Eigenvoice Modeling With Sparse Training Data

Patrick Kenny, *Member, IEEE*, Gilles Boulianne, *Member, IEEE*, and Pierre Dumouchel, *Member, IEEE*

Abstract—We derive an exact solution to the problem of maximum likelihood estimation of the supervector covariance matrix used in extended MAP (or EMAP) speaker adaptation and show how it can be regarded as a new method of eigenvoice estimation. Unlike other approaches to the problem of estimating eigenvoices in situations where speaker-dependent training is not feasible, our method enables us to estimate as many eigenvoices from a given training set as there are training speakers. In the limit as the amount of training data for each speaker tends to infinity, it is equivalent to cluster adaptive training.

Index Terms—Cluster adaptive training, eigenvoices, extended MAP (EMAP), speech recognition, speaker adaptation.

I. INTRODUCTION

E IGENVOICE modeling has proved to be effective in small vocabulary speech recognition tasks where enough data can be collected to carry out speaker-dependent training for large numbers of speakers [1]–[4]. Our objective in this article is to show how this type of modeling can be extended to situations where the training data is sparse in the sense that speaker-dependent training is not feasible.

Suppose, to begin with, we ask an oracle to supply speaker-dependent models for any speaker. If C denotes the total number of mixture components in a speaker model and F the dimension of the acoustic feature vectors then, for each speaker, we can concatenate the mean vectors associated with the mixture components to form a *supervector* of dimension CF. The idea behind eigenvoice modeling is that principal components analysis can be used to constrain these supervectors to lie in a low dimensional space with little loss of accuracy. This ensures that only a small number of parameters need to be estimated in order to enroll a new speaker. Thus, speaker adaptation saturates quickly in the sense that the recognition accuracy for a test speaker reaches a plateau after a small amount of adaptation data has been collected.

To flesh this out a bit, let M_0 and B denote the mean and covariance matrix of the supervectors for the speaker population. The basic assumption in eigenvoice modeling is that most of the eigenvalues of B are zero. This guarantees that the speaker supervectors are all contained in a linear manifold of low dimension, namely, the set of supervectors of the form $M_0 + O$ where O lies in the range of B. This set of supervectors is known as the *eigenspace*; The *eigenvoices* of the population are the eigenvectors of B corresponding to nonzero eigenvalues. (It will be helpful to bear in mind that the dimension of the eigenspace is

equal to the number of eigenvoices and this is equal to the rank of B.)

In order to construct a speaker-adapted HMM for a given speaker (absent the oracle) we need a mechanism to impose the constraint that the speaker's supervector lies in the eigenspace. In this article, we will use maximum *a posteriori* (MAP) estimation for this purpose rather than use eigenvoices explicitly. The idea behind MAP estimation is that we have a prior probability distribution for the speaker's supervector, namely, the Gaussian distribution with mean M_0 and covariance matrix B. Since this prior distribution is concentrated on the eigenspace, the same is true of the posterior distribution derived from it using the speaker's adaptation data. In particular, the mode of the posterior distribution lies in the eigenspace. Thus, MAP estimation of the speaker's supervector ensures that the constraint is satisfied. The term EMAP adaptation—E for extended [5]—is generally used to refer to MAP adaptation using a supervector covariance matrix B but, since the traditional usage assumes that B is invertible whereas, our concern in this article is with the case where B is of less than full rank, we will use the term eigenvoice MAP instead. The idea of integrating eigenvoice modeling with MAP speaker adaptation has been suggested by other authors [6]–[8].

Of course other mechanisms can be used to constrain the supervectors to lie in the eigenspace [2], [9], [10] but a good case can be made that MAP estimation is the most natural way. First, MAP adaptation implicitly takes account of the eigenvalues of B, as well as the eigenvectors. In fact, if we use the MAP approach, there is no reason in principle why we should make a hard decision to suppress the eigenvectors of B which correspond to minor eigenvalues (as required by other approaches). Secondly, MAP adaptation is asymptotically equivalent to speaker-dependent training as the amount of adaptation data increases. Thirdly, it is natural from a mathematical point of view since, as we will show in this article, it plays a central role in estimating eigenvoices by probabilistic principal components analysis. (In particular, no extra software needs to be developed for speaker adaptation if the eigenvoices are estimated in this way.) Finally, the computational burden of MAP adaptation is quite modest provided that the rank of B is reasonably small since it essentially boils down to inverting an $R \times R$ matrix where R is the rank of B [6]–[8]. The only difficulty with MAP adaptation arises in the case where R is large (more than a few thousand) in which case it is computationally intractable unless B is constrained in some way. (For example, if B is assumed to be of full rank and sparsity constraints are imposed on B^{-1}, MAP speaker adaptation can be implemented by viewing the posterior distribution as a Gaussian Markov random field [11].)

The main issue that needs to be addressed in order to implement an eigenvoice model is to estimate the population co-

variance matrix B. If large numbers of training speakers can be enlisted then we can take the sample covariance matrix of the training set supervectors as an estimate of B. But note that the rank of the sample covariance matrix is just the number of speakers in the training set. In most situations of interest this is far less than the dimension of the supervector space so the sample covariance matrix may not provide a reasonable estimate of the population covariance matrix. In such cases it may be necessary to impose some constraints on the population covariance matrix in order to produce an eigenspace of sufficiently high dimension that MAP adaptation has some hope of being asymptotically equivalent to speaker-dependent training. For instance, we could impose a block diagonal structure on B by ignoring the correlations between mixture components associated with different phonemes as in the seminal paper [5]. (Block diagonal constraints have the effect of boosting the rank of B by a factor of N where N is the number of blocks.) Alternatively, as we will explain, we can boost the rank of B by streaming the acoustic features. Or we could guarantee correct asymptotic behavior, at least in theory, simply by forcing the supervector covariance matrix to be of full rank. (Note that a singular covariance matrix can be made nonsingular by an arbitrarily small perturbation. For example, we could set the zero eigenvalues to a small positive value as in [6]. We say "in theory" because very large amounts of data—hundreds of sentences—may be needed for speaker adaptation to saturate with a covariance matrix of full rank [11].) Generally speaking, increasing the dimension of the eigenspace can be expected to slow down the rate at which speaker adaptation saturates so, in practice, it may not be possible to ensure that speaker adaptation saturates quickly *and* that it behaves like speaker-dependent training as the amount of adaptation data increases.

Assuming that a satisfactory tradeoff between these two considerations can be found, there remains the question of how to estimate the population covariance matrix if speaker-dependent training is not feasible. (Recall that we began our discussion by appealing to an oracle to supply the training speakers' supervectors.) In this article we will show how to estimate the principal eigenvectors of the population covariance matrix directly from the training data without having recourse to speaker-dependent models.

Our solution to this problem is a variant of the probabilistic principal components approach introduced in [12]. This methodology was developed in order to extend principal components analysis to handle priors specified by a mixture of Gaussians (rather than by a single Gaussian as in traditional principal components analysis). Priors of this type give rise to an interesting class of MAP estimators which, loosely speaking, are locally linear but globally non linear. These MAP estimators can be applied to a variety of data compression and pattern recognition tasks but in the context of eigenvoice modeling large numbers of training speakers seem to be needed to take advantage of them.

So we have to limit ourselves to the case of a single Gaussian as in conventional principal components analysis. (This is unfortunate since clear evidence of at least two modes—one male, one female—is presented in [9], [13]). The probabilistic approach still has an advantage over other approaches even in the unimodal case because it enables us to estimate as many eigenvoices as there are speakers in the training set. Even if it turns out that most of the variability in the training data can be captured by a smaller number of eigenvoices, MAP speaker adaptation can use the remaining eigenvoices to good advantage since it implicitly takes account of the corresponding eigenvalues. A clear example of this is given in [14, Table I] which reports the results of some closed-set speaker-identification experiments on a population of 319 speakers. Decreasing the number of eigenvoices from 300 to 100 was found to increase the error rate from 14.8% to 16.5% in the case of mean adaptation and from 14.7% to 17.7% when the variances were adapted as well. The need to use as many eigenvoices as possible is particularly evident in the case of large vocabulary speech recognition given the very high dimensionality of the supervector space. The experiments we report in this article were conducted on a large vocabulary task using as many eigenvoices in each stream as there are speakers in the training set.

We begin by explaining why cluster adaptive training [9] and the maximum likelihood eigenspace method [13] would break down if we attempted to use them to estimate a full complement of eigenvoices in situations where speaker-dependent training is not feasible and how this problem can be avoided by adopting the probabilistic approach.

II. MAXIMUM LIKELIHOOD FORMULATIONS OF THE ESTIMATION PROBLEM

Let $M(s)$ denote the supervector for a speaker s. Our assumption is that for a randomly chosen speaker s, $M(s)$ is Gaussian distributed with mean M_0 and covariance matrix B. The speaker-independent supervector M_0 can be estimated by Baum-Welch training in the usual way so the problem confronting us is how to estimate B if speaker-dependent training is not feasible (so that the supervectors for the training speakers are unobservable). This problem is complicated by the fact that the dimensions of B are enormous but it is amenable to maximum likelihood estimation because the maximum likelihood estimate of B is of low rank in practice (the rank is bounded by the number of training speakers). This fact will allow us to work out an exact solution to the estimation problem (without having to make any approximations as in [11], [15]).

Previous approaches to the problem of estimating eigenvoices have adopted a different perspective: instead of estimating B, they estimate a basis for the eigenspace. This point of view is equivalent to ours in the sense that given an estimate of B we can write down a basis for the eigenspace (namely, the eigenvectors of B corresponding to nonzero eigenvalues) and, conversely, given a basis for the eigenspace we can estimate B (by fitting a multivariate Gaussian distribution to the coordinate representations of the supervectors). However the two points of view give rise to different maximum likelihood formulations of the estimation problem and hence to different estimation procedures.

To describe the likelihood function used to estimate the eigenspace basis in [9], [13] we need to introduce some notation. For each mixture component c, let $M_c(s)$ be the subvector of $M(s)$ which corresponds to it. It is assumed that there is a covariance matrix Σ_c such that, for any speaker s, acoustic

observation vectors (frames) associated with the mixture component are normally distributed with mean $M_c(s)$ and covariance matrix Σ_c. Note that, although Σ_c is independent of s, it is *not* a speaker-independent covariance matrix in the usual sense since it measures deviations from the speaker-dependent mean vectors $M_c(s)$ rather than deviations from a speaker-independent mean vector. Let $\boldsymbol{\Sigma}$ denote the $CF \times CF$ block diagonal matrix whose diagonal blocks are $\Sigma_1, \ldots, \Sigma_C$. For each speaker s, let $\mathcal{X}(s)$ denote the speaker's training data and for each supervector $\boldsymbol{M}$, let $P_{\mathrm{HMM}}(\mathcal{X}(s)|\boldsymbol{M}, \boldsymbol{\Sigma})$ denote the likelihood of $\mathcal{X}(s)$ calculated with the HMM specified by the supervector $\boldsymbol{M}$ and the supercovariance matrix $\boldsymbol{\Sigma}$. (We assume throughout that the HMM transition probabilities and mixture weights are fixed so there is no need to include these in the notation.)

Although the "bias term" $\boldsymbol{M}_0$ can be eliminated by augmenting the dimension of the eigenspace [9], it will be convenient for us to retain it. Given a basis for the eigenspace, let $\boldsymbol{V}$ be the matrix whose columns are the basis supervectors so that every speaker supervector can be written in the from $\boldsymbol{M}_0 + \boldsymbol{V}\boldsymbol{y}$. (The vector $\boldsymbol{y}$ is of dimension $R \times 1$ where R is the dimension of the eigenspace.) The likelihood function which serves as the estimation criterion in [9], [13] is

$$\prod_s \max_{\boldsymbol{y}} P_{\mathrm{HMM}}(\mathcal{X}(s)|\boldsymbol{M}_0 + \boldsymbol{V}\boldsymbol{y}, \boldsymbol{\Sigma}) \qquad (1)$$

where s ranges over the speakers in the training set. (Strictly speaking this is not a likelihood function since it does not integrate to 1. In order to obtain a proper likelihood function the max operation would have to be replaced by a suitable integral with respect to $\boldsymbol{y}$.) The optimization proceeds by iterating the following two steps:

1) For each training speaker s, use the current estimates of $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ to find the supervector which maximizes the HMM likelihood of the speaker's training data $\mathcal{X}(s)$. Set

$$\boldsymbol{y}(s) = \arg\max_{\boldsymbol{y}} P_{\mathrm{HMM}}(\mathcal{X}(s)|\boldsymbol{M}_0 + \boldsymbol{V}\boldsymbol{y}, \boldsymbol{\Sigma}).$$

2) Update $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ by maximizing

$$\prod_s P_{\mathrm{HMM}}(\mathcal{X}(s)|\boldsymbol{M}_0 + \boldsymbol{V}\boldsymbol{y}(s), \boldsymbol{\Sigma})$$

where the product extends over all speakers in the training set.

These two steps are referred to as "maximum likelihood eigendecomposition" and the "maximum likelihood eigenspace" method in [13] and as "estimating the cluster weights" and "estimating the cluster means" in [9].

The methods in [9], [13] differ as to how the maximization in the second step is performed. In particular, it is assumed in [13] that, for each mixture component c, Σ_c is equal to the speaker-independent covariance matrix for the mixture component and no attempt is made to re-estimate it. It will be helpful to briefly review Gales's method [9] for estimating $\boldsymbol{V}$ since our approach will require a similar calculation. For each speaker s, let $\boldsymbol{M}(s) = \boldsymbol{M}_0 + \boldsymbol{V}\boldsymbol{y}(s)$ where $\boldsymbol{y}(s)$ is given by the first step and suppose that each frame in the training data $\mathcal{X}(s)$ has been aligned with a mixture component (as in a Viterbi alignment although a forward-backward alignment can also be used).

For each mixture component c, let $N_c(s)$ denote the number of frames aligned with c. Then the quantity to be maximized in the second step is

$$\sum_s \sum_c \left(N_c(s) \log \frac{1}{(2\pi)^{F/2}|\Sigma_c|^{1/2}} \right.$$
$$\left. -\frac{1}{2} \sum_t (X_t - M_c(s))^* \Sigma_c^{-1}(X_t - M_c(s)) \right)$$

where s ranges over all speakers in the training set, c ranges over all mixture components and for each pair (s, c) the sum over t extends over all frames X_t aligned with c. If we ignore the issue of how to estimate $\boldsymbol{\Sigma}$, the problem is just to minimize

$$\sum_s \sum_c \sum_t (X_t - M_c(s))^* \Sigma_c^{-1}(X_t - M_c(s))$$

regarded as a function of $\boldsymbol{V}$. It is characteristic of this type of optimization problem that the covariance matrices drop out so the problem reduces to an exercise in least squares, namely, to minimize

$$\sum_s \sum_c \sum_t (X_t - M_c(s))^*(X_t - M_c(s)).$$

For our purposes, the important thing to note is that linear regression (with the $\boldsymbol{y}(s)$'s as the explanatory variables and the columns of $\boldsymbol{V}$ as the regression coefficients) provides the solution.

Our objection to using the likelihood function (1) as the estimation criterion is that, in order to obtain a reasonable estimate of the eigenspace basis in situations where speaker-dependent training is not feasible, the dimension of the eigenspace has to be strictly less than the number of training speakers. (If not, the maximum value of (1) is obviously attained by the basis which consists of the estimates of the training speakers' supervectors given by speaker-dependent training.) So in order to avoid this type of degeneracy, only a relatively small number of eigenvoices can be estimated or the basis vectors have to be constrained in some other way. (For example, the basis vectors can be constrained to lie in the low dimensional space consisting of all MLLR transforms of $\boldsymbol{M}_0$ [9].) These constraints are unreasonable because they are artifacts of the estimation procedure.

In order to outline the alternative estimation procedure that we will develop, note first that, if R is the rank of $\boldsymbol{B}$, we can write

$$\boldsymbol{B} = \boldsymbol{V}\boldsymbol{V}^*$$

where $\boldsymbol{V}$ is a $CF \times R$ matrix of rank R. This implies that for each speaker s, there is a unique $R \times 1$ vector $\boldsymbol{y}(s)$ such that

$$\boldsymbol{M}(s) = \boldsymbol{M}_0 + \boldsymbol{V}\boldsymbol{y}(s).$$

To say that $\boldsymbol{M}(s)$ is normally distributed with mean $\boldsymbol{M}_0$ and covariance matrix $\boldsymbol{B}$, is equivalent to saying $\boldsymbol{y}(s)$ has a standard normal distribution. So the problem of estimating the supervector covariance matrix can be formulated in terms of estimating $\boldsymbol{V}$ using a different generative model for the training data.

Fix a speaker s and suppose that each frame in the training data $\mathcal{X}(s)$ has been aligned with a mixture component. If the

vector $\boldsymbol{y}(s)$ were observable then, using the notation introduced above, the log likelihood of $\mathcal{X}(s)$ would given by

$$\sum_c \left(N_c(s) \log \frac{1}{(2\pi)^{F/2}|\Sigma_c|^{1/2}} \right.$$
$$\left. -\frac{1}{2} \sum_t (X_t - M_c(s))^* \Sigma_c^{-1} (X_t - M_c(s)) \right).$$

Since we are treating $\boldsymbol{y}(s)$ as a random vector with a standard normal distribution, we define the likelihood of $\mathcal{X}(s)$ by integrating this conditional likelihood function with respect to the standard Gaussian kernel $N(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{I})$. We denote the value of this integral by $P_{\boldsymbol{V}, \boldsymbol{\Sigma}}(\mathcal{X}(s))$ (a closed form expression is given in Proposition 2 below). Thus our approach will be to estimate $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ by maximizing the log likelihood function

$$\prod_s P_{\boldsymbol{V}, \boldsymbol{\Sigma}}(\mathcal{X}(s))$$

where the product extends over all speakers in the training set.

We will develop an EM algorithm for this purpose in which the role of the hidden variables is played by the vectors $\boldsymbol{y}(s)$. We will show that successive estimates of $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ are convergent, at least to the extent that the likelihood of the training data increases on successive iterations. However, as in [12], the EM algorithm generally has more than one fixed point so there is no guarantee that if $(\boldsymbol{V}, \boldsymbol{\Sigma})$ is an EM fixed point then the eigenvectors of $\boldsymbol{V}\boldsymbol{V}^*$ are the R principal eigenvectors of the supervector covariance matrix.

Our procedure consists in iterating the following three steps. We will spell out the details in Section III.

1) For each training speaker s, use the current alignment of the speaker's training data and the current estimates of $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ to carry out MAP speaker adaptation. Use the speaker-adapted model to realign the speaker's training data.

2) The E-step: For each speaker s, calculate the posterior distribution of $\boldsymbol{y}(s)$ using the current alignment of the speaker's training data, the current estimates of $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ and the prior $N(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{I})$.

3) The M-step: Update $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ by a linear regression in which the $\boldsymbol{y}(s)$'s play the role of the explanatory variables.

Calculating the posterior distribution of $\boldsymbol{y}(s)$ in the E-step rather than the maximum likelihood estimate is the key to avoiding the degeneracy problem that arises when (1) is used as the criterion for estimating the eigenvoices in situations where speaker-dependent training is not feasible and the number of eigenvoices is large compared with the number of training speakers. This calculation is also essentially all that is required for the first step. The principal mathematical difficulty that we will encounter is in carrying out the regression in the M-step, given that for each speaker s, $\boldsymbol{y}(s)$ is only observable up to the posterior distribution calculated in the E-step.

III. ADAPTATION AND ESTIMATION PROCEDURES

The main computation that needs to be done both for estimating $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ and for MAP speaker adaptation for a speaker

s is to calculate the posterior distribution of $\boldsymbol{y}(s)$ given the speaker's training data. To explain how this is done we need to introduce some notation. Assuming that the data for the speaker has been aligned with either the speaker-independent or a speaker-adapted HMM, so that each frame is labeled by a mixture component, we denote by $\mathcal{X}(s)$ the entire collection of labeled frames for the speaker. We extract the following statistics from $\mathcal{X}(s)$. For each mixture component $c = 1, \ldots, C$, let $N_c(s)$ be the number of frames in the training data for speaker s which are accounted for by the given mixture component and set

$$S_{X,c}(s) = \sum_t (X_t - \mu_c)$$
$$S_{XX^*,c}(s) = \sum_t (X_t - \mu_c)(X_t - \mu_c)^*$$

where the sums extend over all frames X_t for speaker s that are aligned with the mixture component c and μ_c is the speaker-independent mean vector.

Let $\boldsymbol{N}(s)$ be the $CF \times CF$ block diagonal matrix whose diagonal blocks are $N_1(s)I, \ldots, N_C(s)I$ where I denotes the $F \times F$ identity matrix. Let $\boldsymbol{S}_X(s)$ be the $CF \times 1$ vector obtained by concatenating $S_{X,1}(s), \ldots, S_{X,C}(s)$. Let $\boldsymbol{l}(s)$ be the $R \times R$ matrix defined by

$$\boldsymbol{l}(s) = \boldsymbol{I} + \boldsymbol{V}^* \boldsymbol{\Sigma}^{-1} \boldsymbol{N}(s) \boldsymbol{V}.$$

Proposition 1: For each training or test speaker s, the posterior distribution of $\boldsymbol{y}(s)$ given $\mathcal{X}(s)$ and a parameter set $(\boldsymbol{V}, \boldsymbol{\Sigma})$ is Gaussian with mean

$$\boldsymbol{l}^{-1}(s) \boldsymbol{V}^* \boldsymbol{\Sigma}^{-1} \boldsymbol{S}_X(s)$$

and covariance matrix $\boldsymbol{l}^{-1}(s)$.

The proof of this proposition can be found in the Appendix. Note that $\boldsymbol{l}(s)$ is strictly positive definite (and hence, invertible). Furthermore, if R is reasonably small (at most a few thousand) there is no difficulty in calculating the inverse of $\boldsymbol{l}(s)$. Hence calculating the posterior distribution is straightforward even in the large vocabulary case.

Corollary 1: If, for each training or test speaker s, we denote the posterior mean and covariance of $\boldsymbol{M}(s)$ by $\hat{\boldsymbol{M}}(s)$ and $\hat{\boldsymbol{B}}(s)$, then

$$\hat{\boldsymbol{M}}(s) = \boldsymbol{M}_0 + \boldsymbol{V} \boldsymbol{l}^{-1}(s) \boldsymbol{V}^* \boldsymbol{\Sigma}^{-1} \boldsymbol{S}_X(s)$$
$$\hat{\boldsymbol{B}}(s) = \boldsymbol{V} \boldsymbol{l}^{-1}(s) \boldsymbol{V}^*.$$

This corollary is the key to MAP speaker adaptation. Invoking the Bayesian predictive classification principle [16], we can adapt both the mean vectors and the variances in the speaker independent HMM to a given speaker s as follows. With each mixture component c, we associate the mean vector $\hat{M}_c(s)$ and a covariance matrix given by

$$\Sigma_c + \hat{B}_{cc}(s) \tag{2}$$

where $\hat{B}_{cc}(s)$ is the ccth entry of $\hat{\boldsymbol{B}}(s)$ when $\hat{\boldsymbol{B}}(s)$ is considered as a $C \times C$ block matrix (each block being of dimension $F \times F$). This type of covariance estimation seems to have been

first suggested in [11]. The rationale behind it is that if X is an observation for speaker s and mixture component c, we have

$$X = M_c(s) + E$$

where the residual E is distributed with mean 0 and covariance matrix Σ_c and $M_c(s)$ is distributed with mean $\hat{M}_c(s)$ and co-variance matrix $\hat{B}_{cc}(s)$. Since E and $M_c(s)$ are assumed to be statistically independent

$$\begin{aligned} \text{Cov}(X,X) &= \text{Cov}(M_c(s), M_c(s)) + \text{Cov}(E,E) \\ &= \Sigma_c + \hat{B}_{cc}(s) \end{aligned}$$

which gives (2). We will refer to this as covariance adaptation although this is not strictly correct since it is really a mechanism for incorporating the uncertainty about the MAP estimate of $M(s)$ into the HMM. (Note that as the amount of adaptation data increases, the uncertainty tends to zero and the covariance estimate reduces to Σ_c for all speakers.)

Corollary 2: If, for each training speaker s, $E[\boldsymbol{y}(s)]$ denotes the posterior expectation of $\boldsymbol{y}(s)$ given $\mathcal{X}(s)$ and a parameter set $(\boldsymbol{V}, \boldsymbol{\Sigma})$ and likewise for $E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)]$ then

$$\begin{aligned} E[\boldsymbol{y}(s)] &= \boldsymbol{l}^{-1}(s)\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s) \\ E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)] &= E[\boldsymbol{y}(s)]E[\boldsymbol{y}^*(s)] + \boldsymbol{l}^{-1}(s). \end{aligned}$$

This also follows immediately from Proposition 1 and it is the key to implementing the E-step of the EM algorithm described in Proposition 3 below.

Next we explain how to calculate the likelihood function $P_{\boldsymbol{V},\boldsymbol{\Sigma}}$ that we used in our formulation of the estimation problem. (This calculation is not strictly necessary; its only role is to provide a diagnostic for verifying the implementation of the EM algorithm.)

Proposition 2: For each speaker s, let $G_{\boldsymbol{\Sigma}}(s)$ denote the Gaussian log likelihood function given by the expression

$$\sum_{c=1}^{C} \left(N_c(s) \log \frac{1}{(2\pi)^{F/2}|\Sigma_c|^{1/2}} - \frac{1}{2}\text{tr}\left(\Sigma_c^{-1}S_{XX^*,c}(s)\right) \right). \tag{3}$$

Then

$$\log P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}(s)) = G_{\boldsymbol{\Sigma}}(s) - \frac{1}{2}\log|\boldsymbol{l}(s)| \\ + \frac{1}{2}(\hat{\boldsymbol{M}}(s) - \boldsymbol{M}_0)^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s).$$

Finally, the EM algorithm:

Proposition 3: Suppose we are given initial parameter estimates $(\boldsymbol{V}_0, \boldsymbol{\Sigma}_0)$. For each training speaker s, let $E[\boldsymbol{y}(s)]$ and $E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)]$ be the first and second moments of $\boldsymbol{y}(s)$ calculated with these estimates according to Corollary 2 to Proposition 1. Let $(\boldsymbol{V}, \boldsymbol{\Sigma})$ be new estimates of the model parameters defined as follows: $\boldsymbol{V}$ is the solution of

$$\sum_{s} \boldsymbol{N}(s)\boldsymbol{V}E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)] = \sum_{s} \boldsymbol{S}_X(s)E[\boldsymbol{y}^*(s)] \tag{4}$$

and for each $c = 1, \ldots, C$

$$\Sigma_c = \frac{1}{n_c}\left(\sum_{s} S_{XX^*,c}(s) - M_c \right) \tag{5}$$

where $n_c = \sum_s N_c(s)$ and M_c denotes the cth diagonal block of the $CF \times CF$ matrix

$$\frac{1}{2}\sum_{s} \left(\boldsymbol{S}_X(s)E[\boldsymbol{y}^*(s)]\boldsymbol{V}^* + \boldsymbol{V}E[\boldsymbol{y}(s)]\boldsymbol{S}_X^*(s) \right).$$

Then

$$\sum_{s} \log P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}(s)) \geq \sum_{s} \log P_{\boldsymbol{V}_0,\boldsymbol{\Sigma}_0}(\mathcal{X}(s))$$

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

Equation (4) is the system of normal equations for our linear regression problem. To solve it, note that since the matrices $\boldsymbol{N}(s)$ are diagonal, equating the ith row of the left hand side with the ith row of the right hand side for $i = 1, \ldots, CF$ gives

$$\boldsymbol{V}^i \sum_{s} \boldsymbol{N}^i(s)E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)] = \sum_{s} \boldsymbol{S}_X^i(s)E[\boldsymbol{y}^*(s)]$$

where $\boldsymbol{V}^i$ is the ith row of $\boldsymbol{V}$ and similarly for $\boldsymbol{N}^i(s)$ and $\boldsymbol{S}_X^i(s)$. Note also that if we write i in the form $(c-1)F + f$ where $1 \leq c \leq C$ and $1 \leq f \leq F$ then $\boldsymbol{N}^i(s) = N_c(s)$ so we obtain

$$\boldsymbol{V}^i \sum_{s} N_c(s)E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)] = \sum_{s} \boldsymbol{S}_X^i(s)E[\boldsymbol{y}^*(s)].$$

This is just an $R \times R$ system of equations so there is no difficulty in solving for $\boldsymbol{V}^i$ and in the limit as the amount of training data tends to infinity (so that, for each speaker s, the posterior distribution of $\boldsymbol{y}(s)$ becomes concentrated at a single point, namely the maximum likelihood estimate of $\boldsymbol{y}(s)$)) it is equivalent to [9, eq. (49)]. Equation (5) stands in the same relation to [9, eq. (51)] so Gales's method and ours are asymptotically equivalent.

Note that the EM algorithm does not impose any restrictions on R which seems to suggest that it is possible to estimate a larger number of eigenvoices than there are speakers in the training set. This paradox can be resolved by proving that if $(\boldsymbol{V}, \boldsymbol{\Sigma})$ is a fixed point of the EM algorithm then R (that is, the rank of $\boldsymbol{V}$) is necessarily less than or equal S, the number of speakers in the training set. It follows that there is nothing to be gained by taking $R > S$ so we took $R = S$ in our experiments.

IV. IMPLEMENTATION ISSUES

In order to apply the EM estimation procedure given in Proposition 3 we need the first and second order statistics for each training speaker and mixture component. These statistics can be extracted by aligning the training data with the speaker-independent HMM using either a Viterbi alignment or the Baum–Welch algorithm. (We used the Baum–Welch procedure in our experiments.) But note that after estimates of $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ have been obtained we can use MAP adaptation (i.e., Corollary 1 to Proposition 1) to construct speaker-adapted models for each of the training speakers and use these to align the training data instead of the speaker-independent model. Accordingly, in the training phase of our experiments, we alternate between alignment and EM iterations until the estimates of $(\boldsymbol{V}, \boldsymbol{\Sigma})$ converge. (A similar issue arises when applying MAP adaptation to a test speaker. We deal with it in the same way by performing several alignment iterations on the speaker's adaptation data.)

As we mentioned in the introduction, some constraints have to be imposed on the supervector covariance matrix in order to produce an eigenspace of sufficiently high dimension that MAP adaptation has some hope of being asymptotically equivalent to speaker-dependent training. Our experience has been that the type of block diagonal constraint used in [5] which we referred to in the introduction is not very effective so we experimented with another type of constraint which we implemented by splitting the acoustic features into streams and applying the EM algorithm in each stream separately. This has the effect of boosting the dimension of the eigenspace by a factor of N where N is the number of streams (so that the greatest effect is achieved by treating each acoustic feature as a separate stream).

As we mentioned in Section II, Nguyen and colleagues [13] construct speaker HMMs by applying eigenvoice adaptation to the HMM mean vectors and copying the HMM covariance matrices from a speaker-independent HMM trained in the usual way. Since this is easier to implement than our Bayesian version of Gales's treatment of the HMM covariances, we compared both approaches in all of our experiments. Contrary to our expectations we found that Nguyen's approach (which is really a type of variance inflation) almost always gives better results.

V. EXPERIMENTS

We carried out some experiments on the French language *AUPELF* task [17] using the BREF-80 training set which consists of 10.6 h of data collected from 80 speakers. The speaker-independent HMM was a tied-state triphone model with 782 output distributions each having four mixture components with diagonal covariance matrices. For signal processing we used a 10 ms frame rate and a 26-dimensional acoustic feature vector (13 lifted mel-frequency cepstral coefficients together with their first derivatives). For recognition we used a dictionary containing 20 000 words and a language model consisting of 311 000 bigrams and 80 000 trigrams. (Admittedly a much larger language model would be needed to deal with the problem of homophone confusions in French.) We modeled liaisons in both training and recognition [18].

For the test set we chose 20 speakers (not represented in the training set) and five sentences per speaker for a total of 1 435 words of which 3.0% were out of vocabulary.

A. Speaker Adaptation

We experimented with 26 streams (each of dimension 1) and two streams (each of dimension 13, one for cepstra, and the other for their first derivatives). We used the training set to estimate 80 eigenvoices in each stream and performed supervised adaptation with various adaptation sets comprising 1, 2, 5, 10, 15, 20, and 100 sentences per speaker; the average length of a sentence was 6 s.

1) 26 Streams: Recognition results averaged over the 20 test speakers are reported in Table I. The second column of Table I gives the results of adapting the HMM mean vectors and copying the HMM covariance matrices from the speaker-independent HMM. Adapting only the mean vectors with small amounts of adaptation data gave small improvements in accuracy and performance saturated slowly, only reaching a plateau

TABLE I
RECOGNITION ACCURACIES (%) AVERAGED OVER 20 SPEAKERS, 26 STREAMS. S IS THE NUMBER OF ADAPTATION SENTENCES, M INDICATES MEAN ADAPTATION, MV INDICATES MEAN AND VARIANCE ADAPTATIONS

S	M	MV
0	70.9	70.9
1	71.4	68.6
2	72.3	71.5
5	74.4	73.0
10	75.1	73.9
15	75.2	74.0
20	76.6	74.5
100	76.6	75.5

TABLE II
RECOGNITION ACCURACIES (%) AVERAGED OVER 20 SPEAKERS. TWO STREAMS. S IS THE NUMBER OF ADAPTATION SENTENCES, M INDICATES MEAN ADAPTATION, MV INDICATES MEAN AND VARIANCE ADAPTATION

S	M	MV
0	70.9	70.9
1	74.0	72.8
2	74.9	72.9
5	75.5	72.7
10	75.6	72.3
15	75.8	73.2
20	75.9	73.5
100	75.5	73.5

after 20 sentences (2 min of speech). The third column shows that adapting the variances as well as the mean vectors was unhelpful across the entire range of adaptation sets.

The only way we were able to achieve any improvement with variance adaptation was by imposing block diagonal constraints on B (compare [5]). However these constraints diminished the effectiveness of mean adaptation so that no net gain in performance was obtained. Thus, partitioning the mixture components into four blocks gave 73.4% accuracy for mean adaptation versus 74.0% for mean and variance adaptation (using 100 adaptation sentences for each speaker). Similarly, partitioning the mixture components into ten blocks gave 74.7% accuracy for mean adaptation versus 74.8% for mean and variance adaptation.

2) 2 Streams: Table II reports recognition results obtained under the same conditions as in Table I using two streams rather than 26.

Aside from the fact that variance adaptation is still ineffective, the main thing to note here is that the performance for mean adaptation saturates much more quickly but reaches a lower plateau than in Table I. This type of behavior is to be expected since the eigenspace is of much lower dimension (160 versus 2080). The substantial improvement obtained with a single sentence of adaptation data and a relatively small number of eigenvoices confirms Botterweck's result [16].

B. Multispeaker Modeling

Since variance adaptation with low dimensional eigenspaces was consistently unhelpful, we tried another way of producing speaker-adapted models for the test speakers, namely adding the adaptation data to the training data and estimating the model parameters (V, Σ) using the extended training set. With sufficient adaptation data for the test speakers, this increases the

number of eigenvoices that can be estimated (from 80 to 100 per stream in the case at hand) and so ensures that the eigenspace is big enough to contain the test speakers as well as the training speakers. We refer to this type of training as multispeaker modeling since it only produces speaker-adapted models for the speakers in the extended training set.

We performed a multispeaker recognition experiment using 100 adaptation sentences for each of the test speakers. Performing Baum–Welch estimation of the speaker independent HMM with the extended training set and running recognition on the test speakers gave a new benchmark recognition accuracy of 72.1%. Performing mean adaptation on all of the speakers in the extended training set in the course of estimating (V, Σ) and running recognition on the test speakers gave a recognition accuracy of 76.4%. Adapting both the means and the variances also led to a recognition accuracy of 76.4% so there is no degradation in performance in this case but no improvement either.

VI. Discussion

Eigenvoice methods of acoustic phonetic modeling were developed initially to tackle small vocabulary tasks where speaker-dependent training can be carried out for large numbers of speakers. In this article we have shown how to estimate the principal eigenvoices of a speaker population in situations where the training data is too sparse to permit speaker-dependent training by formulating the problem in terms of maximum likelihood estimation of the supervector covariance matrix used in EMAP speaker adaptation. Unlike other methods of eigenvoice estimation, this approach enables us to estimate as many eigenvoices from a given training set as there are training speakers.

Our results, like Botterweck's, show that eigenvoice MAP can yield substantial improvements in accuracy on a large vocabulary task with very small amounts of adaptation data (one or two sentences). However, it is doubtful that eigenvoice modeling can provide a complete solution to the problem of acoustic phonetic adaptation in the large vocabulary case because, although it seems reasonable to assume that the "true" supervector covariance matrix is of relatively low rank and our approach enables us to extract the largest possible number of eigenvoices from a given training set, it seems unlikely that sufficiently many training speakers could ever be enlisted to estimate the supervector covariance matrix reliably. (An exception is the case of multispeaker modeling where the training and test speaker populations coincide. This type of modeling is more likely to be of use in speaker recognition than in speech recognition [14]. It is also interesting to note that the eigenchannel MAP estimator introduced in [14] which uses the methods developed here to tackle the problem of blind channel compensation for speaker identification does not seem to suffer from this rank deficiency problem.)

If the supervector covariance matrix is *not* reliably estimated then speaker adaptation may saturate quickly but there is no guarantee that it will be asymptotically equivalent to speaker-dependent training. *Ad hoc* constraints designed to increase the dimension of the eigenspace (such as block diagonal constraints or statistical independence conditions on the acoustic features) result in better asymptotic behavior but the improvement is slight and the principal effect of such constraints seems to be to slow down the rate at which speaker adaptation saturates.

If the goal of speaker adaptation is to attain speaker-dependent performance with the smallest possible amount of adaptation data (rather than to attain the best possible performance with one or two sentences of adaptation data), then it may be necessary to adopt a different approach to estimating the supervector covariance matrix in the large vocabulary case. Instead of using a principal components analysis to estimate the principal eigenvectors of the supervector covariance matrix, we could carry out a factor analysis. That is, we could assume a decomposition of the form

$$B = D + VV^*$$

where D is a nonsingular diagonal covariance matrix (this guarantees that B is of full rank). Ideally this type of model will exhibit both the correct asymptotic behavior of classical MAP (the case $V = 0$) and the rapid saturation of eigenvoice MAP (the case $D = 0$).

A factor analysis of this type has been implemented in a connected digit recognition task (where speaker-dependent training is feasible) [8]. However, a factor analysis of the training data in the absence of speaker-dependent models will require a good deal of algorithmic development. (Note that even if speaker-dependent models are given, the natural procedure for estimating a factor loading matrix is an EM algorithm [19].) We will take up this question elsewhere.

Our results to the effect that variance adaptation is generally less effective than simply copying the speaker-independent HMM variances were contrary to our expectations. They seem to indicate that the best way to model variances is to inflate them and raise the question of whether heavy-tailed distributions ought to be used instead of Gaussians in HMM mixture modeling. Since replacing individual Gaussians by discrete scale Richter mixtures has been found to be a moderately effective strategy [20], continuous scale Richter mixtures might be worth exploring. (These have proved to be effective in dealing with the nonGaussian behavior of images containing edges as well as textures [21].) More ambitiously, one could replace each Gaussian in a HMM output distribution by an independent components analyzer, tying the mixing matrices in the spirit of [22]. (See [23] for an outstanding tutorial on ICA. Integrating ICA with HMMs is straightforward in principle [24].) This suggests that non-Gaussian modeling of speaker supervectors could also be considered but we do not have any evidence to indicate whether this is worth pursuing.

Appendix
Proofs of the Propositions

For each speaker s, let $P_{V,\Sigma}(\mathcal{X}(s)|y(s))$ denote the conditional likelihood of $\mathcal{X}(s)$ given $y(s)$ and the parameter set (V, Σ).

Lemma 1: For each speaker s

$$\log P_{V,\Sigma}(\mathcal{X}(s)|y(s)) = G_{\Sigma}(s) + H_{V,\Sigma}(s, y(s))$$

where $G_{\Sigma}(s)$ is defined by (3) and

$$H_{\boldsymbol{V},\boldsymbol{\Sigma}}(s,\boldsymbol{y}) = \boldsymbol{y}^*\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s) - \frac{1}{2}\boldsymbol{y}^*\boldsymbol{V}^*\boldsymbol{N}(s)\boldsymbol{\Sigma}^{-1}\boldsymbol{V}\boldsymbol{y}.$$

Proof: To simplify the notation, let us drop the reference to s. Let $\boldsymbol{O} = \boldsymbol{V}\boldsymbol{y}$. For each mixture component c, let O_c denote the cth block of $\boldsymbol{O}$ (where each block is an $F \times 1$ vector) and let

$$S_{XX^*,c}(O_c) = \sum_t (X_t - \mu_c - O_c)(X_t - \mu_c - O_c)^*$$

where the sum extends over all the frames X_t for the given speaker which are aligned with the mixture component c. The log likelihood of $\mathcal{X}$ conditioned on $\boldsymbol{y}$ is

$$\sum_c \left(N_c \log \frac{1}{(2\pi)^{F/2}|\Sigma_c|^{1/2}} - \frac{1}{2}\mathrm{tr}\left(\Sigma_c^{-1}S_{XX^*,c}(O_c)\right) \right) \tag{6}$$

where the sum extends over all mixture components. This can be simplified by writing

$$S_{XX^*,c}(O_c) = S_{XX^*,c} - S_{X,c}O_c^* - O_c S_{X,c}^* + N_c O_c O_c^*$$

so that

$$\mathrm{tr}\left(\Sigma_c^{-1}S_{XX^*,c}(O_c)\right) = \mathrm{tr}\left(\Sigma_c^{-1}S_{XX^*,c}\right)$$
$$-2S_{X,c}^*\Sigma_c^{-1}O_c + O_c^*\Sigma_c^{-1}N_c O_c$$

for $c = 1, \ldots, C$. This implies that

$$\sum_c \mathrm{tr}\left(\Sigma_c^{-1}S_{XX^*,c}(O_c)\right)$$
$$= \sum_c \mathrm{tr}\left(\Sigma_c^{-1}S_{XX^*,c}\right) - 2\boldsymbol{O}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X + \boldsymbol{O}^*\boldsymbol{N}\boldsymbol{\Sigma}^{-1}\boldsymbol{O}$$

and the result follows by substituting this expression in (6). ∎

a) Proof of Proposition 1: In order to show that the posterior distribution of $\boldsymbol{y}(s)$ is of the stated form it is enough to show that

$$P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\boldsymbol{y}|\mathcal{X}(s)) \propto \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{a}(s))^*\boldsymbol{l}(s)(\boldsymbol{y}-\boldsymbol{a}(s))\right)$$

where

$$\boldsymbol{a}(s) = \boldsymbol{l}^{-1}(s)\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s).$$

Dropping the reference to s, we have

$$P_{\boldsymbol{v},\boldsymbol{\Sigma}}(\boldsymbol{y}|\mathcal{X})$$
$$\propto P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}|\boldsymbol{y})N(\boldsymbol{y}|\boldsymbol{0},\boldsymbol{I})$$
$$\propto \exp\left(\boldsymbol{y}^*\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X - \frac{1}{2}\boldsymbol{y}^*\boldsymbol{V}^*\boldsymbol{N}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}\boldsymbol{y} - \frac{1}{2}\boldsymbol{y}^*\boldsymbol{y}\right)$$
$$= \exp\left(\boldsymbol{y}^*\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X - \frac{1}{2}\boldsymbol{y}^*\boldsymbol{l}\boldsymbol{y}\right)$$
$$\propto \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{a})^*\boldsymbol{l}(\boldsymbol{y}-\boldsymbol{a})\right)$$

as required. ∎

b) Proof of Proposition 2: If $N(\cdot|\boldsymbol{0},\boldsymbol{I})$ denotes the Gaussian kernel with mean $\boldsymbol{0}$ and covariance matrix $\boldsymbol{I}$, then

$$P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}(s)) = \int P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}(s)|\boldsymbol{y})N(\boldsymbol{y}|\boldsymbol{0},\boldsymbol{I})d\boldsymbol{y}.$$

By Lemma 1, we can write this as

$$\log P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}(s)) = G_{\Sigma}(s) + \log \int \exp\left(H_{\boldsymbol{V},\boldsymbol{\Sigma}}(s,\boldsymbol{y})\right)$$
$$\times N(\boldsymbol{y}|\boldsymbol{0},\boldsymbol{I})d\boldsymbol{y}$$

where

$$H_{\boldsymbol{V},\boldsymbol{\Sigma}}(s,\boldsymbol{y}) = \boldsymbol{y}^*\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s) - \frac{1}{2}\boldsymbol{y}^*\boldsymbol{V}^*\boldsymbol{N}(s)\boldsymbol{\Sigma}^{-1}\boldsymbol{V}\boldsymbol{y}.$$

If $N(\boldsymbol{y}|\boldsymbol{0},\boldsymbol{l}^{-1}(s))$ denotes the Gaussian kernel with mean $\boldsymbol{0}$ and covariance matrix $\boldsymbol{l}^{-1}(s)$ then the integral can be written in the form

$$\int \exp\left(\boldsymbol{y}^*\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s)\right) N(\boldsymbol{y}|\boldsymbol{0},\boldsymbol{l}^{-1}(s))d\boldsymbol{y}.$$

By the formula for the Fourier-Laplace transform of the Gaussian kernel [25] this simplifies to

$$\exp\left(-\frac{1}{2}|\boldsymbol{l}(s)| + \frac{1}{2}\boldsymbol{S}_X^*(s)\boldsymbol{\Sigma}^{-1}\boldsymbol{V}\boldsymbol{l}^{-1}(s)\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s)\right)$$

which, by corollary 1 to proposition 1, can be written as

$$\exp\left(-\frac{1}{2}|\boldsymbol{l}(s)| + \frac{1}{2}(\hat{\boldsymbol{M}}(s) - \boldsymbol{M}_0)^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s)\right).$$

so

$$\log P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}(s)) = G_{\Sigma}(s) - \frac{1}{2}|\boldsymbol{l}(s)|$$
$$+ \frac{1}{2}(\hat{\boldsymbol{M}}(s) - \boldsymbol{M}_0)^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s)$$

as required. ∎

It will be helpful to make a few remarks on differentiating functions of a matrix variable before turning to the proof of proposition 3. If $f(X)$ is a function of a matrix variable X, then for any matrix E of the same dimensions as X we set

$$D_E f(X) = \lim_{h\to 0} \frac{f(X+hE) - f(X)}{h}$$

and we refer to $D_E f(X)$ as the derivative of $f(X)$ in the direction of E. These directional derivatives are easily calculated for linear and quadratic functions of X. For the determinant function they are given by

$$D_E|X| = |X|\mathrm{tr}(X^{-1}E).$$

(This can be derived from the identity $XX^\dagger = |X|I$ where $X^\dagger$ is the adjoint of X.) If $D_E f(X) = 0$ for all matrices E then X is a critical point of $f(X)$.

Define an inner product $\langle \cdot, \cdot \rangle$ by setting

$$\langle X, Y \rangle = \mathrm{tr}\left(X^*Y\right).$$

Since, for each X, the functional $E \mapsto D_E f(X)$ is linear, there is a unique matrix $\nabla f(X)$ (the gradient of $f(X)$) such that

$$D_E f(X) = \langle \nabla f(X), E \rangle$$

for all E. So to find the critical points of $f(X)$ we first find the matrix-valued function $\nabla f(X)$ which enables us to express the directional derivatives of $f(X)$ in the form

$$D_E f(X) = \mathrm{tr}(\nabla f(X)^* E)$$

and then set $\nabla f(X) = 0$.

If the domain of $f(X)$ is restricted to *symmetric* matrices X (e.g., covariance matrices), the critical points of $f(X)$ are defined by the condition that $D_E f(X) = 0$ for all symmetric matrices E. In this case, the critical points can be found by setting

$$\frac{1}{2}(\nabla f(X) + \nabla f(X)^*) = 0.$$

c) Proof of Proposition 3: We begin by constructing an EM auxiliary function with the $\boldsymbol{y}$'s as hidden variables. By Jensen's inequality,

$$\sum_s \int \left(\log \frac{P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\boldsymbol{y},\mathcal{X}(s))}{P_{\boldsymbol{V}_0,\boldsymbol{\Sigma}_0}(\boldsymbol{y},\mathcal{X}(s))} \right) P_{\boldsymbol{V}_0,\boldsymbol{\Sigma}_0}(\boldsymbol{y}|\mathcal{X}(s)) d\boldsymbol{y}$$

$$\leq \sum_s \log \int \frac{P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\boldsymbol{y},\mathcal{X}(s))}{P_{\boldsymbol{V}_0,\boldsymbol{\Sigma}_0}(\boldsymbol{y},\mathcal{X}(s))} P_{\boldsymbol{V}_0,\boldsymbol{\Sigma}_0}(\boldsymbol{y}|\mathcal{X}(s)) d\boldsymbol{y}.$$

Since the right hand side of this inequality simplifies to

$$\sum_s \log P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}(s)) - \sum_s \log P_{\boldsymbol{V}_0,\boldsymbol{\Sigma}_0}(\mathcal{X}(s)),$$

the total log likelihood of the training data can be increased by choosing the new estimates of $(\boldsymbol{V},\boldsymbol{\Sigma})$ so as to maximize the left hand side. Since for each speaker s

$$P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\boldsymbol{y},\mathcal{X}(s)) = P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}(s)|\boldsymbol{Y})N(\boldsymbol{y}|\boldsymbol{0},\boldsymbol{I})$$

and similarly for $P_{\boldsymbol{V}_0,\boldsymbol{\Sigma}_0}(\boldsymbol{y},\mathcal{X}(s))$, this is equivalent to optimizing the quantity

$$\sum_s \int \log P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}(s)|\boldsymbol{y})P_{\boldsymbol{V}_0,\boldsymbol{\Sigma}_0}(\boldsymbol{y}|\mathcal{X}(s)) d\boldsymbol{y}$$

which we refer to as the auxiliary function and denote by $\mathcal{A}$.

Observe that

$$\mathcal{A} = \sum_s E\left[\log P_{\boldsymbol{V},\boldsymbol{\Sigma}}(\mathcal{X}(s)|\boldsymbol{y}(s))\right]$$

where $E[\cdot]$ is the conditional expectation operator introduced in the statement of the proposition. Furthermore, by Lemma 1

$$\mathcal{A} = \sum_s G_{\boldsymbol{\Sigma}}(s) + \sum_s E\left[H_{\boldsymbol{V},\boldsymbol{\Sigma}}(s,\boldsymbol{y}(s))\right]$$

and the second term on the right hand side can be simplified as follows:

$$\sum_s E\left[H_{\boldsymbol{V},\boldsymbol{\Sigma}}(s,\boldsymbol{y}(s))\right]$$

$$= \sum_s E\left[\boldsymbol{y}^*(s)\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s)\right.$$

$$\left. - \frac{1}{2}\boldsymbol{y}^*(s)\boldsymbol{V}^*\boldsymbol{N}(s)\boldsymbol{\Sigma}^{-1}\boldsymbol{V}\boldsymbol{y}(s)\right]$$

$$= \sum_s \left(E[\boldsymbol{y}^*(s)]\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s)\right.$$

$$\left. - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{V}^*\boldsymbol{N}(s)\boldsymbol{\Sigma}^{-1}\boldsymbol{V}E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)]\right)\right)$$

$$= \sum_s \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{S}_X(s)E[\boldsymbol{y}^*(s)]\boldsymbol{V}^*\right.\right.$$

$$\left.\left. - \frac{1}{2}\boldsymbol{N}(s)\boldsymbol{V}E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)]\boldsymbol{V}^*\right)\right).$$

We derive the re-estimation formula (4) by differentiating this expression with respect to $\boldsymbol{V}$ and setting the gradient to $\boldsymbol{0}$. If $\boldsymbol{W}$ is a matrix of the same dimensions as $\boldsymbol{V}$ then the derivative with respect to $\boldsymbol{V}$ in the direction of $\boldsymbol{W}$ is given by

$$\sum_s \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{S}_X(s)E[\boldsymbol{y}^*(s)] - \boldsymbol{N}(s)\boldsymbol{V}E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)]\right)\boldsymbol{W}^*\right).$$

In order for this to be equal to $\boldsymbol{0}$ for all $\boldsymbol{W}$ we must have

$$\sum_s \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{S}_X(s)E[\boldsymbol{y}^*(s)] - \boldsymbol{N}(s)\boldsymbol{V}E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)]\right) = \boldsymbol{0}$$

from which (4) follows.

It remains to derive the re-estimation formula (5) for $\boldsymbol{\Sigma}$. Let $\boldsymbol{F}$ be any block diagonal matrix having the same dimensions as $\boldsymbol{\Sigma}$, so that it has the form

$$\begin{pmatrix} F_1 & & \\ & \ddots & \\ & & F_C \end{pmatrix}.$$

Note that, for each c, the derivative of $\log\left|\Sigma_c^{-1}\right|$ with respect to Σ_c^{-1} in the direction of F_c is $\mathrm{tr}(\Sigma_c F_c)$. Hence the directional derivative of $\mathcal{A}$ with respect to $\boldsymbol{\Sigma}^{-1}$ in the direction of $\boldsymbol{F}$ is

$$\frac{1}{2}\sum_s \sum_c \left(N_c(s)\mathrm{tr}(\Sigma_c F_c) - \mathrm{tr}(F_c S_{XX^*,c}(s))\right)$$

$$+ \sum_s \mathrm{tr}\left(\boldsymbol{F}\left(\boldsymbol{S}_X(s)E[\boldsymbol{y}^*(s)]\boldsymbol{V}^*\right.\right.$$

$$\left.\left. - \frac{1}{2}\boldsymbol{N}(s)\boldsymbol{V}E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)]\boldsymbol{V}^*\right)\right)$$

which, by (4), simplifies to

$$\frac{1}{2}\sum_s \sum_c \left(N_c(s)\mathrm{tr}(\Sigma_c F_c) - \mathrm{tr}(F_c S_{XX^*,c}(s))\right)$$

$$+ \frac{1}{2}\sum_s \mathrm{tr}\left(\boldsymbol{F}\boldsymbol{S}_X(s)E[\boldsymbol{y}^*(s)]\boldsymbol{V}^*\right).$$

This can be written as

$$\frac{1}{2} \sum_c \mathrm{tr}\left(F_c \sum_s \left(N_c(s)\Sigma_c - S_{XX^*,c}(s) + \beta_c(s) \right) \right)$$

where, for each mixture component c and speaker s, $\beta_c(s)$ is the cth diagonal block of the $CF \times CF$ matrix $\boldsymbol{S}_X(s)E\left[\boldsymbol{y}^*(s)\right]\boldsymbol{V}^*$. The latter expression evaluates to 0 for all symmetric matrices $\boldsymbol{F}$ iff

$$\sum_s \left(N_c(s)\Sigma_c - S_{XX^*,c}(s) + \beta_c(s) \right)$$
$$+ \sum_s \left(N_c(s)\Sigma_c - S_{XX^*,c}(s) + \beta_c(s) \right)^* = 0$$

for $c = 1, \ldots, C$. Since Σ_c and $S_{XX^*,c}(s)$ are symmetric for all c and s, (5) follows immediately. (The estimates for the covariance matrices can be shown to be positive definite with a little bit of extra work.) ∎

REFERENCES

[1] R. Kuhn *et al.*, "Eigenfaces and eigenvoices: Dimensionality reduction for specialized pattern recognition," in *Proc. IEEE Workshop Multimedia Signal Processing*, Dec. 1998.

[2] ——, "Eigenvoices for speaker adaptation," in *Proc. ICSLP*, Sydney, Australia, Dec. 1998.

[3] ——, "Fast speaker adaptation using a priori knowledge," in *Proc. ICASSP*, Phoenix, AZ, Mar. 1999.

[4] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 695–707, Nov. 2000.

[5] G. Zavaliagkos, R. Schwartz, and J. Makhoul, "Batch, incremental and instantaneous adaptation techniques for speech recognition," in *Proc. ICASSP*, Detroit, MI, May 1995, pp. 676–679.

[6] H. Botterweck, "Anisotropic MAP defined by eigenvoices for large vocabulary continuous speech recognition," in *Proc. ICASSP*, Salt Lake City, UT, May 2001.

[7] E. Jon, D. Kim, and N. Kim, "EMAP-based speaker adaptation with robust correlation estimation," in *Proc. ICASSP*, Salt Lake City, UT, May 20 001.

[8] D. Kim and N. Kim, "Online adaptation of continuous density hidden Markov models based on speaker space model evolution," in *Proc. ICSLP*, Denver, CO, Sep. 2002.

[9] M. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 417–428, Jul. 2000.

[10] R. Westwood, "Speaker Adaptation Using Eigenvoices," M.Phil., Cambridge University, Cambridge, UK, 1999.

[11] B. Shahshahani, "A Markov random field approach to Bayesian speaker adaptation," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 2, pp. 183–191, Mar. 1997.

[12] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, pp. 435–474, 1999.

[13] P. Nguyen, C. Wellekens, and J.-C. Junqua, "Maximum likelihood eigenspace and MLLR for speech recognition in noisy environments," in *Proc. Eurospeech*, Budapest, Hungary, Sep. 1999.

[14] P. Kenny, M. Mihoubi, and P. Dumouchel, "New MAP estimators for speaker recognition," in *Proc. Eurospeech*, Geneva, Switzerland, Sep. 2003.

[15] Q. Huo and C.-H. Lee, "Online adaptive learning of the correlated continuous-density hidden Markov models for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 6, no. 4, pp. 386–397, Jul. 1998.

[16] C.-H. Lee and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," *Proc. IEEE*, vol. 88, pp. 1241–1268, Aug. 2000.

[17] J. Dolmazon *et al.*, "ARC B1—Organization de la première campagne AUPELF pour l'évaluation des systèmes de dictée vocale," in *JST97 FRANCIL*, Avignon, France, Apr. 1997.

[18] G. Boulianne, J. Brousseau, P. Ouellet, and P. Dumouchel, "French large vocabulary recognition with cross-word phonology transducers," in *Proc. ICASSP*, Istanbul, Turkey, Jun. 2000.

[19] D. Rubin and D. Thayer, "EM algorithms for ML factor analysis," *Psychometrika*, vol. 47, pp. 69–76, 1982.

[20] M. Gales and P. Olsen, "Tail distribution modeling using the richter and power exponential distributions," in *Proc. Eurospeech*, Budapest, Hungary, Sep. 1999.

[21] M. Wainwright, E. P. Simoncelli, and A. Willsky, "Random cascades on wavelet trees and their use in modeling and analyzing natural imagery," *Applied Computational Harmonic Analysis*, vol. 11, pp. 89–123, Jul. 2001.

[22] M. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 272–281, 1999.

[23] A. Hyvärinen and E. Oja. (1999) Independent Component Analysis: A Tutorial. [Online]. Available: http://www.cis.hut.fi/aapo/papers/IJCNN99_tutorialweb/

[24] W. Penny, R. Everson, and S. Roberts, "Hidden Markov independent components analysis," in *Advances in Independent Components Analysis*, M. Girolami, Ed.   New York: Springer-Verlag, 2000.

[25] H. Stark and J. Woods, *Probability, Random Processes and Estimation Theory for Engineers*.   Englewood Cliffs, NJ: Prentice-Hall, 1986.

Patrick Kenny (M'04) received the B.A. degree in mathematics from Trinity College, Dublin, Ireland, and the M.Sc. and Ph.D. degrees, also in mathematics, from McGill University.

He was Professor of electrical engineering at INRS-Télécommunication, Montréal, QC, Canada, from 1990 to 1995, where he started up Spoken Word Technologies, RIP to spin off INRS's speech recognition technology. He joined the Centre de Recherche Informatique de Montréal in 1998, where he now holds the position of principal research scientist. His current research interests are concentrated on Bayesian speaker- and channel-adaptation for speech and speaker recognition.

Gilles Boulianne (M'99) received the B.Sc. degree in unified engineering from the Université du Québec, Chicoutimi, QC, Canada, and the M.Sc. degree in telecommunications from INRS-Télécommunications, Montréal, QC. He worked on speech analysis and articulatory speech modeling at the Linguistics Department in the Université du Québec, Montréal, until 1990 and then on large vocabulary speech recognition at INRS and Spoken Word Technologies until 1998, when he joined the Centre de Recherche Informatique de Montréal. His research interests include finite state transducer approaches and practical applications of large vocabulary speech recognition.

Pierre Dumouchel (M'97) received the Ph.D. degree in telecommunications from INRS-Télécommunications, Montréal, QC, Canada, in 1995.

He is currently Vice-President R&D of the Centre de Recherche Informatique de Montréal, a Professor at the Ecole de Technologie Supérieure, and a board member of the Canadian Language Industry Association (AILIA). His research interests include broadcast news speech recognition, speaker recognition and audio-visual content extraction.