]\). As the dimension of \(\vartheta(s)\) is now \((HR_x + R_y + CF)\), it will involve inverting matrix \(\alpha^{-1}(s)\) with size of \((HR_x + R_y + CF) \times (HR_x + R_y + CF)\) as in equation (26). The non-diagonal entries of the matrix \(\alpha^{-1}(s)\) are expected to capture the correlations between the multiple factors. As the dimension of \(\vartheta(s)\) is considerably large, inverting such a matrix can be computationally demanding and in addition, it can suffer from data sparsity. Under such considerations, we tend to adopt the iterative independent estimation approach for such model with multiple factors, which is also termed as Gauss-Seidel method [23, 19, 24].

In this method, the speaker subspace and nuisance subspace are considered as independent (or orthogonal in matrix terminology), hence when updating one subspace parameter, we do not need to consider the role of another or leave it unchanged. To be more specific, we do not need to consider the channel issue when estimating the speaker subspace \(U\) (treat all the recordings from different channels the same for one speaker) and when we estimate the channel subspace, centre the statistics with \(\bar{m} + Uy(s)\) instead of \(\bar{m}\) in equation (12) and (13) for recordings from speaker \(s\) and then treat each recording as one speaker, following the same procedure to estimate \(V\). Finally, fix our estimate of \(U\) and \(V\) and centre the statistics with \(\bar{m} + Uy(s) + Vx(s)\) (normally on another dataset) to estimate the diagonal matrix \(d\) [19].
3.6 Point estimate vs. Uncertainty estimate

Here, we consider two model estimate approaches, namely, the point estimate and the uncertainty estimate for decoding or recognition. By a point estimate, we mean given an utterance $X(s)$ from a new enrolled speaker $s$, we estimate its mean supervector as in section 3.3, and when given a testing utterance, $X(l)$, likelihood is calculated by $\log P(X(l)|m(s))$ as in conventional likelihood calculation for GMMs.

By uncertainty estimate [18], the likelihood is given as

$$P(X(l)|X(s)) = \int P(X(l)|m(s))P(m(s)|X(s))dm$$

which means the model is marginalised out by the posterior distribution given the training data. Such approach is also normally termed as model evidence. If we just use one latent factor, say $m(s) = \bar{m} + U y(s)$, then the above equation can be rewritten as:

$$P(X(l)|X(s)) = \int P(X(l)|y(s))P(y(s)|X(s))dy$$

From proposition 2, we can get the posterior distribution $P(y(s)|X(s))$ and we can easily evaluate. However, if we used multiple latent factors, say $m(s) = \bar{m} + U y(s) + V x(s) + dz(s)$ as in the case of joint factor analysis (JFA), then the likelihood should be calculated by

$$P(X(l)|X(s)) = \int P(X(l)|H)P(H|X(s))dH$$

where we denote $H = (y, x, z)$. The exact solution requires the posterior distribution of the joint factors, and just as we discussed in the above section 3.5, it may be prohibitive to directly estimate the joint posterior distribution of the latent variables when they are high dimensional. However, one solution for this problem is turn to variational inference [15], in which, the joint posterior distribution can be approximated by factorial form which decouples the correlation among different factors, as

$$Q(H) = Q_1(y)Q_2(x)Q_3(z)$$

And the likelihood can be calculated simply by

$$\log P(X(l)|X(s))$$

$$= E_Q[\log P(X(l), H)] + \mathcal{H}(Q) + KL(Q||P)$$

where $\mathcal{H}(Q)$ denotes the entropy of $Q$ and $KL(Q||P)$ means the KL-divergence between the two distributions. From some preliminary results on speaker recognition [25], such an uncertainty estimate method only achieves notable improvement on short utterances compared to the point estimate, while for long utterances, the uncertainties of latent factors are very small and it is not very profitable to take into account the uncertainties.
Nevertheless, it will be interesting to test the two scoring methods in subspace GMM based speech recognition systems, and we expect to achieve performance gains by taking into the consideration of uncertainties as the training data is much more sparse compared with the model complexity in speech recognition scenario.

Finally, the way to collect the Baum-Welch statistics may affect the final recognition accuracy [19]. It is reported that using a speaker-dependent model to collect the statistics may bring much performance gains in some cases (typically for simple models which do not contain the speaker factor), but may also cause notable performance degradation (when using multiple factors). Hence, it’s generally agreed that collecting the statistics from a background GMM turns out to be more robust. We will be aware these issues in our work for subspace GMM based speech recognition systems (especially for speaker adapted systems).

4 Subspace GMM for Speech Recognition

The idea of the subspace GMM can be easily applied to HMM-based speech recognition system, and theoretical and practical development have been done in the 2009 JHU summer workshop by Povey, Burget, et al [7, 26]. From their results, the subspace GMM based system can achieve comparable performance with considerably fewer parameters when compared to a conventional modelling approach. In addition, since such an approach incorporates the prior knowledge into the models, it possible to utilise the out-of-domain data to train the shared parameters. An example is the multilingual experiments in [27]. In this section, we will explore the application of subspace method to speech recognition based on their work together with some additional discussions.

4.1 Basic model description

Applying the subspace GMMs to speech recognition is done by tying a subset of parameters of all the HMM state GMMs in a shared subspace [7]. The basic model can be expressed as follows:

\[
P(x(t)|j) = \sum_{c=1}^{C} w_{jc}N(x(t); \mu_{jc}, \Sigma_c) \tag{48}
\]

\[
\mu_{jc} = m_c + U_c y(j) \tag{49}
\]

\[
w_{jc} = \frac{\exp w_{c}^* y(j)}{\sum_{c'=1}^{C} \exp w_{c'}^* y(j)} \tag{50}
\]

where \( w = [w_1^*, \ldots, w_C^*]^T \) is a \( CR \times 1 \) supervector defining the variation space for Gaussian mixture weights (we still suppose the rank of \( U \) is \( R \)). \( P(x(t)|j) \) denotes the conditional likelihood for state \( j \). We now refer the subspace defined by matrix \( U \) as the phone subspace as it models the correlations among all the phone states in HMMs.
Compared with our discussion for subspace GMMs in section 3, the likelihood calculation and model update here can follow generally the same procedures just bearing in mind that the state model \( j \) now plays the role for speaker model \( s \), and the posterior of Gaussian mixture component should be scaled by the state posterior probability, i.e.

\[
\gamma_{jc}(t) = p(j, c|x(t)) = \frac{\gamma_{j}(t) p(x(t)|j, c)}{p(x(t)|j)}
\]

where \( \gamma_{j}(t) = p(j|x(t)) \) which can be obtained by standard forward-backward or Viterbi algorithm. In the subspace GMM we discussed in section 3, we kept the Gaussian mixture weights the same for all the speaker models. However, for speech recognition using subspace GMM, it was found from the preliminary results that relaxing the constraint to allow the weights to be state dependent as equation (50) shows can bring performance gains [7]. As the latent variable \( y(j) \) is now also governed by the weight projection matrix \( w \), we have to modify the posterior distribution of \( y(j) \) in proposition 2 to account for the weight variations. The posterior distribution of \( y(j) \) is now given as:

\[
\log P_U \left( y(j)|\mathcal{A}(j) \right) = -\frac{1}{2} y^*(j) \alpha(j) y(j) + y^*(j) \beta(j) + Q(w_j) + \text{const}
\]

where \( Q(w_j) \) is the auxiliary function for the weight projection matrix, defined as

\[
Q(w_j) = \sum_c \gamma_{jc} \log w_{jc}
\]

\[
= \sum_c \gamma_{jc} \left( w_{c}^* y(j) - \log \sum_i \exp(w_{i}^* y(j)) \right)
\]

\[
= \text{const} + \sum_c \gamma_{jc} \left( w_{c}^* y(j) - \log \frac{\sum_i \exp(w_{i}^* y(j))}{\sum_i \exp(\tilde{w}_{i}^* \tilde{y}(j))} \right)
\]

where \( \tilde{y}(j) \) and \( \tilde{w} \) are the 'old' value. The denominator is introduced for using 'old' estimate of weight to represent the auxiliary function which we will see later. As there are sum operation in the log function, we have to use some approximation approach to linearise the variable \( y(j) \). Here we first use the inequality \( 1 - (x/\bar{x}) \leq -\log x/\bar{x} \) to obtain

\[
Q(w_j) = \text{const} + \sum_c \gamma_{jc} \left( w_{c}^* y(j) - \frac{\sum_i \exp(w_{i}^* y(j))}{\sum_i \exp(\tilde{w}_{i}^* \tilde{y}(j))} \right)
\]

And then \( \exp(x) \) is further approximated by a quadratic function, i.e.

\[
\exp(x) \simeq \exp(x_0) \left( 1 + (x - x_0) + \frac{1}{2} (x - x_0)^2 \right) = \text{const} + \exp(x_0) \left( x(1 - x) + \frac{1}{2} x^2 \right)
\]
Then, it will lead to

$$Q(w_j) = const + \sum_c \gamma_{jc} \left[ w_c^* y(j) - \sum_i C \exp(w_i^* \bar{y}(j)) \left( w_i^* y(j)(1 - w_i^* \bar{y}(j)) \right) \right]$$

$$+ \frac{1}{2} y^*(j) w_i w_i^* y(j)$$

$$= const + \sum_c \gamma_{jc} \left[ w_c^* y(j) - \sum_i C \bar{w}_{jc} \left( w_i^* y(j)(1 - w_i^* \bar{y}(j)) \right) \right]$$

$$+ \frac{1}{2} y^*(j) w_i w_i^* y(j)$$

$$= const - \frac{1}{2} y^*(j) \left( \sum_c \gamma_{jc} w_c w_c^* y(j) + y^*(j) \sum_c w_c(\gamma_{jc} - \gamma_{jc}(1 - w_c^* \bar{y}(j))) \right)$$

Hence, the posterior probability of $y(j)$ is still a quadratic function if expressed in log format, i.e.

$$\log P(y(j)|\mathcal{Y}(j)) = -\frac{1}{2} y^*(j) \hat{\alpha}(j)y(j) + y^*(j) \hat{\beta}(j) + const$$

(52)

where $\hat{\alpha}(j)$ and $\hat{\beta}(j)$ are now defined as:

$$\hat{\alpha}(j) = \alpha(j) + \sum_c \gamma_{jc} w_c w_c^*$$

$$= I + \sum_c \gamma_{jc}(U_c^* \Sigma_c^{-1} U_c + w_c w_c^*)$$

$$\hat{\beta}(j) = \beta(j) + \sum_c w_c(\gamma_{jc} - \gamma_{jc}(1 - w_c^* \bar{y}(j)))$$

(53)

$$= \sum_c \left[ U_c^* \Sigma_c^{-1} F_{jc} + w_c(\gamma_{jc} - \gamma_{jc}(1 - w_c^* \bar{y}(j))) \right]$$

We can conclude that by introducing the weight projection matrix as in equation (50), the posterior distribution of $y(j)$ is still Gaussian with mean $\hat{\alpha}^{-1}(j) \hat{\beta}(j)$ and covariance $\hat{\alpha}^{-1}(j)$. Hence the conclusions we get in section 3 still hold in this case. Although we derive the conclusion with some approximations, as the number of parameters in phone subspace is far large than that in weight projection matrix, we assume the estimation of $y(j)$ is dominated by mean parameters rather than weights. Hence the approximations we made will cause little affect to the distribution of $y(j)$. The update formulas for the weight projection matrix $w$ together with proof is given in appendix B.

4.2 Modelling the speaker subspace

Speaker variation is one of major causes of performance degradation of speech recognition system. In order to reduce the mismatch between the models and acoustic data from a
particular speaker, the most popular way is the MLLR-based speaker adaptation \cite{5,28}. However, in the framework of subspace GMMs, we can account for speaker variation at the model level where a subspace capturing the speaker variability is incorporated into the model, just like the role of channel subspace in speaker recognition which we discussed in section 3.5. To be specific, the $c$th Gaussian component mean which is made speaker-dependent is now given as:

$$
\mu_{jc}(s) = \bar{m}_c + U_c y(j) + V_c x(s) \tag{54}
$$

where $V$ is a low rank rectangular matrix defining the speaker subspace, and $x(s)$ is the attribute in the subspace for speaker $s$ which is also standard Gaussian distributed. Note that $y(j)$ is a state dependent latent variables whereas $x(s)$ is a speaker dependent. Given the low rank of $V$, the degree of freedom for speaker adaptation in this framework is quite small.

The model estimation procedure can follow exactly what we discussed in section 3.5, with the only difference being that the posterior of Gaussian component is scaled by the state posterior as equation (51) shows. During decoding, two approaches of scoring schemes, namely, point estimate and uncertainty estimate discussed in section 3.6 could be explored and compared. We believe performance gains may be achieved by taking the uncertainties of the latent factors in phone and/or speaker subspace into consideration, as the acoustic data aligned to a individual state is very limited.

Another interesting work in this area is to compare and combine the conventional speaker adaptation techniques like MLLR with speaker subspace space adaptation. Although we can still incorporate MLLR based adaptation into this framework without any further modification, since we can express the Gaussian component means explicitly, more efficient approach to combine the two methods is necessary for further performance gains. In \cite{7}, the authors found speaker subspace adaptation alone did not achieve comparable performance with CMLLR, while combining the two (first perform feature level MLLR and then speaker subspace adaptation follows) can obtain some additive performance improvement. We will explore the issues in future works.

4.3 Model extension with sub-states and sub-models

We can extend the proposed model with sub-states and sub-models to better fit the training data \cite{7}. The idea of sub-states to use a mixture of latent variables in the phone
subspace, i.e.

\[ P(x|j) = \sum_{m=1}^{M_j} c_{jm} \sum_{c=1}^{C} w_{jmc} N(x; \mu_{jmc}, \Sigma_c) \]  

(55)

\[ \mu_{jmc} = \tilde{m}_c + U_{c} y(jm) + V_{c} x(j) \]  

(56)

\[ w_{jmc} = \frac{\exp w^*_c y(jm)}{\sum_{c'} \exp w^*_c y(jm)} \]  

(57)

where \( \sum_{m=1}^{M_j} c_{jm} = 1 \). The model with sub-states increases the degrees of freedom that a state can vary given the training data, and it can be viewed as a mixture of subspace GMMs for each phone state. We can rewrite equation (55) as

\[ P(x|j) = P(x|y(j1), \ldots, y(j M_j)) = \sum_{m=1}^{M_j} c_{jm} P(x|y(jm)) \]

where each latent variable \( y(jm) \) is Gaussian distributed constrained in the same subspace defined by matrix \( U \) for all \( m = 1, \ldots, M_j \). The priors of these latent variables are also standard Gaussian and the posteriors can be estimated by the same way as proposition 2. The advantage of sub-states is that they can increase the predictive power of each state by only a small number of additional parameters (as the dimension of latent variables \( y(jm) \) is normally only in tens). However, in the meantime, as all the latent variables \( y(jm) \) are constrained in the same subspace, the model may reach its capacity limits quickly by sub-state splitting.

The idea of the sub-model extension can be viewed as a mixture of subspaces which we considered in section 3.4, in which the model can be rewritten as

\[ P(x|j) = \sum_{k=1}^{K} c_{jk} \sum_{c=1}^{C} w_{jkc} N(x; \mu_{jkc}, \Sigma_c) \]  

(58)

\[ \mu_{jkc} = \tilde{m}_{k,c} + U_{kc} y(jk) + V_{c} x(j) \]  

(59)

\[ w_{jkc} = \frac{\exp w^*_{kc} y(jk)}{\sum_{c'} \exp w^*_{kc'} y(jk)} \]  

(60)

where \( \sum_{m=1}^{M_j} c_{jm} = 1 \). As discussed in section 3.4, using mixtures of subspace can extend the unimodal assumption of GMM parameters into multimodal which is a more natural assumption for complicated distributions. We will highlight this idea in our future work as on the one hand, it is indeed reasonable to assume that the Gaussian mixture means of thousands of HMM states are multimodal rather than unimodal, and on the other hand, this is also one way to increase the degree of freedom of each state for better predictive power. Refer to section 3.4 for a more detailed discussion about mixtures of
subspace. In addition, the ideas of sub-models and sub-states can be put together as they split the model at different levels.

It’s also necessary to note that in both of the extended models, we do not perform any modification of the speaker subspace, since we believe the variation of speakers in the model space is much simpler compared with context dependent phones. Finally, the introduction of sub-states and sub-models does not bring much change to the model training and adaptation procedure, with the only difference being that the statistics should be split according to sub-states or sub-models.

4.4 Discriminative training

Discriminative training is an important technique for large vocabulary speech recognition system to achieve state-of-the-art performance [29, 30, 31, 32]. The maximum mutual information (MMI) criteria commonly used for discriminative HMM training is to maximise the posterior of the model rather than the likelihood of the data as in maximum-likelihood estimate (MLE) criteria. The objective function for MMI can be expressed as

$$F_{MMI}(\theta) = \sum_{r=1}^{R} \log P_{\theta}(M_{s_r}|X_r)$$

$$= \sum_{r=1}^{R} \log \frac{P_{\theta}(X_r|M_{s_r})P(s_r)}{\sum_s P_{\theta}(X_r|M_s)P(s)}$$

(61)

where $M_{s_r}$ is the model corresponding to the correct transcription $s_r$ for the $r_{th}$ spoken utterance $X_r$, and $P(s_r)$ is the language model for utterance $r$. In practise, a scale factor $k$ is always included in the objective function to balance the acoustic model and language model probabilities, i.e.

$$F_{MMI}(\theta) = \sum_{r=1}^{R} \log \frac{P_{\theta}^k(X_r|M_{s_r})P^k(s_r)}{\sum_s P_{\theta}^k(X_r|M_s)P^k(s)}$$

(62)

Discriminative model estimation is carried out using the weak-sense auxiliary function of MMI defined as [32]

$$G(\theta, \hat{\theta}) = G^{num}(\theta, \hat{\theta}) - G^{den}(\theta, \hat{\theta}) + G^{sm}(\theta, \hat{\theta})$$

(63)

where $G^{num}(\theta, \hat{\theta})$ is the conventional auxiliary function for the numerator in equation (62) (which is the standard maximum-likelihood estimate (MLE) objective function) i.e.

$$G^{num}_r(\theta, \hat{\theta}) = \sum_{j_r=1}^{J_r} \sum_c \gamma_{j_r,c} \log (X_r|M(j_r,c))$$

(64)
where \( \{ j_r, 1, \ldots, J_r \} \) is the HMM state sequence for the correct transcription of \( r_{th} \) sentence, and \( G_r^{num}(\theta, \tilde{\theta}) = \sum_r G_r^{num}(\theta, \tilde{\theta}) \) (we do not consider the scale factor \( k \) now). \( \gamma_{j_r,c} \) is the posterior of state and component pair \((j_r, c)\) based on the 'old' model parameters. \( G_{den}(\theta, \tilde{\theta}) \) is of similar definition except that the state sequence now ranges over all the competing transcriptions. \( G_{sm}(\theta, \tilde{\theta}) \) is the smoothing function to ensure the final auxiliary function is convex.

For the subspace GMMs based speech recognition system, performing discriminative training involves another state level latent variable set which we denote as \( H(j) \) for state \( j \) \((H(j)\) may contain multiple latent variables in general cases, e.g. \( H(j) = \{ y(j), x(j), z(j) \} \) as for joint factor analysis case) in the auxiliary function. Take the numerator for example, the auxiliary function should be modified as

\[
G_r^{num}(\theta, \tilde{\theta}) = \sum_{j_r=1}^{J_r} \sum_c \gamma_{j_r,c} \int P_{\tilde{\theta}}(H(j_r)|X_r) \log (X_r, H(j_r)|M(j_r,c)) dH \\
= \sum_{j_r=1}^{J_r} \sum_c \gamma_{j_r,c} E_{H(j_r)} [\log (X_r, H(j_r)|M(j_r,c))]  
\]

(65)

With similar definition for \( G_{\Sigma}(j_r, c) \) and \( H_{U,\Sigma}(j_r, c) \) with equation (15) and (16), we can rewritten the above equation as

\[
G_r^{num}(\theta, \tilde{\theta}) \\
= \sum_{j_r=1}^{J_r} \sum_c E_{H(j_r)}[G_{\Sigma}(j_r, c) + H_{U,\Sigma}(j_r, c)] \\
= \sum_{j_r,c} \left\{ \frac{1}{2} \left[ \gamma_{j_r,c} \log (2\pi)^{-F} |\Sigma_c^{-1}| - \text{tr} (\Sigma_c^{-1} S_{j_r,c}) \right] \\
+ E_{H(j_r)} \left[ \gamma_{j_r,c} \log w_{j_r,c} - \frac{1}{2} H^*(j_r) U_c^* \Sigma_c^{-1} U_c H(j_r) + H^*(j_r) U_c^* \Sigma_c^{-1} F_{j_r,c} \right] \right\}  
\]

(66)

We now just consider updating \((U, \Sigma)\) only during discriminative training for simplicity (however, it’s straightforward to extend to the whole parameter set). In this case, the expression (66) can be rewritten as

\[
G_r^{num}(\theta, \tilde{\theta}) = \text{const} + \sum_{j_r,c} \left\{ \frac{1}{2} \log |\Sigma_c^{-1}| - \text{tr}(\Sigma_c^{-1} S_{j_r,c}) \right\} \\
- \frac{1}{2} \text{tr} \left( U_c^* \gamma_{j_r,c} \Sigma_c^{-1} U_c E[H(j_r)H^*(j_r)] \right) + E[H^*(j_r) U_c^* \Sigma_c^{-1} F_{j_r,c}]  
\]

(67)

By taking into the term of \( G_r^{den}(\theta, \tilde{\theta}) \) which of the similar form of expression (67) except
that the statistics are collected from competing lattices, we can obtain
\[
\frac{\partial (G^\text{num}_r - G^\text{den}_r)}{\partial U_c} = \sum_{j_r} \left( -U_c \gamma^\text{num}_{j_r,c} E[H(j_r)H^*(j_r)] + \bar{F}^\text{num}_{j_r,c} E[H^*(j_r)] \right) \\
- \sum_{j'_r} \left( -U_c \gamma^\text{den}_{j'_r,c} E[H(j'_r)H^*(j'_r)] + \bar{F}^\text{den}_{j'_r,c} E[H^*(j'_r)] \right)
\]

We denote \((j = 1, \ldots, J)\) as the union of all the numerator and denominator states, then the update formula for \(U_c\) is
\[
\bar{U}_c = \left( \sum_j (F^\text{num}_{jc} - \bar{F}^\text{den}_{jc}) E[H^*(j)] \right) \left( \sum_j (\gamma^\text{num}_{jc} - \gamma^\text{den}_{jc}) E[H(j)H^*(j)] \right)^{-1}
\]

By the same procedure, we can obtain the update formula for \(\Sigma_c\) is as
\[
\bar{\Sigma}_c = \frac{\sum_j (S^\text{num}_{jc} - S^\text{den}_{jc} - \frac{1}{2} (M(j) + M^*(j)))}{\sum_j (\gamma^\text{num}_{jc} - \gamma^\text{den}_{jc})}
\]

where \(M(j)\) is defined as
\[
M(j) = (F^\text{num}_{jc} - \bar{F}^\text{den}_{jc}) E[H^*(j)] \bar{U}_c^*
\]

When we derived the expression (68) and (69), we dropped the smoothing term \(G^\text{sm}(\theta, \hat{\theta})\).

We will explore the suitable smoothing term in future works. Finally, to apply the ideas such as multiple factors, uncertainty estimate and mixtures of subspaces, etc., into discriminative training seems not trivial, and this may also be one of the focuses in our future works.

5 Experimental results

In this section, we will present some experimental results of subspace GMMs on speech recognition. To set up the subspace GMM based system, we used the toolkits developed by Povey, et al. in the 2009 JHU summer workshop [7]. And for the baseline system, the HTK speech recognition toolkit [33] was used for system building. In section 5.1, we will first describe the dataset and system configurations. Experimental results as well as some discussion are given in section 5.2.

5.1 System setup

All the experiments below are based on the 5K task of Wall Street Journal (WSJ) corpus [34], in which, the training data contains 7193 utterances with a total duration
of about 14 hours after removing silence. For testing, we used a subset of the WSJ1 5K development data obtained by deleting sentences with out-of-vocabulary (OOV) words which resulted in about 200 sentences from 10 speakers. We used the standard 5k non-verbalised closed bigram language model (LM) for all the testing.

For the acoustic features, 12 dimensional MFCC coefficients plus C0 were extracted, and cepstral mean subtraction (CMS) on sentence level was applied. First and second derivatives were computed and appended to each feature vector, resulting in 39 dimensional feature vectors. We did not carry out any other feature level compensation techniques such as VTLN. We trained the cross-word triphone acoustic models with about 4k tied states by HTK as our baseline, and the number of monophones we used was 39 without silence and short pause as in [35]. Each HMM state was modelled by 16 component GMMs with diagonal covariance matrices.

As for the subspace GMM based systems, a seed Universal Background Model (UBM) of 256 Gaussian components with full covariance matrices was trained on the whole development data. The phone subspace projection matrix $U$ was initialised as in [7] in which, the first column was actually the UBM mean supervector. The state specific vector $y(j)$ in equation (49) was initialised to be $e = [1, 0, \ldots, 0]$. In our systems, we do not include the offset term $\bar{m}$ as expression (49) shows. The subspace GMM system was then built by substituting GMMs in the baseline system with subspace GMMs. After initialisation, several EM iterations were performed until convergence.

5.2 Results and discussion

The experimental results are given in table 1, in which, the SGMM-1 systems indicates that the rank of phone subspace is set to be 40, while that for SGMM-2 systems is 60. We also compared the results of subspace GMM based systems with different number of sub-states as showed in table 1. Note however, the number of sub-states is an average for each triphone state, and some states may have more sub-states while others may have fewer. See details of sub-states splitting in [7]. We also present the number of parameters for each system to illustrate the size of our models. For subspace GMM based systems, the phone projection matrix and global Gaussian covariance matrices normally will account for more than 80% of the parameters. As we can see for the table, the number of parameter in subspace GMM systems is about a quarter of that of the baseline system.

Among the subspace GMM based systems with different configurations, the best we get is the one of 40 dimensional phone subspace and 4 sub-states on average for each triphone state, with the word error rate (WER) of 25.5%. However, compared with the baseline system, the performance of the subspace GMM based systems is much worse, which is about 10% performance degradation. As for the reasons, we fond the insertion errors would account for nearly half of the word error rate (WER), which is showed by the last column of table 1. We assume the large insertion error may due to the globally
shared covariance matrices $\Sigma$. In order to capture the residual variance in all the states, the globally shared $\Sigma$ may be considerable large which lead to the inferior discriminative ability between different states. We will look into this phenomena further in future works.

Table 1: Comparison of the results between HTK and SGMM based systems.

<table>
<thead>
<tr>
<th>$\Phi$</th>
<th>#state</th>
<th>#sub-state</th>
<th>#rank(U)</th>
<th>#para</th>
<th>WER(%)</th>
<th>ins(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTK</td>
<td>4k</td>
<td>-</td>
<td>-</td>
<td>5.0M</td>
<td>16.2</td>
<td>1.7</td>
</tr>
<tr>
<td>SGMM-1</td>
<td>8k</td>
<td>2</td>
<td>40</td>
<td>1.02M</td>
<td>13.8</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>16k</td>
<td>4</td>
<td>40</td>
<td>1.08M</td>
<td>13.4</td>
<td>1.4</td>
</tr>
<tr>
<td>SGMM-2</td>
<td>4k</td>
<td>1</td>
<td>60</td>
<td>1.24M</td>
<td>14.8</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td>8k</td>
<td>2</td>
<td>60</td>
<td>1.32M</td>
<td>14.7</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>16k</td>
<td>4</td>
<td>60</td>
<td>1.40M</td>
<td>14.0</td>
<td>1.2</td>
</tr>
</tbody>
</table>

It is also apparent from the results that, sub-state splitting can improve the system performance for both kinds of SGMM systems, and in particular, splitting the triphone states into 4 sub-states on average can bring about more than 5% relative performance gains for both SGMM-1 and SGMM-2. These results are consistent with those reported in [14]. Meanwhile, increasing the rank of phone subspace matrix from 40 to 60 does not as expected, whereas on contrary, slight performance degradation was observed. To look into this phenomena, we calculated the size of phone subspace for each system.

Table 2: Comparison of the size between the phone subspace and Gaussian mixture covariance matrices.

<table>
<thead>
<tr>
<th>rank(U)</th>
<th>sub-states</th>
<th>$tr(UU^*)$</th>
<th>$tr(\Sigma)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>s=1</td>
<td>165.236</td>
<td>80,553</td>
</tr>
<tr>
<td></td>
<td>s=2</td>
<td>211.067</td>
<td>76,734</td>
</tr>
<tr>
<td></td>
<td>s=4</td>
<td>345.231</td>
<td>72,760</td>
</tr>
<tr>
<td>60</td>
<td>s=1</td>
<td>230.442</td>
<td>79,444</td>
</tr>
<tr>
<td></td>
<td>s=2</td>
<td>336.081</td>
<td>75,371</td>
</tr>
<tr>
<td></td>
<td>s=4</td>
<td>639.660</td>
<td>70,480</td>
</tr>
<tr>
<td>UBM</td>
<td>-</td>
<td>-</td>
<td>113,890</td>
</tr>
</tbody>
</table>

Table 2 provides a comparison of the trace of the phone subspace and the total size of Gaussian component covariance matrices with different configurations. Not surprisingly, with the triphone state splitting, the trace of the phone subspace increases while that for Gaussian mixture covariance matrices decrease accordingly, which means as we split the sub-states (equivalent with increase the number of states or classes), the phone subspace will account for more variabilities in the acoustic data while the variability captured by global covariance matrices shrinks in the meantime. In addition, the trace of the phone subspace matrix with rank 60 is considerably large than that of phone subspace with rank 40 under the same number of sub-states. Just as we discussed in section 3.3, for MAP estimate of the model, the total variance of the acoustic data aligned to the $c_{th}$
Gaussian component can be approximated by the sum of the two factors as

$$\text{cov}(x_c(j), x_c(j)) = \Sigma_c + U_c\alpha^{-1}(j)U_c^*$$

where we have replaced the script $s$ in section 3.3 to $j$ to indicate it’s for $j_{th}$ state. In particular, the term $U_c\alpha^{-1}(j)U_c^*$ denotes the variance of mean estimate of the state specific vector $y(j)$ as in equation (49), namely, $\text{cov}(\mu_c(j), \mu_c(j)) = U_c\alpha^{-1}(j)U_c^*$. As it show in table 2, the large size of phone subspace can reduce the total variance but on the other hand, it may also increase the bias of the Gaussian mean estimated from the training data. Perhaps this is one of major reasons that increasing the size of subspace may not improve the system performance. And the results also motivate us to consider the uncertainty estimate for complex models to restrict the possible bias.

![Figure 1: Plots of eigenvalues for phone subspace matrices of SGMM-1 and SGMM-2, as well as the mean and variance for state specific vectors calculated by each dimension.](image)

It’s also interesting to note that, in these subspace GMM based systems, we did not include the offset term $\tilde{m}$ for Gaussian mean supervectors as we discussed in section 3. However, from the results, it seems necessary to include such an offset explicitly in the model. As we can see from figure 1, the eigenvalues of the phone subspace matrix for both
kinds of systems do not decrease exponentially, rather, there is a sharp drop between the first and second largest eigenvalues for all the systems. If we look into the state specific vectors, we can find the first dimension of $y(j)$ for all state $j$ is approximately 1 with very little variance, as it showed from the second line of figure 1. Bear in mind that we initialised the first column of phone subspace matrix by a UBM mean supervector, and during the subsequent EM iterations, this column was treated more like the offset than one dimensional space. If we drop the first column of the phone subspace matrix, we would get the eigenvalues like figure 2 shows. Hence, in future works, we would prefer to include the offset term explicitly and update it during EM iterations as expression (25) shows to build the model.

6 Future works

From the experiments described in the above section, we can get some insight into the subspace GMMs based speech recognition approach, and from the results, improved accuracy with low cost is achievable by this new approach. However, the research work in this direction is still at a very early stage, and ideas such as uncertainty estimate, mixtures of subspaces and discriminative training in this framework outlined in this paper could be investigate further in future works. Here we outline a work plan for short term and long term research goals as follows.
6.1 Short-term plan

The short term plan here mainly means the work for the next six months, and during this period, we will base our experiment on WSJ 20K. Since the plan is for the foreseeable future, we prefer to give a comparatively detailed plan and expect to generally go as the plan.

1. Explore the idea of mixtures of subspaces.

The idea of mixtures of subspaces is actually equivalent to that of sub-models in [26] with some extensions. However, Povey did not go deeply into this idea and did not report any experiments. However, we are interested to explore it further. As we discussed in section 3, the mixtures of subspaces relax the unimodal assumption of the distribution of GMM mean supervectors to multimodal, which we believe is a more natural assumption and also likely to improve performance of SGMM based speech recognition systems. Even though such approach have not been well explored in the speech field except one attempt in [20] for channel compensation in speaker recognition, similar ideas such as mixtures of PPCA [16] and mixtures of factor analysers [22] have been comparatively well investigated in the machine learning field, to which we will refer in future work.

2. Investigate the idea of multiple factors and different combinations.

In our experiments, we just adopted a single phone subspace to describe the GMM mean supervectors which and be expressed as \( m(j) = U y(j) \) for \( j \)th HMM state. However, in future work, we are interested to test the model as \( m(j) = \tilde{m} + U y(j) + dz(j) \), namely, an offset and a diagonal loading term is introduced to model each GMM mean supervector. This model is actually a standard factor analysis model and if \( d = \sigma I \), it reduces to probabilistic PCA (PPCA) model [8]. The diagonal loading term is introduced to capture the residual variance of GMM mean supervectors which’s not covered by the subspace and in speaker recognition, it have found this term is useful to model the long duration speech utterances [19]. We will look into it in the task of speech recognition.

3. Compare the two scoring methods, i.e., point estimate and uncertainty estimate.

In section 3.6, we have discussed these two methods to evaluate our models and from it’s application in speaker recognition [36, 25] we can generally conclude that the uncertainty estimate is superior when the training data is more sparse compared with model complexity which sounds quite reasonable. In speech recognition, we expect the uncertainty estimate can achieve performance gains and more importantly, provide us room to build comparatively complex models such as mixtures of subspaces or multiple factors based system.

4. Speaker adaptation by speaker subspace and/or CMLLR.

In [7], the authors have conducted some experiments of speaker adaptive training (SAT) [37] by Constrained MLLR (CMLLR) and speaker subspace adaptation or both in this framework. From their results, CMLLR based SAT system outperformed that based
on speaker subspace adaptation which did bring performance gains though. However, speaker subspace based adaptation on top of feature space MLLR seems to achieve additive improvement from their results. In future work, we will look into this problem further and aim to find a more efficient approach to combine the conventional and the subspace based speaker adaptation into this framework.

6.2 Long-term plan

The long term plan mainly indicates the work for second and third year Ph.D. We just point out the possible directions for the work here.

1. Discriminative training for subspace GMM based speech recognition.

Discriminative training is an important technique to achieve state-of-the-art performance for large vocabulary speech recognition, and as the subspace GMM based method is a novel approach for speech recognition, little work has been done to explore discriminative training in this new framework. In [38], Povey generally discussed this problem but without deep investigation and experimental results. In section 4.4, we also roughly outlined the basic principles for discriminative training section for subspace GMM based speech recognition, yet insightful research to this problem has not been carried out, and in addition, whether the discriminative training on top of multiple subspaces, multiple factors and uncertainty estimate which have been introduced above could achieve further performance gains is not clear yet. We will put the work in this direction as one of our focuses in second year Ph.D.

2. Subspace GMM based speech recognition on conversation and meeting data.

Another long term goal for the Ph.D work is to look into issues of applying this new framework of speech recognition to conversation and meeting dataset such as Switchboard and AMI data [39]. In particular, we are interested in the comparison and combination with microphone array based speech recognition [40].

7 Conclusion

In this paper, we have discussed the principles of subspace Gaussian mixture models for probabilistic density estimation and also reviewed some possible extensions of applying this approach to probabilistic model estimation such as multiple factors, model adaptation, mixtures of subspace and uncertainty estimate, etc., based on its application in speaker recognition. Besides, we have also presented relatively detailed discussion of issues for applying this method into speech recognition systems, and addressed the general framework of this new speech recognition approach based on some other researchers’ work. Finally, some preliminary experimental results of subspace GMM based speech
recognition are given based on WSJ0 corpus and we also outlined the plan for short term and long term Ph.D research work.

References


A Proofs of propositions in section 3

A.1 Proof of proposition 1

Proof. The conditional likelihood of acoustic data with Baum-Welch alignment given a Gaussian mixture model can be expressed as:

\[
\log P(\mathcal{X}(s)|w, \mu, \Sigma) = \sum_{t,c} \gamma_{sc}(t) \left\{ \log w_c + \log \mathcal{N}(x(t)|\mu_c, \Sigma_c) \right\}
\]

Replace \( \mu_c \) with \( \bar{m}_c + U_c y(s) \) and carry some rearrangement, we will have

\[
\log P(\mathcal{X}(s)|y(s)) = \sum_{t,c} \gamma_{sc}(t) \left\{ \log w_c + \frac{1}{2} \log (2\pi)^{-F|\Sigma_c^{-1}|} - \frac{1}{2} \text{tr}(\Sigma_c^{-1}\bar{x}_c(t)\bar{x}_c^*(t)) \\
+ y^*(s)U_c^*\Sigma_c^{-1}\bar{x}_c(t) - \frac{1}{2} y^*(s)U_c^*\Sigma_c^{-1}U_c y(s) \right\}
\]

\[
= \sum_{c=1}^C \{ G_{\Sigma}(s,c) + H_{U,\Sigma}(s,c) \}
\]

Just as required. \( \square \)
A.2 Proof of proposition 2

Proof. To prove the above proposition, it’s suffice to show that \( \log P_{U,\Sigma}(y(s)|X(s)) \) is a quadratic function of \( y(s) \) which can be expressed as

\[
\log P_{U,\Sigma}(y(s)|X(s)) = -\frac{1}{2} y^*(s) \alpha(s) y(s) + y^*(s) \beta(s) + \text{const}
\]  

(71)

As we already know the prior density of \( y(s) \) is standard normal Gaussian, by the Bayesian theorem, we can obtain:

\[
\log P_{U,\Sigma}(y(s)|X(s)) \propto \log P(X(s)|y(s), \Sigma) \mathcal{N}(y(s); 0, I)
\]

\[
\propto \sum_{c=1}^{C} \left\{ -\frac{1}{2} y^*(s) U^*_c \Sigma^{-1}_c y(s) + y^*(s) U^*_c \Sigma^{-1}_c F_{sc} \right\} - \frac{1}{2} y(s)^* y(s)
\]

\[
= -\frac{1}{2} y^*(s) \left\{ I + \sum_{c=1}^{C} U^*_c \Sigma^{-1}_c U_c \right\} y(s) + y^*(s) \sum_{c=1}^{C} U^*_c \Sigma^{-1}_c F_{sc}
\]

\[
= -\frac{1}{2} y^*(s) \alpha(s) y(s) + y^*(s) \beta(s)
\]

Hence, the proposition is valid. \( \square \)

A.3 Proof of proposition 3

Proof. From proposition 1, we can express the marginal likelihood as follows:

\[
\log P_{U,\Sigma}(X(s)) = \log \int P_{U,\Sigma}(X|y(s)) \mathcal{N}(y(s)|0, I) dy
\]

\[
= \sum_{c=1}^{C} G_{\Sigma}(s, c) + \log \int \exp \left( \sum_{c=1}^{C} H_{U,\Sigma}(s, c) \right) \mathcal{N}(y(s)|0, I) dy
\]

We now take out the term in the initial function and with some rearrangement we will
obtain

\[
\exp \left( \sum_{c=1}^{C} H_{U,\Sigma}(s, c) \right) \mathcal{N}(y(s)|0, I)
\]

\[
= \frac{1}{(2\pi)^{F/2}} \exp \left( -\frac{1}{2} y^*(s) \left( I + \sum_{c=1}^{C} U_{c}^{*} \gamma_{sc} \Sigma_{c}^{-1} U_{c} \right) y(s) + y^*(s) \left( \sum_{c=1}^{C} U_{c}^{*} \Sigma_{c}^{-1} F_{sc} \right) \right)
\]

\[
= \frac{1}{(2\pi)^{F/2}} \exp \left( -\frac{1}{2} y^*(s) \alpha y(s) + y^*(s) \beta \right)
\]

\[
= \frac{1}{(2\pi)^{F/2}T} \exp \left( -\frac{1}{2} (y(s) - \alpha^{-1}\beta)^* \alpha (y(s) - \alpha^{-1}\beta) + \frac{1}{2} \beta^* \alpha^{-1}\beta \right)
\]

\[
= |\alpha^{-1}|^{1/2} \exp \left( \frac{1}{2} \|\alpha^{-1}\beta\|^2 \right) \mathcal{N}(y(s)|\alpha^{-1/2}\beta, \alpha^{-1})
\]

where we have dropped \(s\) in \(\alpha, \beta\) to make the expression uncluttered. By marginalising out \(y(s)\) by integration and take log expression of the result, we can get equation(20).

\[\square\]

A.4 Proof of proposition 4

\[\textbf{Proof.}\] To prove equation (23), we can express the auxiliary function for the hyper-parameter \(U\) as follows:

\[
Q(U, \hat{U}) = \sum_{s=1}^{S} \int P(y(s)|X(s), \hat{\theta}) \log P(X(s), y(s)|\theta) dy = \sum_{s,c} G\Sigma(s, c) + \sum_{s,c} E_y[H_{U,\Sigma}(s, c)]
\]  

(72)

where \(G\Sigma(s, c)\) and \(H_{U,\Sigma}(s, c)\) are defined as equation (15) and (16). \(E_y[H_{U,\Sigma}(s, c)]\) means the expectation of \(H_{U,\Sigma}(s, c)\) with respect to the distribution of \(y\). With some algebra, we can express this expectation as:

\[
\sum_{s,c} E_y[H_{U,\Sigma}(s, c)]
\]

\[
= \sum_{s,c} E_y \left[ -\frac{1}{2} y^*(s) U_{c}^{*} \gamma_{sc} \Sigma_{c}^{-1} U_{c} y(s) + y^*(s) U_{c}^{*} \Sigma_{c}^{-1} F_{sc} \right]
\]

\[
= \sum_{s,c} \frac{1}{2} \text{tr} \left( U_{c}^{*} \gamma_{sc} \Sigma_{c}^{-1} U_{c} E[y(s)y^*(s)] \right) + E \left[ y^*(s) U_{c}^{*} \Sigma_{c}^{-1} F_{sc} \right]
\]

\[
= \sum_{s,c} \text{tr} \left( \Sigma_{c}^{-1} \left( -\frac{1}{2} \gamma_{sc} U_{c} E[y(s)y^*(s)] U_{c}^{T} + F_{sc} E[y^*(s)] U_{c}^{*} \right) \right)
\]
We set the derivatives of $Q(U, \hat{U})$ respect to $U_c$ to 0 and we will obtain
\[
\sum_s \Sigma^{-1}_c \left( -U_c \gamma_{sc} E[y(s)y^*(s)] + \hat{F}_{sc} E[y^*(s)] \right) = 0 \tag{73}
\]
And hence, equation (23) follows obviously.

2). We now turn to the update the covariance matrix $\Sigma$, similarly, we can express its auxiliary function in the form of
\[
Q(\Sigma, \hat{\Sigma}) = \text{const} + \sum_{s,c,t} \gamma_{sc}(t) E_y \left[ \frac{1}{2} \log |\Sigma^{-1}_c| - \frac{1}{2} \left( \bar{x}_c(t) - \bar{\mu}_c \right)^* \Sigma^{-1}_c \left( \bar{x}_c(t) - \bar{\mu}_c \right) \right]
\]
where $\bar{\mu}_c = \bar{m}_c + \bar{U}_c y(s)$. Set the derivatives of $Q(\Sigma, \hat{\Sigma})$ respect to $\Sigma^{-1}_c$, we can get the equation as:
\[
\sum_{s,t} \gamma_{sc}(t) \Sigma^{-1}_c - \sum_{s,t} \gamma_{sc}(t) E_y \left[ (\bar{x}_c(t) - \bar{\mu}_c) (\bar{x}_c(t) - \bar{\mu}_c)^* \right] = 0
\]
Hence, the update formula for $\Sigma_c$ is:
\[
\Sigma_c = \frac{\sum_{s,t} \gamma_{sc}(t) E_y \left[ (\bar{x}_c(t) - \bar{\mu}_c) (\bar{x}_c(t) - \bar{\mu}_c)^* \right]}{\sum_{s,t} \gamma_{sc}(t)}
\]
\[
= \sum_s \left[ S_{sc} - \hat{F}_{sc} E[y(s)^*]U_c^* - U_c^* E[y(s)] \hat{F}_{sc}^* + \gamma_{sc} U_c E[y(s)y^*(s)] U_c^* \right] / \sum_s \gamma_{sc}
\]
And by equation (73), we can simplify the update formula for $\Sigma_c$ as equation (24) shows.

3). Finally, the auxiliary function of the model bias term $\bar{m}$ is
\[
Q(\bar{m}, \hat{\bar{m}})
\]
\[
= \text{const} + \sum_{s,c,t} \gamma_{sc}(t) E_y \left[ -\frac{1}{2} (x_c(t) - \bar{m}_c - U_c y)^* \Sigma^{-1}_c (x_c(t) - \bar{m}_c - U_c y) \right]
\]
\[
= \text{const} + \frac{1}{2} \sum_{s,c} \left[ F_{sc}^* \Sigma^{-1}_c \bar{m}_c + \bar{m}_c^* \Sigma^{-1}_c F_{sc} - \gamma_{sc} y^*(s)U_c^* \Sigma^{-1}_c \bar{m}_c \right.
\]
\[
- \gamma_{sc} \bar{m}_c^* \Sigma^{-1}_c U_c y(s) - \gamma_{sc} \bar{m}_c \Sigma^{-1}_c \bar{m}_c \]
Similarly, set the derivatives of $Q(\bar{m}, \hat{\bar{m}})$ respect to $\bar{m}_c$ to 0, we can get
\[
\sum_s \Sigma^{-1}_c \left( F_{sc} - \gamma_{sc} U_c y(s) - \gamma_{sc} \bar{m}_c \right) = 0
\]
And equation (25) follows. \qed
B Update of the weight projection

Proposition 5: Given the current estimate of weight projection matrix $\tilde{w}_c$ for each Gaussian component $c$, the update formula is given as

$$w_c = \tilde{w}_c - \left( \sum_j \max(\gamma_{jc}, \gamma_j w_{jc}) y(j) y^*(j) \right)^{-1} \sum_j (\gamma_{jc} - \gamma_j w_{jc}) y(j)$$  \hspace{1cm} (74)

Proof. To calculate the weight projection function, we can first express the auxiliary function for weight projection as

$$Q(w) = \sum_{jc} \gamma_{jc} \log w_{jc}$$

$$= \sum_{jc} \gamma_{jc} \left( w_{jc} y(j) - \log \sum_c \exp(w_{jc} y(j)) \right)$$  \hspace{1cm} (75)

With the same operation, namely, by the inequality $1 - x \leq -\log x$, we can rewritten the formula as:

$$Q(w) = \text{const} + \sum_{jc} \gamma_{jc} w_{jc} y(j) - \sum_j \gamma_j \frac{\exp(w_{jc} y(j))}{\sum_i \exp(w_i y(j))}$$

$$= \text{const} + \sum_{jc} \gamma_{jc} w_{jc} y(j) - \sum_j \gamma_j \frac{\exp(w_{jc} y(j))}{\sum_i \exp(w_i y(j))}$$  \hspace{1cm} (76)

where $\tilde{w}$ is the weight projection before update. Since we can not get the optimal solution directly, here we can use a quadratic polynomial function to approximate it. The first and second derivatives of $w_c$ is at the original point $w_c$ is as follows:

$$\frac{\partial Q(w)}{\partial w_c} |_{w_c} = \sum_j (\gamma_{jc} - \gamma_j w_{jc}) y(j)$$  \hspace{1cm} (77)

$$\frac{\partial^2 Q(w)}{\partial w_c^2} |_{w_c} = -\sum_j \gamma_j w_{jc} y(j) y^*(j)$$  \hspace{1cm} (78)

where $w_{jc} = \frac{\exp(w_{jc} y(j))}{\sum_i \exp(w_i y(j))}$ is the ‘old’ weight. The weight projection can be updated by

$$w_c = \tilde{w}_c + \left( \frac{\partial^2 Q(w)}{\partial w_c^2} |_{w_c} \right)^{-1} \left( \frac{\partial Q(w)}{\partial w_c} |_{w_c} \right)$$  \hspace{1cm} (79)

In practice, it’s found in [26] that $\frac{\partial^2 Q(w)}{\partial w_c^2} |_{w_c}$ is calculated by $-\sum_j \max(\gamma_{jc}, \gamma_j w_{jc}) y(j) y^*(j)$ seems make the update more stable (by ML estimate without subspace, the weight should be $w_{jc} = \frac{\gamma_{jc}}{\gamma_j}$, hence this approach make it closer to ML estimate). In addition, the auxiliary function (75) and (76) should be calculated for diagnostics on each iteration, and their values should both increase for a valid update. \hfill \Box