

# Joint Factor Analysis of Speaker and Session Variability: Theory and Algorithms

Patrick Kenny

**Abstract**— We give a full account of the algorithms needed to carry out a joint factor analysis of speaker and session variability in a training set in which each speaker is recorded over many different channels and we discuss the practical limitations that will be encountered if these algorithms are implemented on very large data sets.

## I. INTRODUCTION

This article is intended as a companion to [1] where we presented a new type of likelihood ratio statistic for speaker verification which is designed principally to deal with the problem of inter-session variability, that is the variability among recordings of a given speaker. This likelihood ratio statistic is based on a joint factor analysis of speaker and session variability in a training set in which each speaker is recorded over many different channels (such as one of the Switchboard II databases). Our purpose in the current article is to give detailed algorithms for carrying out such a factor analysis. Although we have only experimented with the applications of this model in speaker recognition we will also explain how it could serve as an integrated framework for progressive speaker-adaptation and on-line channel adaptation of HMM-based speech recognizers operating in situations where speaker identities are known.

## II. OVERVIEW OF THE JOINT FACTOR ANALYSIS MODEL

The joint factor analysis model can be viewed Gaussian distribution on speaker- and channel-dependent (or, more accurately, session-dependent) HMM supervectors in which most (but not all) of the variance in the supervector population is assumed to be accounted for by a small number of hidden variables which we refer to as speaker and channel factors. The speaker factors and the channel factors play different roles in that, for a given speaker, the values of the speaker factors are assumed to be the same for all recordings of the speaker but the channel factors are assumed to vary from one recording to another. For example, the Gaussian distribution on speaker-dependent supervectors used in eigenvoice MAP [2] is a special case of the factor analysis model in which there are no channel factors and all of the variance in the speaker-dependent HMM supervectors is assumed to be accounted

for by the speaker factors. In this case, for a given speaker, the values of the speaker factors are the co-ordinates of the speaker's supervector relative to a suitable basis of the eigenspace.

The general model combines the priors underlying classical MAP [3], eigenvoice MAP [2] and eigenchannel MAP [4] so we begin by reviewing these and showing how a single prior can be constructed which embraces all of them.

### A. Speaker and channel factors

We assume a fixed HMM structure containing a total of  $C$  mixture components. Let  $F$  be the dimension of the acoustic feature vectors so that, associated with each mixture component, there is an  $F$ -dimensional mean vector and an  $F \times F$  dimensional covariance matrix which in this article we will take to be diagonal. For each mixture component  $c = 1, \dots, C$ , let  $m_c$  denote the corresponding speaker-independent mean vector (which is usually estimated by Baum-Welch training) and let  $\mathbf{m}$  denote the  $CF \times 1$  supervector obtained by concatenating  $m_1, \dots, m_C$ .

To begin with let us ignore channel effects and assume that each speaker  $s$  can be modeled by a single speaker-dependent supervector  $\mathbf{M}(s)$  (common to all recordings of the speaker). For each mixture component  $c$ , let  $M_c(s)$  be the corresponding subvector of  $\mathbf{M}(s)$ .

Classical MAP is often presented as a rule of thumb ('interpolate between the speaker-dependent and speaker-independent estimates of the HMM mean vectors') without explaining what this rule has to do with prior and posterior distributions. In order to achieve the type of generalization that we are aiming for we will have to spell out the role of these distributions explicitly. The assumption concerning the prior on speaker supervectors is that all of the HMM mean vectors  $M_c(s)$  are statistically independent (where  $c$  ranges over all mixture components and  $s$  over all speakers) and, for each mixture component  $c$ , the marginal distribution of  $M_c(s)$  is normal and independent of  $s$ . If we further assume that the prior distribution is normal then the basic assumption is that there is a diagonal matrix  $\mathbf{d}$  such that, for a randomly chosen speaker  $s$ ,

$$\mathbf{M}(s) = \mathbf{m} + \mathbf{d}\mathbf{z}(s) \quad (1)$$

where  $\mathbf{z}(s)$  is a hidden vector distributed according to the standard normal distribution,  $N(\mathbf{z}|\mathbf{0}, \mathbf{I})$ . (Although we will only consider the case where  $\mathbf{d}$  is diagonal, a generalization to the case where  $\mathbf{d}$  is a block diagonal matrix with each block being of dimension  $F \times F$  is possible.) Given some adaptation data for a speaker  $s$ , MAP adaptation consists in calculating

The authors are with the Centre de recherche informatique de Montréal (CRIM).

P. Kenny can be reached at (514) 840 1235 x 4624, email pkenny@crim.ca

This work was supported in part by the Natural Science and Engineering Research Council of Canada and by the Ministère du Développement Économique et Régional et de la Recherche du gouvernement du Québec

the posterior distribution of  $M(s)$ ; the MAP estimate of  $M(s)$  is just the mode of this posterior. We will explain later how this give rise to the rule of thumb.

Provided that  $\mathbf{d}$  is non-singular, classical MAP adaptation is guaranteed to be asymptotically equivalent to speaker-dependent training as the amount of adaptation data increases. However, in the absence of observations for a given speaker and mixture component, the classical MAP estimator falls back to the speaker-independent estimate of the HMM mean vector. Thus if the number of mixture components  $C$  is large, classical MAP tends to saturate slowly in the sense that large amounts of enrollment data are needed to use it to full advantage.

Eigenvoice MAP assumes instead that there is a rectangular matrix  $\mathbf{v}$  of dimensions  $CF \times R$  where  $R \ll CF$  such that, for a randomly chosen speaker  $s$ ,

$$M(s) = \mathbf{m} + \mathbf{v}\mathbf{y}(s) \quad (2)$$

where  $\mathbf{y}(s)$  is a hidden  $R \times 1$  vector having a standard normal distribution. Since the dimension of  $\mathbf{y}(s)$  is much smaller than that of  $\mathbf{z}(s)$ , eigenvoice MAP tends to saturate much more quickly than classical MAP. But this approach to speaker adaptation suffers from the drawback that, in estimating  $\mathbf{v}$  from a given training set, it is necessary to assume that  $R$  is less than or equal to the number of training speakers [2] so that a very large number of training speakers may be needed to estimate  $\mathbf{v}$  properly. Thus in practice there is no guarantee that eigenvoice MAP adaptation will exhibit correct asymptotic behavior as the quantity of enrollment data for a speaker increases.

The strengths and weaknesses of classical MAP and eigenvoice MAP complement each other. (Eigenvoice MAP is preferable if small amounts of data are available for speaker adaptation and classical MAP if large amounts are available.) An obvious strategy to combine the two is to assume a decomposition of the form

$$M(s) = \mathbf{m} + \mathbf{v}\mathbf{y}(s) + \mathbf{d}\mathbf{z}(s) \quad (3)$$

where the hidden vector

$$\begin{pmatrix} \mathbf{y}(s) \\ \mathbf{z}(s) \end{pmatrix}$$

has a standard normal distribution. Given some adaptation data for a speaker  $s$ , speaker adaptation can be implemented by calculating the posterior distribution of  $M(s)$  (as in classical MAP or eigenvoice MAP). In this situation it is no longer appropriate to speak of eigenvoices; rather  $\mathbf{v}$  is a ‘factor loading matrix’ and the components of  $\mathbf{y}(s)$  are ‘speaker factors’. If  $\mathbf{d} = \mathbf{0}$  then all speaker supervectors are contained in the affine space defined by translating the range of  $\mathbf{v}\mathbf{v}^*$  by  $\mathbf{m}$  so we will call this the *speaker space*. We will use this terminology in the general case ( $\mathbf{d} \neq \mathbf{0}$ ) as well because, although it may be unrealistic to assume that all speaker supervectors are contained in a linear manifold of low dimension, the intuition underlying eigenvoice modeling seems to be quite sound and our experience has been that  $\mathbf{d}$  is relatively small in practice.

In [4] we presented a quick and dirty solution to the problem of channel adaptation of speaker HMM’s which we called eigenchannel MAP. Suppose that for a speaker  $s$  we

have obtained a speaker adapted HMM or, equivalently, a point estimate  $\mathbf{m}(s)$  of  $M(s)$  by some speaker adaptation technique such as eigenvoice MAP or classical MAP. Given a collection of recordings for the speaker, let  $M_h(s)$  denote the supervector corresponding to the recording  $h$  where  $h = 1, 2, \dots$ . Our starting point in [4] was to assume that there is a matrix  $\mathbf{u}$  of low rank such that for each recording  $h$ , the prior distribution of  $M_h(s)$  is given by

$$M_h(s) = \mathbf{m}(s) + \mathbf{u}\mathbf{x}_h(s) \quad (4)$$

where  $\mathbf{x}_h(s)$  is a hidden vector having a standard normal distribution. Given some adaptation data from the recording, the speaker supervector  $\mathbf{m}(s)$  can be adapted to a new set of recording conditions by calculating the posterior distribution of  $M_h(s)$  just as in eigenvoice MAP. In fact the form of this model is so similar to (2) that no new mathematics is needed to develop it. Furthermore, as we explained in [4], it is less susceptible to the rank deficiency problem that afflicts eigenvoice MAP in practice. The trouble with this model is that it neglects to take channel effects into account in deriving the point estimate  $\mathbf{m}(s)$  (a problem which is not easily remedied). The assumption that  $M(s)$  can be replaced by a point estimate is also unsatisfactory since eigenvoice MAP and (especially) classical MAP produce posterior distributions on  $M(s)$  rather than point estimates. (It is only in the case where the amount of adaptation data is large that the posterior distributions become concentrated on points.)

In order to develop an integrated approach to speaker- and channel-adaptation we will work out the model obtained by substituting the right hand side of (3) for  $\mathbf{m}(s)$  in (4). Thus we assume that, for a given speaker  $s$  and recording  $h$ ,

$$\left. \begin{aligned} M(s) &= \mathbf{m} + \mathbf{v}\mathbf{y}(s) + \mathbf{d}\mathbf{z}(s) \\ M_h(s) &= M(s) + \mathbf{u}\mathbf{x}_h(s). \end{aligned} \right\} \quad (5)$$

The range of  $\mathbf{u}\mathbf{u}^*$  can be thought of as the *channel space*. Alternatively, since the channel factors may be capturing intra-speaker variability as much as channel variability, it could be termed the intra-speaker space. (This would be in keeping with the terminology used in the dual eigenspace approach to face recognition [5].)

So our factor analysis model will be specified by a quintuple of hyperparameters  $\mathbf{\Lambda}$  of the form  $(\mathbf{m}, \mathbf{u}, \mathbf{v}, \mathbf{d}, \mathbf{\Sigma})$  where

- 1)  $\mathbf{m}$  is a supervector (of dimension  $CF \times 1$ ).
- 2)  $\mathbf{u}$  is a matrix of dimension  $CF \times R_C$  ( $R_C$  for channel rank).
- 3)  $\mathbf{v}$  is a matrix of dimension  $CF \times R_S$  ( $R_S$  for speaker rank).
- 4)  $\mathbf{d}$  is a  $CF \times CF$  diagonal matrix.
- 5)  $\mathbf{\Sigma}$  is a  $CF \times CF$  diagonal covariance matrix whose diagonal blocks we denote by  $\Sigma_c$  for  $c = 1, \dots, C$ .

To explain the role of the covariance matrices in 5), fix a mixture component  $c$ . For each speaker  $s$  and recording  $h$ , let  $M_{hc}(s)$  denote the subvector of  $M_h(s)$  corresponding to the given mixture component. We assume that, for all speakers  $s$  and recordings  $h$ , observations drawn from the mixture component  $c$  are distributed with mean  $M_{hc}(s)$  and covariance matrix  $\Sigma_c$ . In the case where  $\mathbf{u} = \mathbf{0}$  and  $\mathbf{v} = \mathbf{0}$ , so that the factor analysis model reduces to the prior for classical MAP,

the ‘relevance factors’ of [6] are the diagonal entries of  $\mathbf{d}^{-2}\Sigma$  (as we will explain).

The term  $\mathbf{d}\mathbf{z}(s)$  is included in the factor analysis model in order to ensure that it inherits the asymptotic behavior of classical MAP but it is costly in terms of both mathematical and computational complexity. The reason for this is that, although the increase in the number of free parameters is relatively modest since (unlike  $\mathbf{u}$  and  $\mathbf{v}$ )  $\mathbf{d}$  is assumed to be diagonal, the increase in the number of hidden variables is enormous (assuming that there are at most a few hundred speaker and channel factors). On the other hand if  $\mathbf{d} = \mathbf{0}$ , the model is quite simple since the basic assumption is that each speaker- and channel-dependent supervector is a sum of two supervectors one of which is contained in the speaker space and the other in the channel space.<sup>1</sup> We will use the term Principal Components Analysis (PCA) to refer to the case  $\mathbf{d} = \mathbf{0}$  and reserve the term Factor Analysis to describe the general case.

### B. The likelihood function

To describe the likelihood function for the factor analysis model, suppose that we are given a set of recordings for a speaker  $s$  indexed by  $h = 1, \dots, H(s)$ . For each recording  $h$ , assume that each frame has been aligned with a mixture component and let  $\mathcal{X}_h(s)$  denote the collection of labeled frames for the recording. Set

$$\underline{\mathcal{X}}(s) = \begin{pmatrix} \mathcal{X}_1(s) \\ \vdots \\ \mathcal{X}_{H(s)}(s) \end{pmatrix}$$

and let  $\underline{\mathbf{X}}(s)$  be the vector of hidden variables defined by

$$\underline{\mathbf{X}}(s) = \begin{pmatrix} \mathbf{x}_1(s) \\ \vdots \\ \mathbf{x}_{H(s)}(s) \\ \mathbf{y}(s) \\ \mathbf{z}(s) \end{pmatrix}.$$

If  $\underline{\mathbf{X}}(s)$  were given we could write down  $\mathbf{M}_h(s)$  and calculate the (Gaussian) likelihood of  $\mathcal{X}_h(s)$  for each recording  $h$  so the calculation of the likelihood of  $\underline{\mathcal{X}}(s)$  would be straightforward. Let us denote this conditional likelihood by  $P_{\Lambda}(\underline{\mathcal{X}}(s)|\underline{\mathbf{X}}(s))$ . Since the values of the hidden variables are not given, calculating the likelihood of  $\underline{\mathcal{X}}(s)$  requires evaluating the integral

$$\int P_{\Lambda}(\underline{\mathcal{X}}(s)|\underline{\mathbf{X}})N(\underline{\mathbf{X}}|\underline{\mathbf{0}}, \underline{\mathbf{I}})d\underline{\mathbf{X}} \quad (6)$$

where  $N(\underline{\mathbf{X}}|\underline{\mathbf{0}}, \underline{\mathbf{I}})$  is the standard Gaussian kernel

$$N(\mathbf{x}_1|\mathbf{0}, \mathbf{I}) \cdots N(\mathbf{x}_{H(s)}|\mathbf{0}, \mathbf{I})N(\mathbf{y}|\mathbf{0}, \mathbf{I})N(\mathbf{z}|\mathbf{0}, \mathbf{I}).$$

We denote the value of this integral by  $P_{\Lambda}(\underline{\mathcal{X}}(s))$ ; a closed form expression for  $P_{\Lambda}(\underline{\mathcal{X}}(s))$  is given in Theorem 3 below.

<sup>1</sup>In fact this decomposition is normally unique since the range of  $\mathbf{u}\mathbf{u}^*$  and the range of  $\mathbf{v}\mathbf{v}^*$ , being low dimensional subspaces of a very high dimensional space, will typically only intersect at the origin.

### C. Estimating the hyperparameters

The principal problem we have to deal with is how to estimate the hyperparameters which specify the factor analysis model given that speaker- and channel-dependent HMM supervectors are unobservable. (It is not possible in practice to estimate a speaker- and channel-dependent HMM by maximum likelihood methods from a single recording.) We will show how, if we are given a training set in which each speaker is recorded in multiple sessions, we can estimate the hyperparameters  $\Lambda$  by EM algorithms which guarantee that the total likelihood of the training data increases from one iteration to the next. (The total likelihood of the training data is  $\prod_s P_{\Lambda}(\underline{\mathcal{X}}(s))$  where  $s$  ranges over the training speakers.) We refer to these as speaker-independent hyperparameter estimation algorithms (or simply as *training* procedures) since they consist in fitting the factor analysis model (5) to the entire collection of speakers in the training data rather than to an individual speaker.

One estimation algorithm, which we will refer to simply as maximum likelihood estimation, can be derived by extending Proposition 3 in [2] to handle the hyperparameters  $\mathbf{u}$  and  $\mathbf{d}$  in addition to  $\mathbf{v}$  and  $\Sigma$ . Our experience with it has been that it tends to converge very slowly. Another algorithm can be derived by using the divergence minimization approach to hyperparameter estimation introduced in [7]. This seems to converge much more rapidly but it has the property that it keeps the orientation of the speaker and channel spaces fixed so that it can only be used if these are well initialized. (A similar situation arises when the divergence minimization approach is used to estimate inter-speaker correlations. See the remark following Proposition 3 in [7].) We have experimented with the maximum likelihood approach on its own and with the maximum likelihood approach followed by divergence minimization. With one notable exception we obtained essentially the same performance in both cases (even though the hyperparameter estimates are quite different.)

We also need a speaker-dependent estimation algorithm for the hyperparameters both for constructing likelihood ratio statistics for speaker verification [8], [1] and for progressive speaker adaptation. For this we assume that, for a given speaker  $s$  and recording  $h$ ,

$$\left. \begin{aligned} \mathbf{M}(s) &= \mathbf{m}(s) + \mathbf{v}(s)\mathbf{y}(s) + \mathbf{d}(s)\mathbf{z}(s) \\ \mathbf{M}_h(s) &= \mathbf{M}(s) + \mathbf{u}\mathbf{x}_h(s). \end{aligned} \right\} \quad (7)$$

That is, we make the hyperparameters  $\mathbf{m}$ ,  $\mathbf{v}$  and  $\mathbf{d}$  speaker-dependent but we continue to treat the hyperparameters  $\Sigma$  and  $\mathbf{u}$  as speaker-independent. (There is no reason to suppose that channel effects vary from one speaker to another.) Given some recordings of the speaker  $s$ , we estimate the speaker-dependent hyperparameters  $\mathbf{m}(s)$ ,  $\mathbf{v}(s)$  and  $\mathbf{d}(s)$  by first using the speaker-independent hyperparameters and the recordings to calculate the posterior distribution of  $\mathbf{M}(s)$  and then adjusting the speaker-dependent hyperparameters to fit this posterior. More specifically, we find the distribution of the form  $\mathbf{m}(s) + \mathbf{v}(s)\mathbf{y}(s) + \mathbf{d}(s)\mathbf{z}(s)$  which is closest to the posterior in the sense that the divergence is minimized. This is just the minimum divergence estimation algorithm applied to a single speaker in the case where  $\mathbf{u}$  and  $\Sigma$  are held fixed.

Thus  $\mathbf{m}(s)$  is an estimate of the speaker's supervector when channel effects are abstracted and  $\mathbf{d}(s)$  and  $\mathbf{v}(s)$  measure the uncertainty in this estimate.

#### D. Speaker- and channel-adaptation of HMM's

Given (speaker-independent or speaker-dependent) estimates of the hyperparameters  $\mathbf{m}, \mathbf{u}, \mathbf{v}, \mathbf{d}$  and  $\Sigma$  and a collection of recordings for a speaker  $s$ , we will explain how to calculate the posterior distribution of the hidden variables  $\mathbf{y}(s), \mathbf{z}(s)$  and  $\mathbf{x}_h(s)$  for each recording  $h$ . As in [2], this is by far the most important calculation that needs to be performed in order to implement the model. It turns out to be much more difficult and computationally expensive in the general case than in the PCA case.

Knowing this posterior, it is easy to calculate the posterior distribution of the speaker- and channel-dependent supervector  $\mathbf{M}_h(s)$  for each recording  $h$  and hence to implement MAP speaker- and channel-adaptation for each recording. Note however that, since the values of  $\mathbf{z}(s)$  and  $\mathbf{y}(s)$  are assumed to be common to all of the recordings, the posterior distributions of  $\mathbf{M}_h(s)$  for  $h = 1, 2, \dots$  cannot be calculated independently of each other. Thus a naive approach to the problem of performing speaker- and channel-adaptation to a new set of recording conditions would require processing all of the recordings of the speaker if the calculation is to be carried out exactly. Clearly, in practical situations where multiple recordings of a speaker are available (as in dictation) it would be much more satisfactory to process these recordings sequentially (producing a speaker- and channel-adapted HMM for each recording as it becomes available) rather than in batch mode.

The most natural approach to avoiding the need for batch processing is to apply the speaker-dependent hyperparameter estimation algorithm sequentially. Whenever a new recording of the speaker becomes available we can use the current estimates of the speaker-dependent hyperparameters  $\mathbf{m}(s), \mathbf{v}(s)$  and  $\mathbf{d}(s)$  and the given recording to perform MAP HMM adaptation. We can also use the given recording to update the speaker-dependent hyperparameters.

This enables us to perform progressive speaker-adaptation and on-line channel-adaptation one recording at a time. Note that the speaker-independent hyperparameter estimates play a fundamental role if speaker-dependent priors are allowed to evolve in this way. The speaker-independent hyperparameters needed to initialize the sequential update algorithm for the speaker-dependent hyperparameters, they provide the loading matrix for the channel factors and (since minimum divergence estimation preserves the orientation of the speaker space) they impose constraints how the speaker-dependent priors can evolve.

### III. LIKELIHOOD CALCULATIONS

Suppose we are given a speaker  $s$  and a collection of recordings  $h = 1, \dots, H(s)$  with observable variables  $\underline{\mathbf{x}}(s)$  and hidden variables  $\underline{\mathbf{X}}(s)$ . For a given set of hyperparameters  $\Lambda$  we denote the joint distribution of  $\underline{\mathbf{X}}(s)$  and  $\underline{\mathbf{x}}(s)$  by  $P_\Lambda(\underline{\mathbf{X}}(s), \underline{\mathbf{x}}(s))$ . In this section we will study this joint

distribution and show how to derive the posterior and marginal distributions needed to implement the factor analysis model. More specifically we will calculate the marginal distribution of the observable variables  $P_\Lambda(\underline{\mathbf{x}}(s))$  and the posterior distribution of the hidden variables,  $P_\Lambda(\underline{\mathbf{X}}(s)|\underline{\mathbf{x}}(s))$ . We will also show how this posterior distribution can be used for speaker- and channel- adaptation of HMM's.

#### A. The joint distribution of the observable and hidden variables

Note first that if we fix a recording  $h$ , it is a straightforward matter to calculate the conditional likelihood of the observations  $\mathcal{X}_h(s)$  if the supervector  $\mathbf{M}_h(s)$  is given. So we begin by calculating the conditional distribution of the observable variables given the hidden variables,  $P_\Lambda(\underline{\mathbf{x}}(s)|\underline{\mathbf{X}}(s))$ . Multiplying this by the standard Gaussian kernel  $N(\underline{\mathbf{X}}(s)|\underline{\mathbf{0}}, \underline{\mathbf{I}})$  gives the joint distribution of the observable variables and the hidden variables.

First we define some statistics. For each recording  $h$  and mixture component  $c$ , let  $N_{hc}(s)$  be the total number of observations for the given mixture component and set

$$F_{hc}(s, m_c) = \sum_t (X_t - m_c)$$

$$S_{hc}(s, m_c) = \text{diag} \left( \sum_t (X_t - m_c)(X_t - m_c)^* \right)$$

where the sum extends over all observations  $X_t$  aligned with the given mixture component,  $m_c$  is the  $c$ th block of  $\mathbf{m}$ , and  $\text{diag}()$  sets off-diagonal entries to 0. These statistics can be extracted from the training data in various ways; the simplest algorithm conceptually is to extract them by means of a Viterbi alignment with a speaker- and channel-independent HMM but a forward-backward alignment could also be used. (Also, the speaker- and channel-adapted HMM's described in Section III E below could be used in place of the speaker- and channel-independent HMM.)

Let  $\mathbf{N}_h(s)$  be the  $CF \times CF$  diagonal matrix whose diagonal blocks are  $N_{hc}(s)I$  (for  $c = 1, \dots, C$ ) where  $I$  is the  $F \times F$  identity matrix. Let  $\mathbf{F}_h(s, \mathbf{m})$  be the  $CF \times 1$  vector obtained by concatenating  $F_{hc}(s, m_c)$  (for  $c = 1, \dots, C$ ). Similarly, let  $\mathbf{S}_h(s, \mathbf{m})$  be the  $CF \times CF$  diagonal matrix whose diagonal blocks are  $S_{hc}(s, m_c)$  (for  $c = 1, \dots, C$ ).

Let  $\underline{\mathbf{N}}(s)$  be the  $H(s)CF \times H(s)CF$  matrix whose diagonal blocks are  $\mathbf{N}_h(s)$  for  $h = 1, \dots, H(s)$  where  $H(s)$  is the number of recordings of the speaker  $s$ . Let  $\underline{\mathbf{F}}(s, \mathbf{m})$  be the  $H(s)CF \times 1$  vector obtained by concatenating  $\mathbf{F}_h(s, \mathbf{m})$ . Let  $\underline{\mathbf{S}}(s)$  be the  $H(s)CF \times H(s)CF$  matrix whose diagonal entries are all equal to  $\Sigma$ . Let  $\underline{\mathbf{V}}(s)$  be the matrix of dimension  $H(s)CF \times (H(s)R_C + R_S + CF)$  defined by

$$\underline{\mathbf{V}}(s) = \begin{pmatrix} \mathbf{u} & & \mathbf{v} & \mathbf{d} \\ & \ddots & \vdots & \vdots \\ & & \mathbf{u} & \mathbf{v} & \mathbf{d} \end{pmatrix} \quad (8)$$

We are now in a position to write down a formula for the conditional distribution of the observable variables given the hidden variables.

*Theorem 1: For each speaker  $s$ ,*

$$\log P_{\Lambda}(\underline{\mathcal{X}}(s)|\underline{\mathbf{X}}(s)) = G_{\Sigma}(s, \mathbf{m}) + H_{\Lambda}(s, \underline{\mathbf{X}}(s))$$

where

$$G_{\Sigma}(s, \mathbf{m}) = \sum_{h=1}^{H(s)} \left( \sum_{c=1}^C N_{hc}(s) \log \frac{1}{(2\pi)^{F/2} |\Sigma_c|^{1/2}} - \frac{1}{2} \text{tr}(\Sigma^{-1} \mathbf{S}_h(s, \mathbf{m})) \right)$$

and

$$H_{\Lambda}(s, \underline{\mathbf{X}}) = \underline{\mathbf{X}}^* \mathbf{V}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m}) - \frac{1}{2} \underline{\mathbf{X}}^* \mathbf{V}^*(s) \underline{\mathbf{N}}(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{V}}(s) \underline{\mathbf{X}}.$$

*Proof:* Set

$$\underline{\mathbf{O}}(s) = \underline{\mathbf{V}}(s) \underline{\mathbf{X}}(s)$$

so that for each recording  $h$ ,

$$\mathbf{M}_h(s) = \mathbf{m} + \mathbf{O}_h(s).$$

For each  $h = 1, \dots, H(s)$ ,

$$\begin{aligned} \log P_{\Lambda}(\mathcal{X}_h(s)|\underline{\mathbf{X}}(s)) &= \sum_{c=1}^C N_{hc}(s) \log \frac{1}{(2\pi)^{F/2} |\Sigma_c|^{1/2}} \\ &\quad - \frac{1}{2} \sum_{c=1}^C \sum_t (X_t - M_{hc}(s))^* \Sigma_c^{-1} (X_t - M_{hc}(s)). \end{aligned}$$

where, for each mixture component  $c$ ,  $t$  extends over all frames that are aligned with the mixture component. Ignoring the factor of  $-\frac{1}{2}$  for the time being, the second term here can be expressed in terms of  $\mathbf{O}_h(s)$  as follows:

$$\begin{aligned} &\sum_{c=1}^C \sum_t (X_t - m_c - O_{hc}(s))^* \Sigma_c^{-1} (X_t - m_c - O_{hc}(s)) \\ &= \sum_{c=1}^C \sum_t (X_t - m_c)^* \Sigma_c^{-1} (X_t - m_c) \\ &\quad - 2 \sum_{c=1}^C \sum_t O_{hc}^*(s) \Sigma_c^{-1} (X_t - m_c) \\ &\quad + \sum_{c=1}^C O_{hc}^*(s) N_{hc}(s) \Sigma_c^{-1} O_{hc}(s) \\ &= \sum_{c=1}^C \text{tr}(\Sigma_c^{-1} \mathbf{S}_{hc}(s, m_c)) \\ &\quad - 2 \sum_{c=1}^C O_{hc}^*(s) \Sigma_c^{-1} \mathbf{F}_{hc}(s, m_c) \\ &\quad + \sum_{c=1}^C O_{hc}^*(s) N_{hc}(s) \Sigma_c^{-1} O_{hc}(s) \\ &= \text{tr}(\Sigma^{-1} \mathbf{S}_h(s, \mathbf{m})) \\ &\quad - 2 \mathbf{O}_h^*(s) \Sigma^{-1} \mathbf{F}_h(s, \mathbf{m}) \\ &\quad + \mathbf{O}_h^*(s) \mathbf{N}_h(s) \Sigma^{-1} \mathbf{O}_h(s) \end{aligned}$$

so that

$$\begin{aligned} \log P_{\Lambda}(\mathcal{X}_h(s)|\underline{\mathbf{X}}(s)) &= \sum_{c=1}^C N_{hc}(s) \log \frac{1}{(2\pi)^{F/2} |\Sigma_c|^{1/2}} \\ &\quad - \frac{1}{2} \text{tr}(\Sigma^{-1} \mathbf{S}_h(s, \mathbf{m})) \\ &\quad + \mathbf{O}_h^*(s) \Sigma^{-1} \mathbf{F}_h(s, \mathbf{m}) \\ &\quad - \frac{1}{2} \mathbf{O}_h^*(s) \mathbf{N}_h(s) \Sigma^{-1} \mathbf{O}_h(s). \end{aligned}$$

Hence

$$\begin{aligned} \log P_{\Lambda}(\underline{\mathcal{X}}(s)|\underline{\mathbf{X}}(s)) &= \sum_{h=1}^{H(s)} \log P_{\Lambda}(\mathcal{X}_h(s)|\underline{\mathbf{X}}(s)) \\ &= \sum_{h=1}^{H(s)} \left( \sum_{c=1}^C N_{hc}(s) \log \frac{1}{(2\pi)^{F/2} |\Sigma_c|^{1/2}} - \frac{1}{2} \text{tr}(\Sigma^{-1} \mathbf{S}_h(s, \mathbf{m})) \right) + \mathbf{O}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m}) \\ &\quad - \frac{1}{2} \mathbf{O}^*(s) \underline{\mathbf{N}}(s) \underline{\Sigma}^{-1}(s) \mathbf{O}(s) \\ &= G_{\Sigma}(s, \mathbf{m}) + H_{\Lambda}(s, \underline{\mathbf{X}}(s)) \end{aligned}$$

as required.  $\blacksquare$

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

The posterior distribution of the hidden variables given the observable variables,  $P_{\Lambda}(\underline{\mathbf{X}}(s)|\underline{\mathcal{X}}(s))$ , can be calculated just as in Proposition 1 of [2]. Let

$$\underline{\mathbf{L}}(s) = \underline{\mathbf{I}} + \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{N}}(s) \underline{\mathbf{V}}(s).$$

*Theorem 2: For each speaker  $s$ , the posterior distribution of  $\underline{\mathbf{X}}(s)$  given  $\underline{\mathcal{X}}(s)$  is Gaussian with mean*

$$\underline{\mathbf{L}}^{-1}(s) \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m})$$

and covariance matrix  $\underline{\mathbf{L}}^{-1}(s)$ .

*Proof:* By Theorem 1,

$$\begin{aligned} \log P_{\Lambda}(\underline{\mathcal{X}}(s)|\underline{\mathbf{X}}) &= G_{\Sigma}(s, \mathbf{m}) + \underline{\mathbf{X}}^* \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m}) \\ &\quad - \frac{1}{2} \underline{\mathbf{X}}^* \underline{\mathbf{V}}^*(s) \underline{\mathbf{N}}(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{V}}(s) \underline{\mathbf{X}}. \end{aligned}$$

So

$$\begin{aligned} P_{\Lambda}(\underline{\mathbf{X}}|\underline{\mathcal{X}}(s)) &\propto P_{\Lambda}(\underline{\mathcal{X}}(s)|\underline{\mathbf{X}}) N(\underline{\mathbf{X}}|\underline{\mathbf{0}}, \underline{\mathbf{I}}) \\ &= \exp \left( \underline{\mathbf{X}}^* \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m}) - \frac{1}{2} \underline{\mathbf{X}}^* \underline{\mathbf{L}}(s) \underline{\mathbf{X}} \right) \\ &\propto \exp \left( -\frac{1}{2} (\underline{\mathbf{A}} - \underline{\mathbf{X}})^* \underline{\mathbf{L}}(s) (\underline{\mathbf{A}} - \underline{\mathbf{X}}) \right) \end{aligned}$$

where

$$\underline{\mathbf{A}} = \underline{\mathbf{L}}(s)^{-1} \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m})$$

as required.  $\blacksquare$

We use the notations  $E[\cdot]$  and  $\text{Cov}(\cdot, \cdot)$  to indicate expectations and covariances calculated with the posterior distribution specified in Theorem 2. (Strictly speaking the notation should include a reference to the speaker  $s$  but this will always be clear from the context.) All of the computations needed to implement the factor analysis model can be cast in terms of these posterior expectations and covariances if one bears in mind that correlations can be expressed in terms of covariances by relations such as

$$E[z(s)y^*(s)] = E[z(s)]E[y^*(s)] + \text{Cov}(z(s), y(s))$$

etc. The next two sections which give a detailed account of how to calculate these posterior expectations and covariances can be skipped on a first reading.

Although the hidden variables in the factor analysis model are all assumed to be independent in the prior (5), this is not true in the posterior (the matrix  $\underline{L}(s)$  is sparse but  $\underline{L}^{-1}(s)$  is not). It turns out however that the only entries in  $\underline{L}^{-1}(s)$  whose values are actually needed to implement the model are those which correspond to non-zero entries in  $\underline{L}(s)$ . In the PCA case it is straightforward to calculate only the portion of  $\underline{L}^{-1}(s)$  that is needed but in the general case calculating more entries of  $\underline{L}^{-1}(s)$  seems to be unavoidable. The calculation here involves inverting a large matrix and this has important practical consequences if there is a large number of recordings for the speaker: the computation needed to evaluate the posterior is  $O(H(s))$  in the PCA case but  $O(H^3(s))$  in the general case. We will return to this question in Section VII.

### C. Evaluating the posterior in the general case

Let

$$\mathbf{X}_h(s) = \begin{pmatrix} \mathbf{x}_h(s) \\ \mathbf{y}(s) \end{pmatrix}$$

for  $h = 1, \dots, H(s)$ . Dropping the reference to  $s$  for convenience, the posterior expectations and covariances we will need to evaluate are  $E[z]$ ,  $\text{diag}(\text{Cov}(z, z))$  and, for  $h = 1, \dots, H$ ,  $E[\mathbf{X}_h]$ ,  $\text{Cov}(\mathbf{X}_h, \mathbf{X}_h)$  and  $\text{Cov}(\mathbf{X}_h, z)$ . (Note that  $E[\underline{\mathbf{X}}]$  can be written down if  $E[\mathbf{X}_h]$  is given for  $h = 1, \dots, H$ .) We will also need to calculate the determinant of  $\underline{L}$  in order to evaluate the formula for the complete likelihood function given in Theorem 3 below.

We will use the following identities which hold for any symmetric positive definite matrix:

$$\begin{pmatrix} \alpha & \beta \\ \beta^* & \gamma \end{pmatrix}^{-1} = \begin{pmatrix} \zeta^{-1} & -\zeta^{-1}\beta\gamma^{-1} \\ -\gamma^{-1}\beta^*\zeta^{-1} & \gamma^{-1} + \gamma^{-1}\beta^*\zeta^{-1}\beta\gamma^{-1} \end{pmatrix}$$

and

$$\begin{vmatrix} \alpha & \beta \\ \beta^* & \gamma \end{vmatrix} = |\zeta||\gamma|$$

where

$$\zeta = \alpha - \beta\gamma^{-1}\beta^*.$$

By Theorem 2  $\underline{L} = \underline{I} + \underline{V}^*\underline{\Sigma}^{-1}\underline{N}\underline{V}$ , where  $\underline{V}$  is given by (8). A straightforward calculation shows that  $\underline{L}$  can be written as

$$\begin{pmatrix} a_1 & & b_1 & c_1 \\ & \ddots & \vdots & \vdots \\ & & a_H & b_H \\ b_1^* & \cdots & b_H^* & I + v^*\Sigma^{-1}Nv \\ c_1^* & \cdots & c_H^* & d\Sigma^{-1}Nv & v^*\Sigma^{-1}Nd \\ & & & I + \Sigma^{-1}Nd^2 \end{pmatrix} \quad (9)$$

where

$$N = N_1 + \cdots + N_H$$

and

$$\begin{aligned} a_h &= I + u^*\Sigma^{-1}N_h u \\ b_h &= u^*\Sigma^{-1}N_h v \\ c_h &= u^*\Sigma^{-1}N_h d \end{aligned}$$

for  $h = 1, \dots, H$ . So taking

$$\begin{aligned} \alpha &= \begin{pmatrix} a_1 & & b_1 \\ & \ddots & \vdots \\ & & a_H & b_H \\ b_1^* & \cdots & b_H^* & I + v^*\Sigma^{-1}Nv \end{pmatrix} \\ \beta &= \begin{pmatrix} c_1 \\ \vdots \\ c_H \\ v^*\Sigma^{-1}Nd \end{pmatrix} \\ \gamma &= I + \Sigma^{-1}Nd^2 \\ \zeta &= \alpha - \beta\gamma^{-1}\beta^* \end{aligned}$$

we have

$$\begin{aligned} \underline{L}^{-1} &= \begin{pmatrix} \zeta^{-1} & -\zeta^{-1}\beta\gamma^{-1} \\ -\gamma^{-1}\beta^*\zeta^{-1} & \gamma^{-1} + \gamma^{-1}\beta^*\zeta^{-1}\beta\gamma^{-1} \end{pmatrix} \\ |\underline{L}| &= |\zeta|^{-1}|\gamma|. \end{aligned}$$

Since  $\text{Cov}(\underline{\mathbf{X}}, \underline{\mathbf{X}}) = \underline{L}^{-1}$ , this enables us to write down the posterior covariances that we need.

In order to calculate the posterior expectations, set  $\mathbf{F}_h = \mathbf{F}_h(s, \mathbf{m})$  for  $h = 1, \dots, H$  and let  $\mathbf{F} = \mathbf{F}_1 + \cdots + \mathbf{F}_H$ . Recall that by Theorem 2,

$$E[\underline{\mathbf{X}}] = \underline{L}^{-1}\underline{V}^*\underline{\Sigma}^{-1}\underline{\mathbf{F}}$$

and by (8)

$$\underline{V}^*\underline{\Sigma}^{-1}\underline{\mathbf{F}} = \begin{pmatrix} u^*\Sigma^{-1}\mathbf{F}_1 \\ \vdots \\ u^*\Sigma^{-1}\mathbf{F}_H \\ \frac{v^*\Sigma^{-1}\mathbf{F}}{d\Sigma^{-1}\mathbf{F}} \end{pmatrix}.$$

Hence

$$E[\underline{\mathbf{X}}] = \begin{pmatrix} \zeta^{-1} & -\zeta^{-1}\beta\gamma^{-1} \\ -\gamma^{-1}\beta^*\zeta^{-1} & \gamma^{-1} + \gamma^{-1}\beta^*\zeta^{-1}\beta\gamma^{-1} \end{pmatrix} \times \begin{pmatrix} \mathbf{u}^*\Sigma^{-1}\mathbf{F}_1 \\ \vdots \\ \mathbf{u}^*\Sigma^{-1}\mathbf{F}_H \\ \mathbf{v}^*\Sigma^{-1}\mathbf{F} \\ d\Sigma^{-1}\mathbf{F} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{U}} \\ -\gamma^{-1}\beta^*\underline{\mathbf{U}} + \gamma^{-1}d\Sigma^{-1}\mathbf{F} \end{pmatrix}$$

where

$$\underline{\mathbf{U}} = \zeta^{-1} \begin{pmatrix} \mathbf{u}^*\Sigma^{-1}(\mathbf{F}_1 - \Sigma^{-1}\mathbf{N}_1d^2\gamma^{-1}\mathbf{F}) \\ \vdots \\ \mathbf{u}^*\Sigma^{-1}(\mathbf{F}_H - \Sigma^{-1}\mathbf{N}_Hd^2\gamma^{-1}\mathbf{F}) \\ \mathbf{v}^*\gamma^{-1}\Sigma^{-1}\mathbf{F} \end{pmatrix}.$$

This enables us to write down the posterior expectations that we need.

#### D. Evaluating the posterior in the PCA case

Continuing to suppress the reference to  $s$ , the only quantities that we need to evaluate in the PCA case are the determinant of  $\underline{\mathbf{L}}$  and  $E[\mathbf{X}_h]$  and  $\text{Cov}(\mathbf{X}_h, \mathbf{X}_h)$  for  $h = 1, \dots, H$ . Inverting the matrix  $\underline{\mathbf{L}}$  reduces to inverting the sparse matrix  $\underline{\mathbf{K}}$  defined by

$$\underline{\mathbf{K}} = \begin{pmatrix} \mathbf{a}_1 & & \mathbf{b}_1 \\ & \ddots & \vdots \\ & & \mathbf{a}_H & \mathbf{b}_H \\ \mathbf{b}_1^* & \dots & \mathbf{b}_H^* & \mathbf{I} + \mathbf{v}^*\Sigma^{-1}\mathbf{N}\mathbf{v} \end{pmatrix}$$

where

$$\mathbf{N} = \mathbf{N}_1 + \dots + \mathbf{N}_H$$

and

$$\begin{aligned} \mathbf{a}_h &= \mathbf{I} + \mathbf{u}^*\Sigma^{-1}\mathbf{N}_h\mathbf{u} \\ \mathbf{b}_h &= \mathbf{u}^*\Sigma^{-1}\mathbf{N}_h\mathbf{v} \end{aligned}$$

for  $h = 1, \dots, H$ .

Recall that if a positive definite matrix  $\mathbf{X}$  is given then a factorization of the form  $\mathbf{X} = \mathbf{T}^*\mathbf{T}$  where  $\mathbf{T}$  is an upper triangular matrix is called a Cholesky decomposition of  $\mathbf{X}$ ; we will write  $\mathbf{T} = \mathbf{X}^{1/2}$  to indicate that these conditions are satisfied. One way of inverting  $\underline{\mathbf{K}}$  is to use the block Cholesky algorithm given in [9] to find an upper triangular block matrix  $\underline{\mathbf{T}}$  having the same sparsity structure as  $\underline{\mathbf{K}}$  such that  $\underline{\mathbf{K}} = \underline{\mathbf{T}}^*\underline{\mathbf{T}}$ . The non-zero blocks of  $\underline{\mathbf{T}}$  are calculated as follows. For  $h = 1, \dots, H$ :

$$\begin{aligned} \mathbf{T}_{hh} &= \mathbf{K}_{hh}^{1/2} \\ \mathbf{T}_{h,H+1} &= \mathbf{T}_{hh}^{*-1}\mathbf{K}_{h,H+1} \end{aligned}$$

and

$$\mathbf{T}_{H+1,H+1} = \left( \mathbf{K}_{H+1,H+1} - \sum_{h=1}^H \mathbf{T}_{h,H+1}^* \mathbf{T}_{h,H+1} \right)^{1/2}.$$

As in the previous section, set  $\mathbf{F}_h = \mathbf{F}_h(s, \mathbf{m})$  for  $h = 1, \dots, H$  and set  $\mathbf{F} = \mathbf{F}_1 + \dots + \mathbf{F}_H$ . By Theorem 2, for  $h = 1, \dots, H$ , the posterior expectation  $E[\underline{\mathbf{X}}]$  is given by solving the equation  $\underline{\mathbf{K}}\underline{\mathbf{U}} = \underline{\mathbf{B}}$  for  $\underline{\mathbf{U}}$  where

$$\underline{\mathbf{B}} = \begin{pmatrix} \mathbf{u}^*\Sigma^{-1}\mathbf{F}_1 \\ \vdots \\ \mathbf{u}^*\Sigma^{-1}\mathbf{F}_H \\ \mathbf{v}^*\Sigma^{-1}\mathbf{F} \end{pmatrix}.$$

This can be solved by back substitution in the usual way. First solve the equation  $\underline{\mathbf{T}}^*\underline{\mathbf{C}} = \underline{\mathbf{B}}$  for  $\underline{\mathbf{C}}$  (taking advantage of the sparsity of  $\underline{\mathbf{T}}$ ):

$$\begin{aligned} \mathbf{C}_h &= \mathbf{T}_{hh}^{*-1}\mathbf{B}_h \quad (h = 1, \dots, H) \\ \mathbf{C}_{H+1} &= \mathbf{T}_{H+1,H+1}^{*-1} \left( \mathbf{B}_{H+1} - \sum_{h=1}^H \mathbf{T}_{h,H+1}^* \mathbf{C}_h \right) \end{aligned}$$

Then solve the equation  $\underline{\mathbf{T}}\underline{\mathbf{U}} = \underline{\mathbf{C}}$  for  $\underline{\mathbf{U}}$ :

$$\begin{aligned} \mathbf{U}_{H+1} &= \mathbf{T}_{H+1,H+1}^{-1}\mathbf{C}_{H+1} \\ \mathbf{U}_h &= \mathbf{T}_{hh}^{-1}(\mathbf{C}_h - \mathbf{T}_{h,H+1}\mathbf{U}_{H+1}) \quad (h = 1, \dots, H). \end{aligned}$$

As for the conditional covariances they are given by

$$\text{Cov}(\mathbf{X}_h, \mathbf{X}_h) = \begin{pmatrix} \mathbf{S}_{hh} & \mathbf{S}_{h,H+1} \\ \mathbf{S}_{H+1,h} & \mathbf{S}_{H+1,H+1} \end{pmatrix}$$

for  $h = 1, \dots, H$  where  $\underline{\mathbf{S}}$  is the inverse of  $\underline{\mathbf{K}}$ . Note that the only blocks of  $\underline{\mathbf{S}}$  that we need are those which correspond to the non-zero blocks of  $\underline{\mathbf{K}}$ . (Although we had to calculate all of the blocks of  $\zeta^{-1}$  in the previous subsection we do not have to calculate all of the blocks of  $\underline{\mathbf{K}}$ .) The blocks that we need can be calculated as follows:

$$\mathbf{S}_{H+1,H+1} = \mathbf{T}_{H+1,H+1}^{-1}\mathbf{T}_{H+1,H+1}^{*-1}$$

and for  $h = 1, \dots, H$

$$\begin{aligned} \mathbf{S}_{h,H+1} &= -\mathbf{T}_{hh}^{-1}\mathbf{T}_{h,H+1}\mathbf{S}_{H+1,H+1} \\ \mathbf{S}_{hh} &= \mathbf{T}_{hh}^{-1}\mathbf{T}_{hh}^{*-1} - \mathbf{T}_{hh}^{-1}\mathbf{T}_{h,H+1}\mathbf{S}_{h,H+1}^*. \end{aligned}$$

Finally the determinant of  $\underline{\mathbf{L}}$  is just the determinant of  $\underline{\mathbf{K}}$  and this is given by

$$|\mathbf{K}| = \prod_{h=1}^{H+1} |\mathbf{T}_{hh}|^2$$

which is easily evaluated since the determinant of a triangular matrix is just the product of the diagonal elements.

#### E. HMM adaptation

Suppose that we have a recording  $h$  of speaker  $s$  we wish to construct a HMM adapted to both the speaker and the recording. Set

$$\mathbf{w} = \begin{pmatrix} \mathbf{u} & \mathbf{v} \end{pmatrix}$$

and

$$\mathbf{X}_h(s) = \begin{pmatrix} \mathbf{x}_h(s) \\ \mathbf{y}(s) \end{pmatrix}$$

so that (5) can be written in the form

$$\mathbf{M}_h(s) = \mathbf{m} + \mathbf{w}\mathbf{X}_h(s) + \mathbf{d}\mathbf{z}(s).$$

Then the posterior distribution of  $\mathbf{M}_h(s)$  has mean  $\hat{\mathbf{M}}_h(s)$  given by

$$\mathbf{m} + \mathbf{w}E[\mathbf{X}_h(s)] + \mathbf{d}E[\mathbf{z}(s)]$$

and covariance matrix

$$\text{Cov}(\mathbf{w}\mathbf{X}_h(s) + \mathbf{d}\mathbf{z}(s), \mathbf{w}\mathbf{X}_h(s) + \mathbf{d}\mathbf{z}(s)).$$

Let  $\mathbf{D}_h(s)$  be the diagonal matrix obtained by setting the off-diagonal entries of this matrix to be 0. That is,

$$\begin{aligned} \mathbf{D}_h(s) = & \text{diag}(\mathbf{w} \text{Cov}(\mathbf{X}_h(s), \mathbf{X}_h(s))\mathbf{w}^* \\ & + 2 \text{diag}(\mathbf{w} \text{Cov}(\mathbf{X}_h(s), \mathbf{z}(s))\mathbf{d} \\ & + \mathbf{d} \text{diag}(\text{Cov}(\mathbf{z}(s), \mathbf{z}(s)))\mathbf{d}). \end{aligned}$$

For each mixture component  $c = 1, \dots, C$  let  $\hat{M}_{hc}(s)$  be the  $c$ th block of  $\hat{\mathbf{M}}_h(s)$  and let  $D_{hc}(s)$  be the  $c$ th block of  $\mathbf{D}_h(s)$ . Applying the Bayesian predictive classification principle as in [2], we can construct a HMM adapted to the given speaker and recording by assigning the mean vector  $\hat{M}_{hc}(s)$  and the diagonal covariance matrix  $\Sigma_c + D_{hc}(s)$  to the mixture component  $c$  for  $c = 1, \dots, C$ . Whether this type of variance adaptation is useful in practice is a matter for experimentation; our experience in [2] suggests that simply copying the variances from a speaker- and channel-independent HMM may give better results.

As we mentioned in Section II D, the posterior expectations and covariances needed to implement MAP HMM adaptation can be calculated either in batch mode or sequentially, that is, using speaker-independent hyperparameters (5) and all of the recordings of the speaker or using speaker-dependent hyperparameters (7) and only the current recording  $h$ .

In the classical MAP case (where  $\mathbf{u} = \mathbf{0}$  and  $\mathbf{v} = \mathbf{0}$ ), the calculations in Section III C above show that, if MAP adaptation is carried out in batch mode, then for each recording  $h$ ,

$$\hat{\mathbf{M}}_h(s) = \mathbf{m} + (\mathbf{I} + \mathbf{d}^2 \Sigma^{-1} \mathbf{N}(s))^{-1} \mathbf{d}^2 \Sigma^{-1} \mathbf{F}(s, \mathbf{m})$$

where

$$\begin{aligned} \mathbf{N}(s) &= \sum_{h=1}^{H(s)} \mathbf{N}_h(s) \\ \mathbf{F}(s, \mathbf{m}) &= \sum_{h=1}^{H(s)} \mathbf{F}_h(s, \mathbf{m}). \end{aligned}$$

Since  $\mathbf{u} = \mathbf{0}$ , the posterior distribution of  $\mathbf{M}_h(s)$  is the same for all recordings  $h$  (channel effects are not modeled in classical MAP). Denoting the common value of  $\hat{\mathbf{M}}_h(s)$  for all recordings by  $\hat{\mathbf{M}}(s)$  and writing  $\mathbf{r} = \mathbf{d}^{-2} \Sigma$  we obtain

$$\hat{\mathbf{M}}(s) = \mathbf{m} + (\mathbf{r} + \mathbf{N}(s))^{-1} \mathbf{F}(s, \mathbf{m}).$$

The diagonal entries of  $\mathbf{r}$  are referred to as ‘relevance factors’ in [6]. They can be interpreted by observing that if, for each  $i = 1, \dots, CF$ ,  $m_i$  denotes the  $i$ th entry of  $\mathbf{m}$  and similarly for  $r_i$  and  $M_i(s)$ , then the MAP estimate of  $M_i(s)$

is the same as the maximum likelihood estimate calculated by treating  $m_i$  as an extra observation of  $M_i(s)$  occurring with frequency  $r_i$ . Relevance factors are usually estimated empirically (after being tied across all mixture components and acoustic feature dimensions). However the hyperparameter estimation algorithms that we will develop will enable us to bring the maximum likelihood principle to bear on the problem of estimating  $\mathbf{d}$  and  $\Sigma$  so they provide a principled way of estimating relevance factors (without any tying).

#### F. The marginal distribution of the observable variables

Here we show how to evaluate the likelihood function  $P_\Lambda(\underline{\mathcal{X}}(s))$  defined by (6). This is the complete likelihood function for the factor analysis model in the sense that this term is used in the EM literature. Its primary role is to serve as a diagnostic for verifying the implementation of the EM algorithms for hyperparameter estimation that we will present. It also serves as the basis for constructing the likelihood ratio statistics used for speaker verification in [8]. The proof of Theorem 3 is formally identical to the proof of Proposition 2 in [2].

*Theorem 3: For each speaker  $s$ ,*

$$\begin{aligned} \log P_\Lambda(\underline{\mathcal{X}}(s)) &= G_{\Sigma}(s, \mathbf{m}) - \frac{1}{2} \log |\underline{\mathbf{L}}(s)| \\ &\quad + \frac{1}{2} E[\underline{\mathbf{X}}^*(s)] \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m}). \end{aligned}$$

*Proof:* By (6)

$$P_\Lambda(\underline{\mathcal{X}}(s)) = \int P_\Lambda(\underline{\mathcal{X}}(s) | \underline{\mathbf{X}}) N(\underline{\mathbf{X}} | \underline{\mathbf{0}}, \underline{\mathbf{I}}) d\underline{\mathbf{X}}.$$

and, by Theorem 1, we can write this as

$$\begin{aligned} \log P_\Lambda(\underline{\mathcal{X}}(s)) &= G_{\Sigma}(s, \mathbf{m}) \\ &\quad + \log \int \exp(H_\Lambda(s, \underline{\mathbf{X}})) N(\underline{\mathbf{X}} | \underline{\mathbf{0}}, \underline{\mathbf{I}}) d\underline{\mathbf{X}} \end{aligned}$$

where

$$\begin{aligned} H_\Lambda(s, \underline{\mathbf{X}}) &= \underline{\mathbf{X}}^* \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m}) \\ &\quad - \frac{1}{2} \underline{\mathbf{X}}^* \underline{\mathbf{V}}^*(s) \underline{\mathbf{N}}(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{V}}(s) \underline{\mathbf{X}} \end{aligned}$$

so that

$$\begin{aligned} H_\Lambda(s, \underline{\mathbf{X}}) - \frac{1}{2} \underline{\mathbf{X}}^* \underline{\mathbf{X}} &= \underline{\mathbf{X}}^* \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m}) \\ &\quad - \frac{1}{2} \underline{\mathbf{X}}^* \underline{\mathbf{L}}(s) \underline{\mathbf{X}}. \end{aligned}$$

To evaluate the integral we appeal to the formula for the Fourier-Laplace transform of the Gaussian kernel [10]. If  $N(\underline{\mathbf{X}} | \underline{\mathbf{0}}, \underline{\mathbf{L}}^{-1}(s))$  denotes the Gaussian kernel with mean  $\mathbf{0}$  and covariance matrix  $\underline{\mathbf{L}}(s)$  then

$$\begin{aligned} &\int \exp(\underline{\mathbf{X}}^* \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m})) N(\underline{\mathbf{X}} | \underline{\mathbf{0}}, \underline{\mathbf{L}}^{-1}(s)) d\underline{\mathbf{X}} \\ &= \exp\left(-\frac{1}{2} \log |\underline{\mathbf{L}}(s)| + \frac{1}{2} \underline{\mathbf{F}}^*(s, \mathbf{m}) \underline{\Sigma}^{-1}(s) \underline{\mathbf{V}}(s) \right. \\ &\quad \left. \times \underline{\mathbf{L}}^{-1}(s) \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m})\right). \end{aligned}$$

By Theorem 2, the latter expression can be written as

$$\exp\left(-\frac{1}{2} \log |\underline{\mathbf{L}}(s)| + \frac{1}{2} E[\underline{\mathbf{X}}(s)]^* \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m})\right)$$



so

$$\begin{aligned} \log P_{\Lambda}(\underline{\mathcal{X}}(s)) &= G_{\Sigma}(s, \mathbf{m}) - \frac{1}{2} \log |\underline{\mathbf{L}}(s)| \\ &\quad + \frac{1}{2} E[\underline{\mathbf{X}}(s)]^* \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m}) \end{aligned}$$

as required. ■

#### IV. MAXIMUM LIKELIHOOD HYPERPARAMETER ESTIMATION

We now turn to the problem of estimating the hyperparameter set  $\Lambda$  from a training set comprising several speakers with multiple recordings for each speaker (such as one of the Switchboard databases). We continue to assume that, for each speaker  $s$  and recording  $h$ , all frames have been aligned with mixture components.

Both of the estimation algorithms that we present in this and in the next section are EM algorithms which when applied iteratively produce a sequence of estimates of  $\Lambda$  having the property that the complete likelihood of the training data, namely

$$\prod_s \log P_{\Lambda}(\underline{\mathcal{X}}(s))$$

where  $s$  ranges over the training speakers, increases from one iteration to the next. The major difference between the two approaches is that we will impose constraints on the speaker and channel spaces in the next section but not in this one. Accordingly we will refer to the approach presented in this section simply as maximum likelihood estimation.

Since our concern is with speaker-independent estimation of  $\Lambda$ , it is generally reasonable to estimate  $\mathbf{m}$  by speaker-independent Baum-Welch training in the usual way so will concentrate on the problem of how to estimate the remaining hyperparameters if  $\mathbf{m}$  is given. The basic idea in the maximum likelihood estimation algorithm is to extend the argument used to prove Proposition 3 of [2] to handle  $\mathbf{u}$  and  $\mathbf{d}$  as well as  $\mathbf{v}$  and  $\Sigma$ . This entails accumulating the following statistics over the training set on each iteration where the posterior expectations are calculated using the current estimates of the hyperparameters,  $s$  ranges over the training speakers and, for each speaker  $s$ , the recordings are labeled  $h = 1, \dots, H(s)$ . Set

$$\mathbf{X}_h(s) = \begin{pmatrix} \mathbf{x}_h(s) \\ \mathbf{y}(s) \end{pmatrix}$$

for each speaker  $s$  and recording  $h$ . The accumulators are

$$N_c = \sum_s \sum_{h=1}^{H(s)} N_{hc}(s) \quad (10)$$

$$\mathfrak{A}_c = \sum_s \sum_{h=1}^{H(s)} N_{hc}(s) E[\mathbf{X}_h(s) \mathbf{X}_h^*(s)] \quad (11)$$

$$\mathfrak{B} = \sum_s \sum_{h=1}^{H(s)} N_h(s) E[\mathbf{z}(s) \mathbf{X}_h^*(s)] \quad (12)$$

$$\mathfrak{C} = \sum_s \sum_{h=1}^{H(s)} \mathbf{F}_h(s, \mathbf{m}) E[\mathbf{X}_h^*(s)] \quad (13)$$

$$\mathbf{a} = \sum_s \text{diag}(\mathbf{N}(s) E[\mathbf{z}(s) \mathbf{z}^*(s)]) \quad (14)$$

$$\mathbf{b} = \sum_s \text{diag}(\mathbf{F}(s, \mathbf{m}) E[\mathbf{z}^*(s)]) \quad (15)$$

In (10) and (11),  $c$  ranges over all mixture components and the quantities  $\mathbf{N}(s)$  and  $\mathbf{F}(s, \mathbf{m})$  in (14) and (15) are defined by

$$\begin{aligned} \mathbf{N}(s) &= \sum_{h=1}^{H(s)} \mathbf{N}_h(s) \\ \mathbf{F}(s, \mathbf{m}) &= \sum_{h=1}^{H(s)} \mathbf{F}_h(s, \mathbf{m}) \end{aligned}$$

for each training speaker  $s$ .

*Theorem 4: Let  $\mathbf{m}$  be given. Suppose we have a hyperparameter set of the form  $(\mathbf{m}, \mathbf{u}_0, \mathbf{v}_0, \mathbf{d}_0, \Sigma_0)$  and we use it to calculate the accumulators (10) – (15). Define a new hyperparameter set  $(\mathbf{m}, \mathbf{u}, \mathbf{v}, \mathbf{d}, \Sigma)$  as follows.*

- (i) Set  $\mathbf{w} = \begin{pmatrix} \mathbf{u} & \mathbf{v} \end{pmatrix}$ . For each mixture component  $c = 1, \dots, C$  and for each  $f = 1, \dots, F$ , set  $i = (c-1)F + f$  and let  $w_i$  denote the  $i$ th row of  $\mathbf{w}$  and  $d_i$  the  $i$ th entry of  $\mathbf{d}$ . Then  $w_i$  and  $d_i$  are defined by the equation

$$\begin{pmatrix} w_i & d_i \end{pmatrix} \begin{pmatrix} \mathfrak{A}_c & \mathfrak{B}_i^* \\ \mathfrak{B}_i & \mathfrak{a}_i \end{pmatrix} = \begin{pmatrix} \mathfrak{C}_i & \mathfrak{b}_i \end{pmatrix} \quad (16)$$

where  $\mathfrak{B}_i$  is the  $i$ th row of  $\mathfrak{B}$ ,  $\mathfrak{a}_i$  is the  $i$ th entry of  $\mathbf{a}$ ,  $\mathfrak{C}_i$  is the  $i$ th row of  $\mathfrak{C}$  and  $\mathfrak{b}_i$  is the  $i$ th entry of  $\mathbf{b}$ .

- (ii) Let  $\mathfrak{M}$  be the diagonal  $CF \times CF$  matrix given by

$$\mathfrak{M} = \text{diag}(\mathfrak{C}\mathbf{w}^* + \mathbf{b}\mathbf{d}).$$

Set

$$\Sigma = \mathbf{N}^{-1} \left( \sum_s \mathbf{S}(s, \mathbf{m}) - \mathfrak{M} \right) \quad (17)$$

where  $\mathbf{N}$  is the  $CF \times CF$  diagonal matrix whose diagonal blocks are  $N_1 \mathbf{I}, \dots, N_C \mathbf{I}$  and

$$\mathbf{S}(s, \mathbf{m}) = \sum_{h=1}^{H(s)} \mathbf{S}_h(s, \mathbf{m}) \quad (18)$$

for each training speaker  $s$ .

Then if  $\Lambda_0 = (\mathbf{m}, \mathbf{u}_0, \mathbf{v}_0, \mathbf{d}_0, \Sigma_0)$  and  $\Lambda = (\mathbf{m}, \mathbf{u}, \mathbf{v}, \mathbf{d}, \Sigma)$ ,

$$\sum_s \log P_{\Lambda}(\underline{\mathcal{X}}(s)) \geq \sum_s \log P_{\Lambda_0}(\underline{\mathcal{X}}(s))$$

where  $s$  ranges over the training speakers.

Before turning to the proof of this theorem note that (16) is a low dimensional system of equations ( $R_S + R_C + 1$  equations in  $R_S + R_C + 1$  unknowns where  $R_S$  is the number of speaker factors and  $R_C$  the number of channel factors) so there is no difficulty in solving it in practice. In fact a very efficient algorithm can be developed by first calculating the Cholesky decomposition of the  $(R_S + R_C) \times (R_S + R_C)$  matrix  $\mathfrak{A}_c$  and using this to calculate the Cholesky decompositions of the matrices

$$\begin{pmatrix} \mathfrak{A}_c & \mathfrak{B}_i^* \\ \mathfrak{B}_i & \mathfrak{a}_i \end{pmatrix}$$

for  $f = 1, \dots, F$ . There is a further simplification in the PCA case since only the equation

$$w_i \mathbf{a}_c = \mathbf{c}_i$$

needs to be solved.

In our experience calculating the posterior expectations needed to accumulate the statistics (10) – (15) accounts for almost all of the computation needed to implement the algorithm. As we have seen, calculating posterior expectations in the PCA case is much less expensive than in the general case; furthermore, since  $\mathbf{z}(s)$  plays no role in this case, the accumulators (12), (14) and (15) are not needed. Assuming that the accumulators (11) are stored on disk rather than in memory, this results in a 50% reduction in memory requirements in the PCA case.

*Proof:* We begin by constructing an EM auxiliary function with the  $\underline{\mathbf{X}}(s)$ 's as hidden variables. By Jensen's inequality,

$$\begin{aligned} & \sum_s \int \left( \log \frac{P_{\Lambda}(\underline{\mathbf{X}}, \underline{\mathcal{X}}(s))}{P_{\Lambda_0}(\underline{\mathbf{X}}, \underline{\mathcal{X}}(s))} \right) P_{\Lambda_0}(\underline{\mathbf{X}} | \underline{\mathcal{X}}(s)) d\underline{\mathbf{X}} \\ & \leq \sum_s \log \int \frac{P_{\Lambda}(\underline{\mathbf{X}}, \underline{\mathcal{X}}(s))}{P_{\Lambda_0}(\underline{\mathbf{X}}, \underline{\mathcal{X}}(s))} P_{\Lambda_0}(\underline{\mathbf{X}} | \underline{\mathcal{X}}(s)) d\underline{\mathbf{X}}. \end{aligned}$$

Since the right hand side of this inequality simplifies to

$$\sum_s \log P_{\Lambda}(\underline{\mathcal{X}}(s)) - \sum_s \log P_{\Lambda_0}(\underline{\mathcal{X}}(s)),$$

the total log likelihood of the training data can be increased by choosing the new estimates of the hyperparameters so as to make the left hand side positive. Since for each speaker  $s$ ,

$$P_{\Lambda}(\underline{\mathbf{X}}, \underline{\mathcal{X}}(s)) = P_{\Lambda}(\underline{\mathcal{X}}(s) | \underline{\mathbf{X}}) N(\underline{\mathbf{X}} | \underline{\mathbf{0}}, \underline{\mathbf{I}})$$

and similarly for  $P_{\Lambda_0}(\underline{\mathbf{X}}, \underline{\mathcal{X}}(s))$ , the left hand side can be written as

$$\sum_s (\mathcal{A}_{\Lambda}(\underline{\mathcal{X}}(s)) - \mathcal{A}_{\Lambda_0}(\underline{\mathcal{X}}(s)))$$

where

$$\mathcal{A}_{\Lambda}(\underline{\mathcal{X}}) = \int \log P_{\Lambda}(\underline{\mathcal{X}} | \underline{\mathbf{X}}) P_{\Lambda_0}(\underline{\mathbf{X}} | \underline{\mathcal{X}}) d\underline{\mathbf{X}}.$$

Thus we can ensure that

$$\sum_s \log P_{\Lambda}(\underline{\mathcal{X}}(s)) \geq \sum_s \log P_{\Lambda_0}(\underline{\mathcal{X}}(s))$$

by choosing  $\Lambda$  so that

$$\sum_s (\mathcal{A}_{\Lambda}(\underline{\mathcal{X}}(s)) - \mathcal{A}_{\Lambda_0}(\underline{\mathcal{X}}(s))) \geq 0$$

and this can be achieved by maximizing

$$\sum_s \mathcal{A}_{\Lambda}(\underline{\mathcal{X}}(s))$$

with respect to  $\Lambda$ . We refer this as the auxiliary function. We derive (16) and (17) by calculating the gradients of the auxiliary function with respect to  $\mathbf{w}$ ,  $\mathbf{d}$  and  $\Sigma$  and setting these to zero.

Observe that for each speaker  $s$

$$\mathcal{A}_{\Lambda}(\underline{\mathcal{X}}(s)) = E[\log P_{\Lambda}(\underline{\mathcal{X}}(s) | \underline{\mathbf{X}}(s))].$$

So, by Theorem 1, the auxiliary function can be written as

$$\sum_s G_{\Sigma}(s, \mathbf{m}) + \sum_s E[H_{\Lambda}(s, \underline{\mathbf{X}}(s))]. \quad (19)$$

The first term here is independent of  $\mathbf{w}$  and  $\mathbf{d}$  and the second term can be expressed in terms of  $\mathbf{w}$ ,  $\mathbf{d}$  and  $\Sigma$  as follows:

$$\begin{aligned} & \sum_s E[H_{\Lambda}(s, \underline{\mathbf{X}}(s))] \\ & = \sum_s E \left[ \underline{\mathbf{X}}^*(s) \underline{\mathbf{V}}^*(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{F}}(s, \mathbf{m}) \right. \\ & \quad \left. - \frac{1}{2} \underline{\mathbf{X}}^*(s) \underline{\mathbf{V}}^*(s) \underline{\mathbf{N}}(s) \underline{\Sigma}^{-1}(s) \underline{\mathbf{V}}(s) \underline{\mathbf{X}}(s) \right] \\ & = \sum_s \text{tr} \left( \underline{\Sigma}^{-1}(s) \left( \underline{\mathbf{F}}(s, \mathbf{m}) E[\underline{\mathbf{X}}(s)]^* \underline{\mathbf{V}}^*(s) \right. \right. \\ & \quad \left. \left. - \frac{1}{2} \underline{\mathbf{N}}(s) \underline{\mathbf{V}}(s) E[\underline{\mathbf{X}}(s) \underline{\mathbf{X}}^*(s)] \underline{\mathbf{V}}^*(s) \right) \right) \\ & = \sum_s \text{tr} \left( \underline{\Sigma}^{-1}(s) \left( \underline{\mathbf{F}}(s, \mathbf{m}) E[\underline{\mathbf{Q}}(s)]^* \right. \right. \\ & \quad \left. \left. - \frac{1}{2} \underline{\mathbf{N}}(s) E[\underline{\mathbf{Q}}(s) \underline{\mathbf{Q}}^*(s)] \right) \right) \\ & = \sum_s \sum_{h=1}^{H(s)} \text{tr} \left( \underline{\Sigma}^{-1} \left( \underline{\mathbf{F}}_h(s, \mathbf{m}) E[\underline{\mathbf{O}}_h(s)]^* \right. \right. \\ & \quad \left. \left. - \frac{1}{2} \underline{\mathbf{N}}_h(s) E[\underline{\mathbf{O}}_h(s) \underline{\mathbf{O}}_h^*(s)] \right) \right) \end{aligned} \quad (20)$$

where

$$\underline{\mathbf{O}}_h(s) = \mathbf{w} \mathbf{X}_h(s) + \mathbf{d} \mathbf{z}(s)$$

for each speaker  $s$  and recording  $h$ .

Setting the gradient of the auxiliary function with respect to  $\mathbf{w}$  equal to zero gives

$$\begin{aligned} & \sum_s \sum_{h=1}^{H(s)} \mathbf{N}_h(s) (\mathbf{w} E[\mathbf{X}_h(s) \mathbf{X}_h^*(s)] + \mathbf{d} E[\mathbf{z}(s) \mathbf{z}^*(s)]) \\ & = \sum_s \sum_{h=1}^{H(s)} \mathbf{F}_h(s, \mathbf{m}) E[\mathbf{X}_h^*(s)] \end{aligned} \quad (21)$$

and setting the gradient with respect to  $\mathbf{d}$  equal to zero implies that the diagonal of

$$\sum_s \sum_{h=1}^{H(s)} \mathbf{N}_h(s) (\mathbf{w} E[\mathbf{X}_h(s) \mathbf{z}^*(s)] + \mathbf{d} E[\mathbf{z}(s) \mathbf{z}^*(s)]) \quad (22)$$

is equal to the diagonal of

$$\sum_s \mathbf{F}(s, \mathbf{m}) E[\mathbf{z}^*(s)]. \quad (23)$$

For each mixture component  $c = 1, \dots, C$  and for each  $f = 1, \dots, F$ , set  $i = (c-1)F + f$ . Equating the  $i$ th row of the left-hand side of (21) with the  $i$ th row of the right-hand side gives

$$w_i \mathbf{a}_c + d_i \mathbf{b}_i = \mathbf{c}_i. \quad (24)$$

Since  $N_h(s)$  is diagonal, the diagonal of

$$\sum_s \sum_{h=1}^{H(s)} N_h(s) \mathbf{w} E[\mathbf{X}_h(s) \mathbf{z}^*(s)]$$

is the same as the diagonal of

$$\sum_s \sum_{h=1}^{H(s)} \mathbf{w} E[\mathbf{X}_h(s) \mathbf{z}^*(s)] N_h(s)$$

which is to say  $\mathbf{w} \mathfrak{B}^*$ . The  $i$ th diagonal entry of this is  $w_i \mathfrak{B}_i^*$  (that is, the product of  $w_i$  and the transpose of the  $i$ th row of  $\mathfrak{B}$ ), so equating the  $i$ th diagonal entry of (22) with the  $i$ th diagonal entry of (23) gives

$$w_i \mathfrak{B}_i^* + d_i \mathbf{a}_i = \mathbf{b}_i. \quad (25)$$

Combining (24) and (25) gives (16):

$$\begin{pmatrix} w_i & d_i \end{pmatrix} \begin{pmatrix} \mathfrak{A}_c & \mathfrak{B}_i^* \\ \mathfrak{B}_i & \mathbf{a}_i \end{pmatrix} = \begin{pmatrix} \mathfrak{C}_i & \mathbf{b}_i \end{pmatrix}.$$

In order to derive (17) it is more convenient to calculate the gradient of the auxiliary function with respect to  $\Sigma^{-1}$  than  $\Sigma$ . Note first that if we postmultiply the left-hand side of (21) by  $\mathbf{w}^*$  and (22) by  $\mathbf{d}$  and sum the results we obtain

$$\sum_s \sum_{h=1}^{H(s)} N_h(s) E[\mathbf{O}_h(s) \mathbf{O}_h^*(s)]$$

by the definition of  $\mathbf{O}_h(s)$ . On the other hand if we postmultiply the right-hand side of (21) by  $\mathbf{w}^*$  and (23) by  $\mathbf{d}$  and sum the results we obtain

$$\sum_{h=1}^{H(s)} \mathbf{F}_h(s, \mathbf{m}) E[\mathbf{O}_h^*(s)].$$

Since the diagonals of (22) and (23) are equal, it follows that

$$\begin{aligned} & \text{diag} \left( \sum_s \sum_{h=1}^{H(s)} N_h(s) E[\mathbf{O}_h(s) \mathbf{O}_h^*(s)] \right) \\ &= \text{diag} \left( \sum_s \sum_{h=1}^{H(s)} \mathbf{F}_h(s, \mathbf{m}) E[\mathbf{O}_h^*(s)] \right) \end{aligned} \quad (26)$$

at a critical point of the auxiliary function.

Using the expression (19) to calculate the derivative of the second term of (18) with respect to  $\Sigma^{-1}$  in the direction of  $\mathbf{e}$  where  $\mathbf{e}$  is any diagonal matrix having the same dimensions as  $\Sigma$  we obtain

$$\begin{aligned} & \sum_s \sum_{h=1}^{H(s)} \text{tr}((\mathbf{F}_h(s, \mathbf{m}) E[\mathbf{O}_h^*(s)] \\ & - \frac{1}{2} N_h(s) E[\mathbf{O}_h(s) \mathbf{O}_h^*(s)]) \mathbf{e}). \end{aligned}$$

By (26) this simplifies to

$$\begin{aligned} & \frac{1}{2} \sum_s \sum_{h=1}^{H(s)} \text{tr}(\text{diag}(\mathbf{F}_h(s, \mathbf{m}) E[\mathbf{O}_h^*(s)]) \mathbf{e}) \\ &= \frac{1}{2} \text{tr}(\mathfrak{M} \mathbf{e}). \end{aligned} \quad (27)$$

It remains to calculate the gradient of the first term of (18) with respect to  $\Sigma^{-1}$ . Note that, for each  $c = 1, \dots, C$ , the derivative of  $\log |\Sigma_c^{-1}|$  with respect to  $\Sigma_c^{-1}$  in the direction of  $\mathbf{e}_c$  is  $\text{tr}(\Sigma_c \mathbf{e}_c)$ . Hence, for each speaker  $s$ , the derivative of  $G_\Sigma(s, \mathbf{m})$  with respect to  $\Sigma^{-1}$  in the direction of  $\mathbf{e}$  is

$$\begin{aligned} & \frac{1}{2} \sum_{h=1}^{H(s)} \sum_{c=1}^C \text{tr}((N_{hc}(s) \Sigma_c - S_{hc}(s, \mathbf{m}_c)) \mathbf{e}_c) \\ &= \frac{1}{2} \sum_{h=1}^{H(s)} \text{tr}((N_h(s) \Sigma - S_h(s, \mathbf{m})) \mathbf{e}) \\ &= \frac{1}{2} \text{tr}((N(s) \Sigma - S(s, \mathbf{m})) \mathbf{e}) \end{aligned} \quad (28)$$

Combining (28) and (29) we obtain the derivative of the auxiliary function with respect to  $\Sigma^{-1}$  in the direction of  $\mathbf{e}$ , namely

$$\frac{1}{2} \text{tr} \left( \left( N \Sigma - \sum_s S(s, \mathbf{m}) + \mathfrak{M} \right) \mathbf{e} \right)$$

In order for this to be equal to zero for all  $\mathbf{e}$  we must have

$$\Sigma = N^{-1} \left( \sum_s S(s, \mathbf{m}) - \mathfrak{M} \right)$$

which is (17). (The estimates for the covariance matrices can be shown to be positive definite with a little bit of extra work.) ■

Baum-Welch training is an appropriate way of estimating the supervector  $\mathbf{m}$  provided that a speaker- and channel-independent HMM (rather than the speaker- and channel-adapted HMM's of Section III E above) is used to align the training data and the training set is the same as that used to estimate the other speaker-independent hyperparameters, namely  $\mathbf{u}, \mathbf{v}, \mathbf{d}$  and  $\Sigma$ . However, even in this situation, it would be more natural to estimate  $\mathbf{m}$  using the same criterion as the other hyperparameters. Theorem 4 could be extended to estimate all 5 hyperparameters simultaneously by means of a standard trick, namely setting

$$\begin{aligned} \mathbf{V} &= \begin{pmatrix} \mathbf{v} & \mathbf{m} \end{pmatrix} \\ \text{and } \mathbf{Y}(s) &= \begin{pmatrix} \mathbf{y}(s) \\ 1 \end{pmatrix} \end{aligned}$$

for each speaker  $s$  so that  $\mathbf{m} + \mathbf{v} \mathbf{y}(s) = \mathbf{V} \mathbf{Y}(s)$  thereby eliminating  $\mathbf{m}$ . Of course this requires re-deriving the estimation formulas in the statement of Theorem 4. A simpler if less efficient way of dealing with this problem is to derive an EM algorithm for estimating  $\mathbf{m}$  on the assumption that the other hyperparameters are given. By alternating between the EM algorithms in Theorems 4 and 5 on successive iterations the entire hyperparameter set  $(\mathbf{m}, \mathbf{u}, \mathbf{v}, \mathbf{d}, \Sigma)$  can be estimated in a consistent way.

*Theorem 5: Let  $\mathbf{u}, \mathbf{v}, \mathbf{d}$  and  $\Sigma$  be given. Suppose we have a hyperparameter set of the form  $(\mathbf{m}_0, \mathbf{u}, \mathbf{v}, \mathbf{d}, \Sigma)$  and we use it to calculate the statistic  $\mathfrak{D}$  defined by*

$$\mathfrak{D} = \sum_s \sum_{h=1}^{H(s)} N_h(s) E[\mathbf{O}_h(s)]$$

where  $s$  ranges over the training speakers and

$$\mathbf{O}_h(s) = \mathbf{w}\mathbf{X}_h(s) + \mathbf{d}\mathbf{z}(s)$$

for each recording  $h$ . For each  $c = 1, \dots, C$ , let

$$m_c = \frac{1}{N_c} \left( \sum_s \sum_{h=1}^{H(s)} F_{hc}(s, 0) - \mathfrak{D}_c \right) \quad (29)$$

where  $\mathfrak{D}_c$  is the  $c$ th block of  $\mathfrak{D}$ . Let  $\mathbf{m}$  be the supervector obtained by concatenating  $m_1, \dots, m_C$ . Then if  $\Lambda_0 = (\mathbf{m}, \mathbf{u}_0, \mathbf{v}_0, \mathbf{d}_0, \Sigma_0)$  and  $\Lambda = (\mathbf{m}, \mathbf{u}, \mathbf{v}, \mathbf{d}, \Sigma)$ ,

$$\sum_s \log P_\Lambda(\mathcal{X}(s)) \geq \sum_s \log P_{\Lambda_0}(\mathcal{X}(s))$$

where  $s$  ranges over the training speakers.

*Proof:* We use the same auxiliary function as in the proof of the previous theorem but regard it as a function of  $\mathbf{m}$  rather than as a function of  $\mathbf{u}, \mathbf{v}, \mathbf{d}$  and  $\Sigma$ . Setting the gradient of the auxiliary function with respect to  $\mathbf{m}$  equal to zero gives

$$\sum_s \sum_{h=1}^{H(s)} \mathbf{F}_h(s, \mathbf{m}) = \sum_s \sum_{h=1}^{H(s)} N_h(s) E[\mathbf{O}_h(s)].$$

Solving this equation for  $\mathbf{m}$  gives (29).  $\blacksquare$

It can be shown that at a fixed point of the maximum likelihood estimation algorithm, the following condition must hold:

$$\frac{1}{S} \sum_s E[\mathbf{y}(s)\mathbf{y}^*(s)] = \mathbf{I}$$

where  $S$  is the number of speakers in the training set and the sum extends over all training speakers. (This is to be expected since, for each speaker  $s$ , the prior distribution of  $\mathbf{y}(s)$  is the standard normal distribution. Similar conditions apply to the other hidden variables.) However, our experience with the maximum likelihood estimation algorithm has been that, no matter how often it is iterated on a training set, this condition is never satisfied. The diagonal entries of the left hand side of this equation are always much less than unity in practice and, to compensate for this, the eigenvalues of  $\mathbf{v}\mathbf{v}^*$  are larger than seems reasonable. (The eigenvalues of  $\mathbf{v}\mathbf{v}^*$  can be calculated by observing that they are the same as the eigenvalues of the low dimensional matrix  $\mathbf{v}^*\mathbf{v}$ .) On the other hand, when the HMM likelihood of the training set is calculated with speaker- and channel-adapted HMM's derived from maximum likelihood estimates of the speaker-independent hyperparameters, the results do seem to be reasonable and this suggests that the maximum likelihood estimation algorithm does a fair job of estimating the orientations of the speaker and channel spaces. (Thus the eigenvectors of  $\mathbf{v}\mathbf{v}^*$  seem to be well estimated if not the eigenvalues.) We don't have any explanation for this anomaly but we mention it because it suggests that other ways of hyperparameter estimation might be worth studying.

## V. MINIMUM DIVERGENCE HYPERPARAMETER ESTIMATION

In this section we use the methods introduced in [7] to study the problem of estimating the hyperparameters in situations

where the orientations of the speaker space and channel space are known. An example of how this type of situation can arise was given at the end of the last section; we will give other examples later.

Throughout this section we fix a hyperparameter set  $(\mathbf{m}_0, \mathbf{u}_0, \mathbf{v}_0, \mathbf{d}_0, \Sigma_0)$  which we denote by  $\Lambda_0$ . Let  $C_0$  denote the range of  $\mathbf{u}_0\mathbf{u}_0^*$  (the channel space) and let  $S_0$  denote the range of  $\mathbf{v}_0\mathbf{v}_0^*$  translated by  $\mathbf{m}_0$  (the speaker space).

Consider a model of the form

$$\left. \begin{aligned} \mathbf{M}(s) &= \mathbf{m}_0 + \mathbf{v}_0\mathbf{y}(s) + \mathbf{d}_0\mathbf{z}(s) \\ \mathbf{M}_h(s) &= \mathbf{M}(s) + \mathbf{u}_0\mathbf{x}_h(s) \end{aligned} \right\} \quad (30)$$

where, instead of assuming that  $\mathbf{y}(s), \mathbf{z}(s)$  and  $\mathbf{x}_h(s)$  have standard normal distributions, we assume that  $\mathbf{y}(s)$  is normally distributed with mean  $\mathbf{m}_y$  and covariance matrix  $\mathbf{K}_{yy}$ ,  $\mathbf{z}(s)$  is normally distributed with mean  $\mu_z$  and diagonal covariance matrix  $\mathbf{K}_{zz}$ , and  $\mathbf{x}_h(s)$  is normally distributed with mean  $\mathbf{0}$  and covariance matrix  $\mathbf{K}_{xx}$ . Set

$$\lambda = (\mathbf{m}_y, \mu_z, \mathbf{K}_{xx}, \mathbf{K}_{yy}, \mathbf{K}_{zz}, \Sigma).$$

This model is easily seen to be equivalent to the model

$$\left. \begin{aligned} \mathbf{M}(s) &= \mathbf{m} + \mathbf{v}\mathbf{y}(s) + \mathbf{d}\mathbf{z}(s) \\ \mathbf{M}_h(s) &= \mathbf{M}(s) + \mathbf{u}\mathbf{x}_h(s) \end{aligned} \right\}$$

where  $\mathbf{x}(s), \mathbf{y}(s)$  and  $\mathbf{x}_h(s)$  have standard normal distributions and

$$\begin{aligned} \mathbf{m} &= \mathbf{m}_0 + \mathbf{v}_0\mu_y + \mathbf{d}_0\mu_z \\ \mathbf{u} &= \mathbf{u}_0\mathbf{K}_{xx}^{1/2} \\ \mathbf{v} &= \mathbf{v}_0\mathbf{K}_{yy}^{1/2} \\ \mathbf{d} &= \mathbf{d}_0\mathbf{K}_{zz}^{1/2}. \end{aligned}$$

Set  $\Lambda = (\mathbf{m}, \mathbf{u}, \mathbf{v}, \mathbf{d}, \Sigma)$ . The channel space for the factor analysis model defined by  $\Lambda$  is  $C_0$  and the speaker space is parallel to  $S_0$ . Conversely, any hyperparameter set that satisfies these conditions arises in this way from some sextuple  $\lambda$ . (Note in particular that if  $\lambda_0 = (\mathbf{0}, \mathbf{0}, \mathbf{I}, \mathbf{I}, \mathbf{I}, \Sigma_0)$  then the hyperparameter set corresponding to  $\lambda_0$  is just  $\Lambda_0$ .) Thus the problem of estimating a hyperparameter subject to the constraints on the speaker and channel spaces can be formulated in terms of estimating the sextuple  $\lambda$ .

We need to introduce some notation in order to do likelihood calculations with the model defined by (30). For each speaker  $s$ , let  $Q_\lambda(\mathbf{X}(s))$  be the distribution on  $\mathbf{X}(s)$  defined by  $\lambda$ . That is,  $Q_\lambda(\mathbf{X}(s))$  is the normal distribution with mean  $\underline{\mu}$  and covariance matrix  $\underline{\mathbf{K}}$  where  $\underline{\mu}$  is the vector whose blocks are  $\mathbf{0}, \dots, \mathbf{0}, \mathbf{m}_y, \mu_z$  and  $\underline{\mathbf{K}}$  is the block diagonal matrix whose diagonal blocks are  $\mathbf{K}_{xx}, \dots, \mathbf{K}_{xx}, \mathbf{K}_{yy}, \mathbf{K}_{zz}$ .

The distribution  $Q_\lambda(\mathbf{X}(s))$  gives rise to a distribution  $Q_\lambda(\mathbf{M}(s))$  for the random vector  $\mathbf{M}(s)$  defined by

$$\mathbf{M}(s) = \begin{pmatrix} \mathbf{M}_1(s) \\ \vdots \\ \mathbf{M}_{H(s)}(s) \end{pmatrix}$$

where  $\mathbf{M}_h(s)$  is given by (30) for  $h = 1, \dots, H(s)$ . This distribution is the image of the distribution  $Q_\lambda(\mathbf{X}(s))$  under

the linear transformation  $\underline{T}_{\Lambda_0}$  defined by

$$\underline{T}_{\Lambda_0} \begin{pmatrix} x_1 \\ \vdots \\ x_{H(s)} \\ y \\ z \end{pmatrix} = \begin{pmatrix} M_1 \\ \vdots \\ M_{H(s)} \end{pmatrix}.$$

In other words,  $Q_{\lambda}(\underline{M}(s))$  is the normal distribution with mean  $\underline{T}_{\Lambda_0} \underline{\mu}$  and covariance matrix  $\underline{T}_{\Lambda_0} \underline{K} \underline{T}_{\Lambda_0}^*$ .

The likelihood of  $\underline{\mathcal{X}}(s)$  can be calculated using  $Q_{\lambda}(\underline{M}(s))$  as follows:

$$Q_{\lambda}(\underline{\mathcal{X}}(s)) = \int P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{M}) Q_{\lambda}(\underline{M}) d\underline{M}$$

where we have use the notation  $P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{M})$  to emphasize the fact that  $\Sigma$  is the only hyperparameter that plays a role in calculating the conditional distribution of  $\underline{\mathcal{X}}(s)$  given that  $\underline{M}(s) = \underline{M}$ . Of course  $P_{\Lambda}(\underline{\mathcal{X}}(s)) = Q_{\lambda}(\underline{\mathcal{X}}(s))$  where  $\Lambda$  is the hyperparameter set corresponding to  $\lambda$  so we also have

$$P_{\Lambda}(\underline{\mathcal{X}}(s)) = \int P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{M}) Q_{\lambda}(\underline{M}) d\underline{M}. \quad (31)$$

We will derive the minimum divergence estimation procedure from the following theorem.

*Theorem 6: Suppose  $\lambda$  satisfies the following conditions:*

$$\begin{aligned} & \sum_s D(P_{\Lambda_0}(\underline{X}|\underline{\mathcal{X}}(s)) \| Q_{\lambda}(\underline{X})) \\ & \leq \sum_s D(P_{\Lambda_0}(\underline{X}|\underline{\mathcal{X}}(s)) \| Q_{\lambda_0}(\underline{X})) \end{aligned}$$

where  $D(\cdot|\cdot)$  indicates the Kullback-Leibler divergence and

$$\begin{aligned} & \sum_s E[\log P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{M})] \\ & \geq \sum_s E[\log P_{\Sigma_0}(\underline{\mathcal{X}}(s)|\underline{M})] \end{aligned}$$

where,  $s$  ranges over the training speakers and  $E[\cdot]$  indicates a posterior expectation calculated with the initial hyperparameter set  $\Lambda_0$ . Then

$$\sum_s \log P_{\Lambda}(\underline{\mathcal{X}}(s)) \geq \sum_s \log P_{\Lambda_0}(\underline{\mathcal{X}}(s))$$

where  $s$  ranges over the training speakers.

In order to prove this theorem we need two lemmas.

*Lemma 1: For any speaker  $s$ ,*

$$\begin{aligned} \log \frac{P_{\Lambda}(\underline{\mathcal{X}}(s))}{P_{\Lambda_0}(\underline{\mathcal{X}}(s))} & \geq -E[\log P_{\Sigma_0}(\underline{\mathcal{X}}(s)|\underline{M})] \\ & \quad + E[\log P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{M})] \\ & \quad + D(Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s)) \| Q_{\lambda_0}(\underline{M})) \\ & \quad - D(Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s)) \| Q_{\lambda}(\underline{M})). \end{aligned}$$

*Proof:* This follows from Jensen's inequality

$$\begin{aligned} & \log \int U(\underline{M}) Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s)) d\underline{M} \\ & \geq \int \log U(\underline{M}) Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s)) d\underline{M} \end{aligned}$$

where

$$U(\underline{M}) = \frac{P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{M}) Q_{\lambda}(\underline{M})}{P_{\Sigma_0}(\underline{\mathcal{X}}(s)|\underline{M}) Q_{\lambda_0}(\underline{M})}.$$

Note that

$$\begin{aligned} \frac{Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s))}{Q_{\lambda_0}(\underline{M})} & = \frac{Q_{\lambda_0}(\underline{\mathcal{X}}(s)|\underline{M})}{Q_{\lambda_0}(\underline{\mathcal{X}}(s))} \\ & = \frac{P_{\Sigma_0}(\underline{\mathcal{X}}(s)|\underline{M})}{P_{\Lambda_0}(\underline{\mathcal{X}}(s))} \end{aligned}$$

so that

$$\begin{aligned} & U(\underline{M}) Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s)) \\ & = \frac{P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{M}) Q_{\lambda}(\underline{M})}{P_{\Sigma_0}(\underline{\mathcal{X}}(s)|\underline{M}) Q_{\lambda_0}(\underline{M})} Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s)) \\ & = \frac{P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{M}) Q_{\lambda}(\underline{M})}{P_{\Lambda_0}(\underline{\mathcal{X}}(s))}. \end{aligned}$$

By (31) the left hand side of the inequality simplifies as follows

$$\begin{aligned} & \int U(\underline{M}) Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s)) d\underline{M} \\ & = \frac{1}{P_{\Lambda_0}(\underline{\mathcal{X}}(s))} \int P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{M}) Q_{\lambda}(\underline{M}) d\underline{M} \\ & = \frac{P_{\Lambda}(\underline{\mathcal{X}}(s))}{P_{\Lambda_0}(\underline{\mathcal{X}}(s))} \end{aligned}$$

and the result follows immediately.  $\blacksquare$

Note that Lemma 1 refers to divergences between distributions on  $\underline{M}(s)$  whereas the statement of Theorem 6 refers to divergences between distributions on  $\underline{X}(s)$ . These two types of divergence are related as follows.

*Lemma 2: For each speaker  $s$ ,*

$$\begin{aligned} & D(P_{\Lambda_0}(\underline{X}|\underline{\mathcal{X}}(s)) \| Q_{\lambda}(\underline{X})) \\ & = D(Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s)) \| Q_{\lambda}(\underline{M})) \\ & \quad + \int D(Q_{\lambda_0}(\underline{X}|\underline{M}) \| Q_{\lambda}(\underline{X}|\underline{M})) Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s)) d\underline{M} \end{aligned}$$

*Proof:* The chain rule for divergences states that, for any distributions  $\pi(\underline{X}, \underline{M})$  and  $\rho(\underline{X}, \underline{M})$ ,

$$\begin{aligned} & D(\pi(\underline{X}, \underline{M}) \| \rho(\underline{X}, \underline{M})) \\ & = D(\pi(\underline{M}) \| \rho(\underline{M})) \\ & \quad + \int D(\pi(\underline{X}|\underline{M}) \| \rho(\underline{X}|\underline{M})) \pi(\underline{M}) d\underline{M}. \end{aligned}$$

The result follows by applying the chain rule to the distributions defined by setting

$$\begin{aligned} \pi(\underline{X}, \underline{M}) & = P_{\Lambda_0}(\underline{X}|\underline{\mathcal{X}}(s)) \delta_{\underline{T}_{\Lambda_0}(\underline{X})}(\underline{M}) \\ \rho(\underline{X}, \underline{M}) & = Q_{\lambda}(\underline{X}|\underline{\mathcal{X}}(s)) \delta_{\underline{T}_{\Lambda_0}(\underline{X})}(\underline{M}) \end{aligned}$$

where  $\delta$  denotes the Dirac delta function.  $\blacksquare$

Since divergences are non-negative it follows from Lemma 2 that

$$\begin{aligned} & D(P_{\Lambda_0}(\underline{X}|\underline{\mathcal{X}}(s)) \| Q_{\lambda}(\underline{X})) \\ & \geq D(Q_{\lambda_0}(\underline{M}|\underline{\mathcal{X}}(s)) \| Q_{\lambda}(\underline{M})). \end{aligned} \quad (32)$$

This is a ‘data processing inequality’ in the sense in which this term is used in information theory where the ‘processing’

is the operation of transforming  $\underline{\mathbf{X}}$  into  $\underline{\mathbf{M}}$  by  $\underline{\mathbf{T}}_{\Lambda_0}$ . In the case where  $\mathbf{d} = \mathbf{0}$  and the  $CF \times R_C$  matrix  $\mathbf{v}_0$  is of rank  $R_C$ , the transformation  $\underline{\mathbf{T}}_{\Lambda_0}$  is injective. Thus, for each  $\mathbf{M}$ , the distributions  $Q_{\lambda_0}(\underline{\mathbf{X}}|\underline{\mathbf{M}})$  and  $Q_{\lambda}(\underline{\mathbf{X}}|\underline{\mathbf{M}})$  are point distributions concentrated on the same point so their divergence is 0. So the inequality becomes an equality in this case (no information is lost). These divergences also vanish in the case  $\lambda = \lambda_0$  so that

$$\begin{aligned} D(P_{\Lambda_0}(\underline{\mathbf{X}}|\underline{\mathcal{X}}(s))||Q_{\lambda_0}(\underline{\mathbf{X}})) \\ = D(Q_{\lambda_0}(\underline{\mathbf{M}}|\underline{\mathcal{X}}(s))||Q_{\lambda_0}(\underline{\mathbf{M}})) \end{aligned} \quad (33)$$

irrespective of whether or not  $\mathbf{d} = \mathbf{0}$ .

We can now prove the assertion of Theorem 6, namely that

$$\sum_s P_{\Lambda}(\underline{\mathcal{X}}(s)) \geq \sum_s P_{\Lambda_0}(\underline{\mathcal{X}}(s))$$

where the sums extend over all speakers in the training set. Note that

$$\begin{aligned} & \sum_s (E[\log P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{\mathbf{M}})] \\ & \quad - D(Q_{\lambda_0}(\underline{\mathbf{M}}|\underline{\mathcal{X}}(s))||Q_{\lambda}(\underline{\mathbf{M}}))) \\ & \geq \sum_s (E[\log P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{\mathbf{M}})] \\ & \quad - D(P_{\Lambda_0}(\underline{\mathbf{X}}|\underline{\mathcal{X}}(s))||Q_{\lambda}(\underline{\mathbf{X}}))) \\ & \geq \sum_s (E[\log P_{\Sigma_0}(\underline{\mathcal{X}}(s)|\underline{\mathbf{M}})] \\ & \quad - D(P_{\Lambda_0}(\underline{\mathbf{X}}|\underline{\mathcal{X}}(s))||Q_{\lambda_0}(\underline{\mathbf{X}}))) \\ & = \sum_s (E[\log P_{\Sigma_0}(\underline{\mathcal{X}}(s)|\underline{\mathbf{M}})] \\ & \quad - D(Q_{\lambda_0}(\underline{\mathbf{M}}|\underline{\mathcal{X}}(s))||Q_{\lambda_0}(\underline{\mathbf{M}}))) \end{aligned}$$

The first inequality here follows from (32), the second by the hypotheses of the Theorem and the concluding equality from (33). The result now follows from Lemma 1.

Theorem 6 guarantees that we can find a hyperparameter set  $\Lambda$  which respects the subspace constraints and fits the training data better than  $\Lambda_0$  by choosing  $\lambda$  so as to minimize

$$\sum_s D(P_{\Lambda_0}(\underline{\mathbf{X}}|\underline{\mathcal{X}}(s))||Q_{\lambda}(\underline{\mathbf{X}}))$$

and maximize

$$\sum_s E[\log P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{\mathbf{M}})].$$

This leads to the following set of re-estimation formulas.

*Theorem 7: Define a new hyperparameter set  $\Lambda$  by setting  $\Lambda = (\mathbf{m}, \mathbf{u}, \mathbf{v}, \mathbf{d}, \Sigma)$  where*

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{v}_0 \mu_y + \mathbf{d}_0 \mu_z \quad (34)$$

$$\mathbf{u} = \mathbf{u}_0 \mathbf{K}_{xx}^{1/2} \quad (35)$$

$$\mathbf{v} = \mathbf{v}_0 \mathbf{K}_{yy}^{1/2} \quad (36)$$

$$\mathbf{d} = \mathbf{d}_0 \mathbf{K}_{zz}^{1/2} \quad (37)$$

$$\begin{aligned} \Sigma = & N^{-1} \sum_s \sum_{h=1}^{H(s)} \left( \mathbf{S}_h(s, \mathbf{m}_0) \right. \\ & - 2 \text{diag}(\mathbf{F}_h(s, \mathbf{m}_0) E[\mathbf{O}_h^*(s)]) \\ & \left. + \text{diag}(E[\mathbf{O}_h(s) \mathbf{O}_h^*(s)] \mathbf{N}_h(s)) \right). \end{aligned} \quad (38)$$

Here  $\mathbf{O}_h(s) = \mathbf{M}_h(s) - \mathbf{m}_0$  for each speaker  $s$  and recording  $h$  and

$$\begin{aligned} \mu_y &= \frac{1}{S} \sum_s E[\mathbf{y}(s)] \\ \mu_z &= \frac{1}{S} \sum_s E[\mathbf{z}(s)] \\ \mathbf{K}_{xx} &= \frac{1}{H} \sum_s \sum_{h=1}^{H(s)} E[\mathbf{x}_h(s) \mathbf{x}_h^*(s)] \\ \mathbf{K}_{yy} &= \frac{1}{S} \sum_s E[\mathbf{y}(s) \mathbf{y}^*(s)] - \mu_y \mu_y^* \\ \mathbf{K}_{zz} &= \text{diag} \left( \frac{1}{S} \sum_s E[\mathbf{z}(s) \mathbf{z}^*(s)] - \mu_z \mu_z^* \right) \\ \mathbf{N} &= \sum_s \sum_{h=1}^{H(s)} \mathbf{N}_h(s) \\ H &= \sum_s H(s); \end{aligned}$$

$S$  is the number of training speakers, the sums extend over all speakers in the training set and the posterior expectations are calculated using the hyperparameter set  $\Lambda_0$ . Then  $\Lambda$  satisfies the conditions of Theorem 6 so that

$$\sum_s P_{\Lambda}(\underline{\mathcal{X}}(s)) \geq \sum_s P_{\Lambda_0}(\underline{\mathcal{X}}(s))$$

where  $s$  ranges over the training speakers.

*Proof:* In order to minimize

$$\sum_{s=1} D(P_{\Lambda_0}(\underline{\mathbf{X}}|\underline{\mathcal{X}}(s))||Q_{\lambda}(\underline{\mathbf{X}})),$$

we use the formula for the divergence of two Gaussian distributions [7]. For each speaker  $s$ ,  $P_{\Lambda_0}(\underline{\mathbf{X}}|\underline{\mathcal{X}}(s))$  is the Gaussian distribution with mean  $E[\underline{\mathbf{X}}(s)]$  and covariance matrix  $\underline{\mathbf{L}}^{-1}(s)$  by Theorem 2. So if  $\lambda = (\mathbf{m}_y, \mu_z, \mathbf{K}_{xx}, \mathbf{K}_{yy}, \mathbf{K}_{zz}, \Sigma)$ ,

$$\begin{aligned} & D(P_{\Lambda_0}(\underline{\mathbf{X}}|\underline{\mathcal{X}}(s))||Q_{\lambda}(\underline{\mathbf{X}})) \\ &= -\frac{1}{2} \log |\underline{\mathbf{L}}^{-1}(s) \underline{\mathbf{K}}^{-1}(s)| \\ & \quad + \frac{1}{2} \text{tr}((\underline{\mathbf{L}}^{-1}(s) + (E[\underline{\mathbf{X}}(s)] - \underline{\boldsymbol{\mu}}(s)) \\ & \quad \times (E[\underline{\mathbf{X}}(s)] - \underline{\boldsymbol{\mu}}(s))^* \underline{\mathbf{K}}^{-1}(s)) \\ & \quad - \frac{1}{2} (R_S + H(s) R_C + CF) \end{aligned} \quad (39)$$

where  $\underline{\boldsymbol{\mu}}(s)$  is the vector whose blocks are  $\mathbf{0}, \dots, \mathbf{0}, \mathbf{m}_y, \mu_z$  (the argument  $s$  is needed to indicate that there are  $H(s)$  repetitions of  $\mathbf{0}$ ) and  $\underline{\mathbf{K}}(s)$  is the block diagonal matrix whose diagonal blocks are  $\mathbf{K}_{xx}, \dots, \mathbf{K}_{xx}, \mathbf{K}_{yy}, \mathbf{K}_{zz}$ . The formulas (34) – (37) are derived by differentiating (39) with respect to  $\mathbf{m}_y, \mu_z, \mathbf{K}_{xx}^{1/2}, \mathbf{K}_{yy}^{1/2}$  and  $\mathbf{K}_{zz}^{1/2}$  and setting the derivatives to zero.

In order to maximize

$$\sum_s E[\log P_{\Sigma}(\underline{\mathcal{X}}(s)|\underline{\mathbf{M}})],$$

observe that by the argument used in proving Theorem 1,

$$\begin{aligned} & \log P_{\Sigma}(\mathcal{X}_h(s) | \mathbf{M}_h(s)) \\ &= \sum_{c=1}^C N_{hc}(s) \log \frac{1}{(2\pi)^{F/2} |\Sigma_c|^{1/2}} \\ & \quad - \frac{1}{2} \text{tr}(\Sigma^{-1} \mathbf{S}_h(s, \mathbf{m}_0)) \\ & \quad + \text{tr}(\Sigma^{-1} \mathbf{F}_h(s, \mathbf{m}_0) \mathbf{O}_h^*(s)) \\ & \quad - \frac{1}{2} \text{tr}(\Sigma^{-1} \mathbf{O}_h(s) \mathbf{O}_h^*(s) \mathbf{N}_h(s)) \end{aligned}$$

for each speaker  $s$  and recording  $h$ . Thus

$$\begin{aligned} & \sum_s E[\log P_{\Sigma}(\mathcal{X}(s) | \mathbf{M})] \\ &= \sum_{c=1}^C N_{hc}(s) \log \frac{1}{(2\pi)^{F/2} |\Sigma_c|^{1/2}} \\ & \quad - \frac{1}{2} \text{tr}(\Sigma^{-1} \mathbf{S}_h(s, \mathbf{m}_0)) \\ & \quad + \text{tr}(\Sigma^{-1} \mathbf{F}_h(s, \mathbf{m}_0) E[\mathbf{O}_h^*(s)]) \\ & \quad - \frac{1}{2} \text{tr}(\Sigma^{-1} E[\mathbf{O}_h(s) \mathbf{O}_h^*(s)] \mathbf{N}_h(s)) \end{aligned}$$

and (38) follows by differentiating this with respect to  $\Sigma^{-1}$  and setting the derivative to zero. ■

## VI. ADAPTING THE FACTOR ANALYSIS MODEL FROM ONE SPEAKER POPULATION TO ANOTHER

We have assumed so far that we have at our disposal a training set in which there are multiple recordings of each speaker. If the training set does not have this property then the speaker-independent hyperparameter estimation algorithms given in the preceding sections cannot be expected to give reasonable results. (To take an extreme example, consider the case where we have just one recording per speaker as in the enrollment data for the target speakers in the current NIST speaker verification evaluations. It is impossible to distinguish between speaker and channel effects in this situation.) We have attempted to deal with this problem by using an ancillary training set in which speakers are recorded in multiple sessions in order to model channel effects. The idea is to first estimate a full set of hyperparameters  $\mathbf{m}, \mathbf{u}, \mathbf{v}, \mathbf{d}$  and  $\Sigma$  on the ancillary training set and then, holding  $\mathbf{u}$  and  $\Sigma$  fixed, re-estimate  $\mathbf{m}, \mathbf{v}$  and  $\mathbf{d}$  on the original training set. In other words, we keep the hyperparameters associated with channel space fixed and re-estimate only the hyperparameters associated with the speaker space. This can be done using either the maximum likelihood or the divergence minimization approach but our experience with NIST data sets [8] has been that only the latter approach is effective. In other words, it is necessary to keep the orientation of the speaker space fixed as well as that of the channel space rather than change it to fit the target speaker population (in order to avoid overtraining on the very limited amount of enrollment data provided by NIST).

The maximum likelihood approach entails accumulating the following statistics over the original training set on each iteration in addition to the statistics  $\mathbf{a}$  and  $\mathbf{b}$  defined by (14) and (15) where the posterior expectations are calculated using

the current values of the hyperparameters and  $s$  ranges over the training speakers:

$$\mathfrak{S}_c = \sum_s \sum_{h=1}^{H(s)} N_{hc}(s) E[\mathbf{y}(s) \mathbf{y}^*(s)] \quad (40)$$

$$\mathfrak{T}_c = \sum_s \sum_{h=1}^{H(s)} N_{hc}(s) E[\mathbf{x}_h(s) \mathbf{y}^*(s)] \quad (41)$$

$$\mathfrak{U} = \sum_s \sum_{h=1}^{H(s)} \mathbf{N}_h(s) E[\mathbf{z}(s) \mathbf{y}^*(s)] \quad (42)$$

$$\mathfrak{V} = \sum_s \sum_{h=1}^{H(s)} \mathbf{N}_h(s) E[\mathbf{z}(s) \mathbf{x}_h^*(s)] \quad (43)$$

$$\mathfrak{W} = \sum_s \sum_{h=1}^{H(s)} \mathbf{F}_h(s, \mathbf{m}) E[\mathbf{y}^*(s)]. \quad (44)$$

In (40) and (41),  $c$  ranges from 1 to  $C$ . These statistics can easily be extracted from the statistics defined in (11) – (15).

*Theorem 8:* Let  $\mathbf{m}_0, \mathbf{u}_0$  and  $\Sigma_0$  be given. Suppose we have a hyperparameter set  $\Lambda_0$  of the form  $(\mathbf{m}_0, \mathbf{u}_0, \mathbf{v}_0, \mathbf{d}_0, \Sigma)$  and we use it to calculate the accumulators defined in (40) – (44).

Define  $\mathbf{v}$  and  $\mathbf{d}$  as follows. For each mixture component  $c = 1, \dots, C$  and for each  $f = 1, \dots, F$ , set  $i = (c-1)F + f$  and let  $u_i$  denote the  $i$ th row of  $\mathbf{u}_0$ ,  $v_i$  the  $i$ th row of  $\mathbf{v}$  and  $d_i$  the  $i$ th entry of  $\mathbf{d}$ . Then  $v_i$  and  $d_i$  are defined by the equation

$$\begin{pmatrix} v_i & d_i \end{pmatrix} \begin{pmatrix} \mathfrak{S}_c & \mathfrak{U}_i^* \\ \mathfrak{U}_i & \mathfrak{a}_i \end{pmatrix} = \begin{pmatrix} \mathfrak{W}_i - u_i \mathfrak{T}_c & \mathfrak{b}_i - u_i \mathfrak{V}_i^* \end{pmatrix}$$

where  $\mathfrak{U}_i, \mathfrak{V}_i$  and  $\mathfrak{W}_i$  denote the  $i$ th rows of  $\mathfrak{U}, \mathfrak{V}$  and  $\mathfrak{W}$  and  $\mathfrak{a}_i$  and  $\mathfrak{b}_i$  denote the  $i$ th entries of  $\mathfrak{a}$  and  $\mathfrak{b}$ .

If  $\Lambda = (\mathbf{m}_0, \mathbf{u}_0, \mathbf{v}, \mathbf{d}, \Sigma_0)$  then

$$\sum_s \log P_{\Lambda}(\mathcal{X}(s)) \geq \sum_s \log P_{\Lambda_0}(\mathcal{X}(s))$$

where  $s$  ranges over the speakers in the (original) training set.

*Proof:* The idea is to use the auxiliary function that we used in proving Theorem 4, regarding it as a function of  $\mathbf{v}$  and  $\mathbf{d}$  alone. ■

The minimum divergence algorithm is much easier to formulate since it is just a special case of Theorem 7.

*Theorem 9:* Suppose we are given a hyperparameter set  $\Lambda_0$  of the form  $(\mathbf{m}_0, \mathbf{u}_0, \mathbf{v}_0, \mathbf{d}_0, \Sigma_0)$ . Define a new hyperparameter set  $\Lambda$  of the form  $(\mathbf{m}, \mathbf{u}_0, \mathbf{v}, \mathbf{d}, \Sigma_0)$  by setting

$$\begin{aligned} \mathbf{m} &= \mathbf{m}_0 + \mathbf{v}_0 \boldsymbol{\mu}_y + \mathbf{d}_0 \boldsymbol{\mu}_z \\ \mathbf{v} &= \mathbf{v}_0 \mathbf{K}_{yy}^{1/2} \\ \mathbf{d} &= \mathbf{d}_0 \mathbf{K}_{zz}^{1/2} \end{aligned}$$

where  $\boldsymbol{\mu}_y, \boldsymbol{\mu}_z, \mathbf{K}_{yy}$  and  $\mathbf{K}_{zz}$  are defined in the statement of Theorem 7. Then

$$\sum_s \log P_{\Lambda}(\mathcal{X}(s)) \geq \sum_s \log P_{\Lambda_0}(\mathcal{X}(s))$$

where  $s$  ranges over the speakers in the (original) training set.

## VII. SPEAKER-DEPENDENT ESTIMATION OF THE HYPERPARAMETERS

Suppose we are given a set of hyperparameter estimates  $\Lambda_0$  such as the speaker-independent estimates derived in Section IV or Section V. We now explain how, given a speaker  $s$  and a collection of recordings  $\mathcal{X}(s)$ , we can use the posterior distribution of the hidden variables  $P_{\Lambda_0}(\mathbf{X}(s)|\mathcal{X}(s))$  to estimate a set of speaker-dependent hyperparameters, that is, a speaker dependent prior. Estimating this speaker-dependent prior plays a key role in our approach to speaker verification (it constitutes the procedure for enrolling target speakers [8], [1]).

Estimating a speaker-dependent prior from the posterior distribution of the hidden variables is not entirely straightforward because the priors specified by (5) do not constitute a conjugate family. Specifically, we assumed in (5) that  $\mathbf{y}(s)$  and  $\mathbf{z}(s)$  have standard normal distributions so that the random vector  $\mathbf{M}(s)$  which we defined by

$$\mathbf{M}(s) = \mathbf{m} + \mathbf{v}\mathbf{y}(s) + \mathbf{d}\mathbf{z}(s)$$

has covariance matrix  $\mathbf{v}\mathbf{v}^* + \mathbf{d}\mathbf{d}^*$ . However the posterior distribution of  $\mathbf{M}(s)$  has covariance matrix

$$\begin{pmatrix} \mathbf{v} & \mathbf{d} \end{pmatrix} \begin{pmatrix} \text{Cov}(\mathbf{y}(s), \mathbf{y}(s)) & \text{Cov}(\mathbf{y}(s), \mathbf{z}(s)) \\ \text{Cov}(\mathbf{z}(s), \mathbf{y}(s)) & \text{Cov}(\mathbf{z}(s), \mathbf{z}(s)) \end{pmatrix} \begin{pmatrix} \mathbf{v}^* \\ \mathbf{d}^* \end{pmatrix}$$

and this cannot generally be written in the form  $\mathbf{v}'\mathbf{v}'^* + \mathbf{d}'\mathbf{d}'^*$ . (Note however that it can be written in this form if  $\mathbf{d} = \mathbf{0}$  by Cholesky decomposition of  $\text{Cov}(\mathbf{y}(s), \mathbf{y}(s))$ .) The import of Theorem 10 is that it is reasonable to ignore the troublesome cross correlations in the posterior (namely  $\text{Cov}(\mathbf{y}(s), \mathbf{z}(s))$  and the off-diagonal terms in  $\text{Cov}(\mathbf{z}(s), \mathbf{z}(s))$ ) for the purpose of estimating a speaker dependent prior. The theorem is just a special case of Theorem 9 (namely the case where the training set consists of a single speaker).

*Theorem 10: Suppose we are given a speaker  $s$ , a hyperparameter set  $\Lambda_0$  where  $\Lambda_0 = (\mathbf{m}_0, \mathbf{u}_0, \mathbf{v}_0, \mathbf{d}_0, \Sigma_0)$  and a collection of recordings  $\mathcal{X}(s)$  indexed by  $h = 1, \dots, H(s)$ . Let  $\Lambda$  be the hyperparameter set  $(\mathbf{m}(s), \mathbf{u}_0, \mathbf{v}(s), \mathbf{d}(s), \Sigma_0)$  where*

$$\begin{aligned} \mathbf{m}(s) &= \mathbf{m}_0 + \mathbf{v}_0\mathbf{m}_{\mathbf{y}}(s) + \mathbf{d}_0\boldsymbol{\mu}_{\mathbf{z}}(s) \\ \mathbf{v}(s) &= \mathbf{v}_0\mathbf{K}_{\mathbf{y}\mathbf{y}}^{1/2}(s) \\ \mathbf{d}(s) &= \mathbf{d}_0\mathbf{K}_{\mathbf{z}\mathbf{z}}^{1/2}(s) \end{aligned}$$

and

$$\begin{aligned} \mathbf{m}_{\mathbf{y}}(s) &= E[\mathbf{y}(s)] \\ \boldsymbol{\mu}_{\mathbf{z}}(s) &= E[\mathbf{z}(s)] \\ \mathbf{K}_{\mathbf{y}\mathbf{y}}(s) &= \text{Cov}(\mathbf{y}(s), \mathbf{y}(s)) \\ \mathbf{K}_{\mathbf{z}\mathbf{z}}(s) &= \text{diag}(\text{Cov}(\mathbf{z}(s), \mathbf{z}(s))) \end{aligned}$$

where these posterior expectations and covariances are calculated using  $\Lambda_0$ . Then

$$P_{\Lambda}(\mathcal{X}(s)) \geq P_{\Lambda_0}(\mathcal{X}(s)).$$

If this speaker-dependent hyperparameter estimation is carried out recursively as successive recordings of the speaker become available, it can be used as a basis for progressive speaker adaptation for either speech recognition or speaker recognition.

But, because of the troublesome cross-correlations in the posterior, it is only in the case where  $\mathbf{d} = \mathbf{0}$  that this type of recursive estimation will give the same results as processing all of the speaker's recordings in batch mode.

## VIII. DISCUSSION

In this article we have sought to model session variability by means of a Gaussian distribution on supervectors. A natural extension which should probably be explored if a suitable training set can be found, would be to use Gaussian mixture distributions instead where each mixture component has associated with it a centroid in the supervector space and a supervector covariance matrix of low rank. (This is motivated in part by the feature mapping technique which treats channel effects as discrete. In [11] this type of mixture distribution is referred to as a mixture of probabilistic principal components analyzers). As far as we have been able to determine there is no obstacle in principle to extending the joint factor analysis model in this way (assuming of course that a suitable training set can be found) but there may be formidable computational obstacles.

A recurring theme in this article has been that life would be much easier if we could simply take  $\mathbf{d} = \mathbf{0}$ . The main reason is that if we are given a collection of recordings for a speaker, the calculation of the posterior distribution of the hidden variables in the joint factor analysis model is much easier in this case. If this calculation is done in batch mode (as in Section III), the computational cost is linear in the number of recordings rather than cubic. Furthermore, although we haven't fully developed the argument, the calculation can be done recursively in this case (as indicated in Section VII). If mixture distributions are incorporated into the model then it seems that the posterior calculation would *have* to be done recursively in order to avoid the combinatorial explosion that would take place if the recordings in batch mode and the possibility that each recording can be accounted for by a different mixture component is allowed. Since we don't have a recursive solution to the problem of calculating the posterior in the general case  $\mathbf{d} \neq \mathbf{0}$  it is not clear yet to what extent the possibility of mixture modeling can be explored in practice.

## REFERENCES

- [1] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," submitted to *IEEE Trans. Audio Speech and Language Processing*. [Online]. Available: <http://www.crim.ca/perso/patrick.kenny/>
- [2] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. Speech Audio Processing*, vol. 13, no. 3, May 2005. [Online]. Available: <http://www.crim.ca/perso/patrick.kenny/>
- [3] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 291–298, Apr. 1994.
- [4] P. Kenny, M. Mihoubi, and P. Dumouchel, "New MAP estimators for speaker recognition," in *Proc. Eurospeech*, Geneva, Switzerland, Sept. 2003. [Online]. Available: <http://www.crim.ca/perso/patrick.kenny/>
- [5] B. Moghaddam, "Principal manifolds and probabilistic subspaces for visual recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 6, pp. 780–788, June 2002.
- [6] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.



- [7] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker adaptation using an eigenphone basis," *IEEE Trans. Speech Audio Processing*, vol. 12, no. 6, Nov. 2004. [Online]. Available: <http://www.crim.ca/perso/patrick.kenny/>
- [8] —, "Speaker and session variability in GMM-based speaker verification," submitted to *IEEE Trans. Audio Speech and Language Processing*. [Online]. Available: <http://www.crim.ca/perso/patrick.kenny/>
- [9] P. A. Steeves. Block Cholesky algorithms. [Online]. Available: <http://webhome.idirect.com/logical/pdf/blk-choleski.pdf>
- [10] H. Stark and J. Woods, *Probability, Random Processes and Estimation Theory for Engineers*. Englewood Cliffs, New Jersey: Prentice Hall, 1986.
- [11] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analysers," *Neural Computation*, vol. 11, pp. 435–474, 1999.