Odyssey 2010
The Speaker and Language Recognition Workshop
28 June – 1 July 2010, Brno, Czech Republic

Joint Factor Analysis for Speaker Recognition reinterpreted as Signal Coding using Overcomplete Dictionaries

Daniel Garcia-Romero and Carol Y. Espy-Wilson

Department of Electrical and Computer Engineering, University of Maryland, College Park, MD
dgromero@umd.edu, espy@umd.edu

Abstract

This paper presents a reinterpretation of Joint Factor Analysis as a signal approximation methodology—based on ridge regression—using an overcomplete dictionary learned from data. A non-probabilistic perspective of the three fundamental steps in the JFA paradigm based on point estimates is provided. That is, model training, hyperparameter estimation and scoring stages are equated to signal coding, dictionary learning and similarity computation respectively. Establishing a connection between these two well-researched areas opens the doors for cross-pollination between both fields. As an example of this, we propose two novel ideas that arise naturally form the non-probabilistic perspective and result in faster hyperparameter estimation and improved scoring. Specifically, the proposed technique for hyperparameter estimation avoids the need to use explicit matrix inversions in the M-step of the ML estimation. This allows the use of faster techniques such as Gauss-Seidel or Cholesky factorizations for the computation of the posterior means of the factors $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ during the E-step. Regarding the scoring, a similarity measure based on a normalized inner product is proposed and shown to outperform the state-of-the-art linear scoring approach commonly used in JFA. Experimental validation of these two novel techniques is presented using closed-set identification and speaker verification experiments over the Switchboard database.

1. Introduction

Joint factor analysis has become the state-of-the-art in speaker recognition systems [1]. Two major properties of this approach are responsible for this. The first one is the ability to obtain a fixed-length representation of a variable length object. That is, we are able to model a speech recording in terms of a fixed-length mean supervector**.** The second, and more important one, is that JFA provides a mechanism to explicitly model the undesired variability in the speech signal (i.e., intersession variability). Based on this, removing undesired components from our representation becomes much easier since they are explicitly captured.

It is well known that many problems involving linear models can be motivated from a probabilistic perspective as well as a deterministic one. For example, a linear curve fitting problem can be motivated based on maximum likelihood or a simple least squares. Both approaches have their strengths and weaknesses (see [2] for example). The JFA paradigm presents a probabilistic perspective around a linear-Gaussian model on speaker supervectors. The main goal of this paper is to provide a non-probabilistic view of the underlying process followed in JFA. The hope is that this alternative perspective will motivate new ways of thinking that result in algorithmic improvements.

2. Joint Factor Analysis overview

Since the introduction of JFA in [1] a great number of modifications have been proposed [3]. In order to remove any ambiguity about our particular choice of JFA variant, this section presents an overview of the three fundamental steps involved in the construction of a speaker recognition system: model training, hyperparameter estimation and score computation.

2.1. Paradigm

The Joint Factor Analysis paradigm [4] assumes that a sequence of T I.I.D. observed vectors, $\mathcal{O} = \{\mathbf{o}_t\}_{t=1}^{T}$ with $\mathbf{o}_t \in \mathbb{R}^F$, comes from a two-stage generative model. The first stage corresponds to a K-component Gaussian Mixture Model (GMM), $\boldsymbol{\lambda} = (\{w_k\}, \{\boldsymbol{\theta}_k\}, \{\boldsymbol{\Sigma}_k\})$, that is responsible for generating each observed vector $\mathbf{o}_t$:

$$p_{\boldsymbol{\lambda}}(\mathbf{o}_t|\{\boldsymbol{\theta}_k\}) = \sum_{k=1}^{K} w_k \frac{1}{(2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp\left\{\frac{1}{2}(\mathbf{o}_t - \boldsymbol{\theta}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{o}_t - \boldsymbol{\theta}_k)\right\}$$

$$\text{with } w_k \in \mathbb{R}, \boldsymbol{\theta}_k \in \mathbb{R}^F \text{ and } \boldsymbol{\Sigma}_k \in \mathbb{R}^{F\times F} \text{ for } k = 1, \dots, K \qquad (1)$$

The weights and covariance matrices of the GMM are considered fixed and known a priori. The means are assumed to be random vectors generated by the second stage of the generative model. In particular, a mean supervector $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T, \dots, \boldsymbol{\theta}_K^T]^T \in \mathbb{R}^{FK}$ is constructed by appending together the means of each mixture component and is assumed to obey an affine linear model (i.e., factor analysis model) of the form

$$\boldsymbol{\theta} = \mathbf{m} + \mathbf{U}\mathbf{x} + \mathbf{V}\mathbf{y} + \mathbf{D}\mathbf{z}, \qquad (2)$$

where the vector $\mathbf{m} \in \mathbb{R}^{FK}$ is a fixed offset, the matrices $\mathbf{V} \in \mathbb{R}^{FK\times P_v}$ and $\mathbf{U} \in \mathbb{R}^{FK\times P_u}$ correspond to factor loadings and the diagonal matrix $\mathbf{D} \in \mathbb{R}^{FK\times FK}$ is a scaling matrix. Moreover, the vectors $\mathbf{y} \in \mathbb{R}^{P_u}$ and $\mathbf{x} \in \mathbb{R}^{P_v}$ are considered as the common-factors and $\boldsymbol{z} \in \mathbb{R}^{FK}$ as the residual-factors. All three vectors, $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ are assumed independent of each other and distributed according to a standard Normal distribution of appropriate dimension. Consequently, equation (2) implies that the prior distribution of the mean supervector $\boldsymbol{\theta}$ is Gaussian with mean and covariance given by

$$\mathbb{E}[\boldsymbol{\theta}] = \mathbf{m} \quad \text{and} \quad \text{Cov}[\boldsymbol{\theta}] = \mathbf{U}\mathbf{U}^T + \mathbf{V}\mathbf{V}^T + \mathbf{D}\mathbf{D}^T. \qquad (3)$$

The rationale behind equation (2) is that, aside from the offset $\mathbf{m}$, the mean supervector is the superposition of three fundamental components with rather distinctive meanings. The component that lives in the $\text{span}(\mathbf{U})$ is used to denote the undesired variability contained in the observed vectors (e.g., convolutive or additive noise). Additionally, the $\text{span}(\mathbf{V})$ is where the basic constituting elements that capture the essence of the observed data live. Finally, the diagonal matrix $\mathbf{D}$ spans the entire ambiance space and provides a mechanism to

account for the residual variability not captured by the other two components.

In equation (1), the weights $\{w_k\}$ and covariance matrices $\{\boldsymbol{\Sigma}_k\}$ of the GMM λ are assumed to be fixed and known a priori. In practice, they are obtained from a previously trained GMM λ_{UBM} called Universal Background Model (UBM). This UBM must be trained using a large collection of data that is representative of the task at hand. Maximum Likelihood estimation is the most common approach [5].

2.2. Model training

Now that all the elements involved in the JFA model have been defined, we are in position to formulate the inference problem (i.e., model training). That is, given a sequence of observed vectors $\boldsymbol{\mathcal{O}} = \{\mathbf{o}_t\}_{t=1}^{T}$, we want to estimate the free parameters of the generating GMM that maximize the posterior distribution—which in this case are only the component means $\{\boldsymbol{\theta}_k\}$. We will also assume that the hyperparameters $\{\mathbf{m}, \mathbf{V}, \mathbf{U}, \mathbf{D}\}$ of the second stage of the generative process are also known (i.e., they have been obtained previously from a development data set via the ML approach described in next section). Thus, our optimization problem takes the form

$$\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta}|\boldsymbol{\mathcal{O}}) = \max_{\boldsymbol{\theta}} \; p_\lambda(\boldsymbol{\mathcal{O}}|\boldsymbol{\theta})\, p_\theta(\boldsymbol{\theta}). \tag{4}$$

In order to keep the formulas as clean as possible, we will refer to the entire collection of loading matrices by $\boldsymbol{\Phi} = [\mathbf{U}\,\mathbf{V}\,\mathbf{D}] \in \mathbb{R}^{FK\times P}$ and all the factors will be collected in $\boldsymbol{\beta} = [\mathbf{x}^T\mathbf{y}^T\mathbf{z}^T]^T \in \mathbb{R}^P$. Using this compact form, the mean supervector can also be expressed as

$$\boldsymbol{\theta} = \mathbf{m} + \boldsymbol{\Phi}\boldsymbol{\beta}. \tag{5}$$

Moreover, based on the prior distributions of the factors $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ as well as their independence, the vector $\boldsymbol{\beta}$ is distributed according to a standard Gaussian distribution. That is

$$p_\beta(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}; \mathbf{0}, \mathbf{I}). \tag{6}$$

Making use of equations (5) and (6) and substituting back into (4) an equivalent minimization problem can be obtained in terms of $\boldsymbol{\beta}$:

$$\min_{\boldsymbol{\beta}}\{-\log p_\lambda(\boldsymbol{\mathcal{O}}|\boldsymbol{\theta} = \mathbf{m} + \boldsymbol{\Phi}\boldsymbol{\beta}) - \log p_\beta(\boldsymbol{\beta})\}. \tag{7}$$

Once the optimal $\boldsymbol{\beta}_{MAP}$ is obtained, we can compute the optimal mean supervector $\boldsymbol{\theta}_{MAP}$ as:

$$\boldsymbol{\theta}_{MAP} = \mathbf{m} + \boldsymbol{\Phi}\boldsymbol{\beta}_{MAP}. \tag{8}$$

As usual, the analytical solution of this problem is not tractable and we use the EM algorithm to obtain a local optimizer. In the E-step we compute the occupations of the mixture component k for the observed vector $\mathbf{o}_t$ as

$$\gamma_{tk} = \frac{w_k\; \mathcal{N}\left(\mathbf{o}_t; \widehat{\boldsymbol{\theta}}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{k=1}^{K} w_k\; \mathcal{N}\left(\mathbf{o}_t; \widehat{\boldsymbol{\theta}}_k, \boldsymbol{\Sigma}_k\right)}, \tag{9}$$

where $\widehat{\boldsymbol{\theta}} = \left[\widehat{\boldsymbol{\theta}}_1^T, \ldots, \widehat{\boldsymbol{\theta}}_K^T\right]^T \in \mathbb{R}^{FK}$ is initialized with $\mathbf{m}$. Then, in the M-step we use the occupations $\{\gamma_{tk}\}$ to compute the complete-data log likelihood, that along with the prior for $\boldsymbol{\beta}$, allow us to obtain the easier to optimize surrogate objective

$$\Psi(\boldsymbol{\beta}) = \frac{1}{2}\sum_{k=1}^{K}\sum_{t=1}^{T}\gamma_{tk}\; (\mathbf{o}_t - \mathbf{m}_k - \boldsymbol{\Phi}_k\boldsymbol{\beta})^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{o}_t - \mathbf{m}_k - \boldsymbol{\Phi}_k\boldsymbol{\beta}) + \frac{1}{2}\,\boldsymbol{\beta}^T\boldsymbol{\beta}, \tag{10}$$

where $\mathbf{m}_k$ is the F-dimensional sub-vector of $\mathbf{m}$ indexed by the mixture component k. In order to obtain a complete vector-form expression for (10) without the summations, the following definitions are useful:

$$\gamma_k = \sum_{t=1}^{T}\gamma_{tk}\;,\;\; \boldsymbol{\Gamma}_k = \gamma_k\mathbf{I} \in \mathbb{R}^{F\times F} \text{ and } \boldsymbol{\Gamma} = \mathrm{diag}(\boldsymbol{\Gamma}_k) \in \mathbb{R}^{FK\times FK}. \tag{11}$$

The scalar γ_k represents how much of the observed data is accounted for by mixture k. The diagonal matrix $\boldsymbol{\Gamma}_k$ is an intermediate construct that replicates the scalar γ_k throughout F diagonal entries and the diagonal matrix $\boldsymbol{\Gamma}$—constructed using the $\mathrm{diag}(\cdot)$ operator—contains the K matrices $\boldsymbol{\Gamma}_k$ in its diagonal entries. Additionally, the following objects are also useful:

$$\boldsymbol{\mu}_k = \frac{1}{\gamma_k}\sum_{t=1}^{T}\gamma_{tk}\;\mathbf{o}_t\;,\;\; \boldsymbol{\mu} = [\boldsymbol{\mu}_1^T, \ldots, \boldsymbol{\mu}_K^T]^T \in \mathbb{R}^{FK} \text{ and } \boldsymbol{\eta} = \boldsymbol{\mu} - \mathbf{m}, \tag{12}$$

with $\boldsymbol{\mu}_k$ representing the weighted average of the observed data that is accounted for by the k^{th} mixture component. Taking equation (10), summing over the index t and using $\boldsymbol{\mu}_k$ from (12) we obtain

$$\Psi(\boldsymbol{\beta}) = \frac{1}{2}\sum_{k=1}^{K}\gamma_k\; (\boldsymbol{\mu}_k - \mathbf{m}_k - \boldsymbol{\Phi}_k\boldsymbol{\beta})^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k - \mathbf{m}_k - \boldsymbol{\Phi}_k\boldsymbol{\beta}) + \frac{1}{2}\,\boldsymbol{\beta}^T\boldsymbol{\beta}. \tag{13}$$

Finally, the summation over k can be taken care of—in an implicit way—by using the supervector notation:

$$\Psi(\boldsymbol{\beta}) = \frac{1}{2}(\boldsymbol{\eta} - \boldsymbol{\Phi}\boldsymbol{\beta})^T\boldsymbol{\Gamma}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\eta} - \boldsymbol{\Phi}\boldsymbol{\beta}) + \frac{1}{2}\,\boldsymbol{\beta}^T\boldsymbol{\beta}, \tag{14}$$

where the diagonal matrix $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\Sigma}_k) \in \mathbb{R}^{FK\times FK}$. Moreover, letting $\mathbf{W} = \boldsymbol{\Gamma}\boldsymbol{\Sigma}^{-1}$, we can obtain the alternative expression:

$$\Psi(\boldsymbol{\beta}) = \frac{1}{2}\left\|\mathbf{W}^{\frac{1}{2}}(\boldsymbol{\eta} - \boldsymbol{\Phi}\boldsymbol{\beta})\right\|_2^2 + \frac{1}{2}\,\|\boldsymbol{\beta}\|_2^2. \tag{15}$$

Noting that by construction $\mathbf{W}$ is diagonal positive semi-definite (or positive definite if all Gaussians are responsible for some data), it is easy to see that $\Psi(\boldsymbol{\beta})$ is strongly convex. Hence, computing the gradient and setting it to zero provides a necessary and sufficient condition for a unique global minimizer. Performing this operation we obtain a closed-form solution to problem (7):

$$\boldsymbol{\beta}_{MAP} = (\mathbf{I} + \boldsymbol{\Phi}^T\mathbf{W}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{W}\,\boldsymbol{\eta}. \tag{16}$$

2.3. Hyperparameters estimation

Since the JFA paradigm is only as good as its hypermeters[1], the estimation of the set $\{\mathbf{m}, \mathbf{V}, \mathbf{U}, \mathbf{D}\}$ has received a lot of attention. In particular, some of the variables being explored are: amount and type of data, number of dimensions of the subspaces, joint or independent estimation, generalization capabilities based on utterance duration and recording environments [6]. The most widespread criterion for the estimation process is the maximization of the likelihood function over a development data set [7].The EM algorithm is used to maximize the likelihood. The offset supervector $\mathbf{m}$ comes from the UBM model. Independent estimation of the

[1] Note that we are not including the covariance matrices $\{\boldsymbol{\Sigma}_k\}$ as part of the hyperparameters to emphasize the fact that we keep them fixed once computed in the UBM training process.

matrices $\mathbf{U},\mathbf{V}$ and $\mathbf{D}$ reduces the computational complexity greatly and provides state-of-the-art results [6]. Hence, that is the setup considered throughout this paper. In particular, given an initial guess $\mathbf{\Phi}_0$—which depending on the matrix being updated is identified with $\mathbf{U}_0, \mathbf{V}_0$ or $\mathbf{D}_0$—the E-step, for each data file r, produces the posterior means $\boldsymbol{\nu}_r = \boldsymbol{\beta}_r^{MAP}$ and correlation matrices $\mathbb{E}[\boldsymbol{\beta}_r \boldsymbol{\beta}_r^T] = (\mathbf{I} + \mathbf{\Phi}_0^T \mathbf{W}_r \mathbf{\Phi}_0)^{-1} + \boldsymbol{\nu}_r \boldsymbol{\nu}_r^T$. The M-step results in the update equation [7]:

$$\mathbf{\Phi}_{\text{new}}^{(k)} = \left(\sum_{r=1}^{R} \gamma_{rk} \ \boldsymbol{\eta}_r^{(k)} \boldsymbol{\beta}_r \right) \left(\sum_{r=1}^{R} \gamma_{rk} \ \mathbb{E}[\boldsymbol{\beta}_r \boldsymbol{\beta}_r^T] \right)^{-1}, \qquad (17)$$

where the super-index (k) indicates the F-dimensional subset of rows corresponding to the mixture k and the index r runs through the elements of the training data set. Thus, if JFA comprises a GMM with Kcomponents and $\mathbf{\Phi}_0 \in \mathbb{R}^{FK \times P}$, the updated $\mathbf{\Phi}_{\text{new}}$ requires the solution of K independent systems of P equations with F right-hand side elements.

2.4. Scoring

Once the hyperparameters and model training procedures are available, the only remaining component for a complete speaker recognition system is a similarity measure between models and test utterances. In [8] a comparison of scoring techniques ranging from a fully Bayesian approach to simple MAP point estimates was presented. The results indicated that—given enough data—a linear approximation of the log-likelihood results in a much faster computation of similarities without any significant loss in performance. Adapting their formulation to our notation, the speaker model is represented by $\hat{\boldsymbol{\eta}} = \mathbf{\Phi}\boldsymbol{\beta}_{MAP} - \mathbf{U}\mathbf{x}_{MAP}^{model}$ and the test utterance is summarized by its normalized, centered and session compensated first order sufficient statistics $\boldsymbol{\eta}_{test} - \mathbf{U}\mathbf{x}_{MAP}^{test}$. Recalling that $\mathbf{W}_{test} = \mathbf{\Gamma}_{test}\mathbf{\Sigma}^{-1}$, the final score is nothing more than the inner product

$$score = \hat{\boldsymbol{\eta}}^T \mathbf{W}_{test} (\boldsymbol{\eta}_{test} - \mathbf{U}\mathbf{x}_{MAP}^{test}) \qquad (18)$$

defined by the diagonal and positive definite matrix[1] $\mathbf{W}_{test}$.

3. JFA as Signal Coding using Overcomplete Dictionaries

In this section we present a reinterpretation of JFA as a signal approximation methodology—based on Ridge regression—using an overcomplete dictionary $\mathbf{\Phi}$ learned from data. With a simple change in perspective we will be able to abstract some of the unimportant details of JFA and bring to the foreground its essential principles. Moreover, establishing a connection between JFA and signal coding (SC) opens the doors for cross-pollination between fields (see [9] for a review of current trends in data-driven overcomplete dictionaries).

3.1. Signal Coding (SC)

We propose to deemphasize the two-stage generative model and focus on the EM part of the inference process. That is, to think of the E-step as a process that given a speech signal $\boldsymbol{\mathcal{O}} = \{\mathbf{o}_t\}_{t=1}^T$ with $\mathbf{o}_t \in \mathbb{R}^F$ and a K-mixture UBM $\lambda_{UBM} = (\{w_k\}, \{\mathbf{m}_k\}, \{\mathbf{\Sigma}_k\})$ produces a fixed-length target vector $\boldsymbol{\eta} \in \mathbb{R}^{FK}$ (see equation (12)) as well as a weighting diagonal matrix $\mathbf{W}$. Then, the M-step can be reinterpreted as a signal coding process—of the target vector $\boldsymbol{\eta}$—based on a weighted regularized linear regression approach. By looking at equation (15), we see that the objective function is comprised of two terms. The first one is a conventional weighted least squares loss; whereas the second is a penalty on the energy of the regression coefficients (i.e., ridge regularization term). These two terms represent a trade-off between the goodness-of-fit and the energy used to fit the target. The goal is to approximate the target vector $\boldsymbol{\eta}$—as well as possible—with a linear combination of the columns of $\mathbf{\Phi}$ while considering that there is a quadratic cost incurred by the amount of usage of each column. The diagonal weighting matrix $\mathbf{W}$ provides a mechanism to emphasize/deemphasize the relative importance of the coefficients of $\boldsymbol{\eta}$ in the approximation process. Fortunately, there is a unique closed-form solution to this problem and it was given in (16). Therefore, when using a JFA paradigm based on point estimates, the model training process is equivalent to a signal approximation. In this case, the signal being approximated happens to be the offsets—with respect to the UBM supervector $\mathbf{m}$—of the normalized first order statistics $\boldsymbol{\eta}$, contextualized by the soft-partition of the acoustic space induced by the UBM.

3.2. Dictionary Learning

Following the jargon particular to the sparse coding community, we will refer to the matrix $\mathbf{\Phi}$ as a dictionary whose columns are denoted as atoms. For JFA, the dictionary is comprised of $\mathbf{\Phi}_{JFA} = [\mathbf{UVD}]$ and is considered overcomplete since there are more columns than rows. This notation also applies to the eigenchannel configuration $\mathbf{\Phi}_{ECH} = [\mathbf{UD}]$ as well as the relevance MAP formulation $\mathbf{\Phi}_{rMAP} = \mathbf{D}_{rMAP}$ (although in this last case the dictionary is not overcomplete). The atoms of the dictionary should represent the basic constituent elements of the signals being coded as well as their typical distortions. In order for this to be the case, the best alternative is to learn these atomic representations from actual data similar to the one being coded. Thus, the process of learning a dictionary from data is equivalent to the estimation of hyperparameters in JFA. Specifically, given a training data set $\mathfrak{D} = \{\boldsymbol{\mathcal{O}}_r\}_{r=1}^R$ with R utterances—after applying the E-step described in the (SC) section—the information in each utterance $\boldsymbol{\mathcal{O}}_r$ is represented by the pair $(\boldsymbol{\eta}_r, \mathbf{W}_r)$. Hence, the dictionary training problem is expressed as:

$$\min_{\mathbf{\Phi}, \{\boldsymbol{\beta}_r\}} \sum_{r=1}^{R} \left\| \mathbf{W}_r^{\frac{1}{2}} (\boldsymbol{\eta}_r - \mathbf{\Phi}\boldsymbol{\beta}_r) \right\|_2^2 + \|\boldsymbol{\beta}_r\|_2^2 . \qquad (19)$$

Note that unlike equation (15), the objective in (19) also involves the dictionary as an optimization variable. Hence, even though when considered as a function of either $\{\boldsymbol{\beta}_r\}$ or $\mathbf{\Phi}$ the objective is convex, it is not the case for the joint optimization in (19). This situation arises quite frequently and the use of alternating optimization [10] is one of the most conventional ways to address it.

3.2.1. Block coordinate descent (BCD) minimization

A particular configuration of alternating optimization known as block coordinate minimization (a.k.a non-linear Gauss-Seidel) is well suited for the case at hand [10]. Specifically, we

[1] Note that in the case where not all Gaussians are responsible for at least one observation, the matrix $\mathbf{W}_{test}$ is in fact positive semi-definite. In that case, equation (18) is still correct if we define the inner product in the subspace where the diagonal entries of $\mathbf{W}_{test}$ are strictly positive.

consider a two step process. In one step, the block of variables $\mathbf{\Phi}$ is fixed and the objective is minimized with respect to $\{\boldsymbol{\beta}_r\}$. In the other step, the dictionary is updated while keeping the coefficients obtained in the previous step $\{\boldsymbol{\beta}_r\}^{opt}$ fixed. Cycling between these two steps is repeated until convergence or sufficient decrease of the objective is obtained. Because the joint objective in (19) is non-convex this method only finds a local minimum and different initial values of the dictionary $\mathbf{\Phi}_0$ lead to different solutions. Note that the first step is exactly the SC stage described in the previous section. The second step is denoted as dictionary update (DU) and is addressed next.

3.2.2. *Dictionary Update (DU)*

Keeping the regression coefficients fixed for all utterances, the minimization of the objective in (19) with respect to the dictionary reduces to

$$\min_{\mathbf{\Phi}} \sum_{r=1}^{R} \left\| \mathbf{W}_r^{\frac{1}{2}} (\boldsymbol{\eta}_r - \mathbf{\Phi}\boldsymbol{\beta}_r) \right\|_2^2. \tag{20}$$

A simple application of the definition of convexity reveals that for any given utterance $(\boldsymbol{\eta}_r, \mathbf{W}_r)$ the corresponding term inside the summation is convex. Subsequently, the positive sum of R convex functions remains convex. Note that unlike in the SC step, in general the DU objective in (20) is not strongly convex and therefore there is no guarantee for a unique minimizer. However, any local minimizer is also global. Again, due to convexity, computing the gradient and setting it to zero provides a necessary and sufficient condition for a local/global minimizer. Using the identity $\mathbf{a}^T\mathbf{b} = \mathrm{tr}\{\mathbf{b}\mathbf{a}^T\}$ the problem in (20) is equivalent to

$$\min_{\mathbf{\Phi}} \sum_{r=1}^{R} \mathrm{tr}\{\mathbf{W}_r \mathbf{\Phi}\boldsymbol{\beta}_r\boldsymbol{\beta}_r^T\mathbf{\Phi}^T\} - 2\,\mathrm{tr}\{\mathbf{W}_r\boldsymbol{\eta}_r\boldsymbol{\beta}_r^T\mathbf{\Phi}^T\}. \tag{21}$$

Setting the gradient with respect to the dictionary $\mathbf{\Phi}$ to zero results in

$$\sum_{r=1}^{R} \mathbf{W}_r \mathbf{\Phi}\boldsymbol{\beta}_r\boldsymbol{\beta}_r^T = \sum_{r=1}^{R} \mathbf{W}_r\boldsymbol{\eta}_r\boldsymbol{\beta}_r^T. \tag{22}$$

Which, when restricted to the F rows corresponding to the k^{th} mixture simplifies to

$$\mathbf{\Phi}_{new}^{(k)} \left(\sum_{r=1}^{R} \gamma_{rk}\boldsymbol{\beta}_r\boldsymbol{\beta}_r^T \right) = \sum_{r=1}^{R} \gamma_{rk}\boldsymbol{\eta}_r^{(k)}\boldsymbol{\beta}_r^T. \tag{23}$$

Comparing this result with the one obtained in (17) we can see that they are the same if we set the posterior covariance matrices $cov(\boldsymbol{\beta}_r\boldsymbol{\beta}_r^T)$ to zero. This is consistent with our formulation since we are ignoring the underlying probabilistic assumptions of the JFA model and treating the problem as a simple signal approximation.

3.2.3. *Dictionary learning algorithm*

An important algorithmic opportunity arises from this new perspective. In particular, we are going to exploit the computational advantage derived from the fact that no explicit matrix inversions are necessary. That is, we no longer need to compute $(\mathbf{I} + \mathbf{\Phi}^T\mathbf{W}_r\mathbf{\Phi})^{-1}$ explicitly for each utterance to perform the dictionary update. This observation affects the DU step slightly but the most important gain comes from the SC step of the dictionary learning process. That is, much faster and numerically stable methods like Gauss-Seidel [11] or Cholesky factorizations can be used in the SC step[1] since no explicit matrix inversions are needed. Regarding the DU step, denoting the sum of R rank-one matrices corresponding to the k^{th} mixture by

$$\sum_{r=1}^{R} \gamma_{rk}\boldsymbol{\beta}_r\boldsymbol{\beta}_r^T = \mathbf{A}_R^{(k)} \in \mathbb{R}^{P\times P} \tag{24}$$

and assuming that R is large enough so that $\mathbf{A}_R^{(k)}$ is invertible, the updated $\mathbf{\Phi}_{\text{new}}$ requires the solution of K independent systems of P equations with F right-hand side elements. A hybrid update formula between (23) and (17) can be obtained by setting

$$\mathbf{A}_R^{(k)} = \gamma_k \left(\mathbf{I} + \mathbf{\Phi}_o^T \mathbf{W}_{avg}^{(k)} \mathbf{\Phi}_o \right)^{-1} + \sum_{r=1}^{R} \gamma_{rk}\boldsymbol{\beta}_r\boldsymbol{\beta}_r^T, \tag{25}$$

where $\mathbf{\Phi}_o$ comes from the previous iteration of the DU (or from a simple PCA initialization for the first iteration). Also, $\mathbf{W}_{avg}^{(k)} = \mathbf{\Gamma}_{avg}^{(k)}\mathbf{\Sigma}^{-1}$ with $\mathbf{\Gamma}_{avg}^{(k)} = \frac{1}{R}\sum_{r=1}^{R}\mathbf{\Gamma}_r^{(k)}$ from the training set and $\gamma_k = \sum_{r=1}^{R}\gamma_{rk}$. In this way, instead of completely neglecting the covariance matrices $cov(\boldsymbol{\beta}_r\boldsymbol{\beta}_r^T)$ of the JFA model, we approximate all of them with a common one obtained by averaging the occupancy matrices $\mathbf{\Gamma}_r^{(k)}$ over the entire training set. Also, using (25) removes any uncertainty about $\mathbf{A}_R^{(k)}$ being invertible.

Dictionary learning algorithm

Input: $\{(\boldsymbol{\eta}_r, \mathbf{W}_r)\}_{r=1}^{R}$ and $\mathbf{\Phi}_o$
Initialize: $\mathbf{W}_{avg} = \frac{1}{R}\sum_{r=1}^{R}\mathbf{W}_r$,
$\mathbf{\Phi}_{new} = \mathbf{\Phi}_{old} = \mathbf{\Phi}_o$
Until convergence:
  SC: Solve for each $\boldsymbol{\beta}_r$ in (16) using Gauss-Seidel or Cholesky with $\mathbf{\Phi}_{new}$
  Dictionary update (DU):
    For each mixture $k = 1{:}K$
      $\mathbf{A}_o = \gamma_k \left(\mathbf{\Phi}_{old}^T \mathbf{W}_{avg}^{(k)} \mathbf{\Phi}_{old} + \mathbf{I} \right)^{-1}$ or $\mathbf{A}_o = 0$ and $\mathbf{C}_o = 0$
      For each utterance $r = 1{:}R$
        $\mathbf{A}_r = \mathbf{A}_{r-1} + \gamma_{rk}\,\boldsymbol{\beta}_r\boldsymbol{\beta}_r^T$
        $\mathbf{C}_r = \mathbf{C}_{r-1} + \gamma_{rk}\boldsymbol{\eta}_r^{(k)}\boldsymbol{\beta}_r^T$
      End for each utterance
      Solve $\mathbf{\Phi}_{new}^{(k)}\mathbf{A}_R = \mathbf{C}_R$ using Gauss-Seidel or Cholesky
    End for each mixture
    $\mathbf{\Phi}_{new} = [\mathbf{\Phi}_{new}^1; \ldots; \mathbf{\Phi}_{new}^K]$
  End Dictionary Update
End until convergence

Figure 1. Dictionary learning algorithm based on alternating minimization with two steps.

Figure 1 summarizes the proposed algorithm for the dictionary learning process. Note that throughout the theoretical presentation in sections 2 and 3 we have used the dictionary $\mathbf{\Phi}$ as a wild-card notation to refer to multiple combinations of the loading matrices $\mathbf{U}, \mathbf{V}$ and $\mathbf{D}$. Hence, the dictionary learning algorithm in figure 1 should be applied in a way consistent with the configuration at hand. As it was the case for the hyperparameter estimation procedure in section 2, the

[1] These techniques are also suitable for the JFA model, but if used instead of an explicit inversion, the task of computing the posterior covariance matrices still remains.

formulation presented in this section is only applicable for the decoupled/independent estimation of $\mathbf{U}, \mathbf{V}$ and $\mathbf{D}$. Therefore, the way to present the data to the dictionary learning algorithm should be consistent with this approach. An experimental analysis regarding the influence of the choice of $\mathbf{A}_R^{(k)}$ in the resulting dictionary $\mathbf{\Phi}$ is presented in section 4. Moreover, the influence in speaker recognition performance is also analyzed.

3.3. Scoring

Given two utterances A and B defined by $(\boldsymbol{\eta}_A, \mathbf{W}_A)$ and $(\boldsymbol{\eta}_B, \mathbf{W}_B)$—after coding them with the dictionary $\mathbf{\Phi}_{JFA} = [\mathbf{UVD}]$—we obtain two approximations $\widehat{\boldsymbol{\eta}}_A = \mathbf{\Phi}_{JFA}\boldsymbol{\beta}_A$ and $\widehat{\boldsymbol{\eta}}_B = \mathbf{\Phi}_{JFA}\boldsymbol{\beta}_B$. Since some of the atoms in the dictionary are explicitly representing undesired distortions (i.e., columns of $\mathbf{U}$), setting to zero the corresponding entries in $\boldsymbol{\beta}_A$ and $\boldsymbol{\beta}_B$ yields a compensated approximation of the signals $\widehat{\boldsymbol{\eta}}_{A|c}$ and $\widehat{\boldsymbol{\eta}}_{B|c}$. Once these compensated signal approximations are obtained, a similarity measure can be defined by means of an inner product

$$score = \langle \widehat{\boldsymbol{\eta}}_{A|c}, \widehat{\boldsymbol{\eta}}_{B|c} \rangle_{\mathbf{W}_\#} = \widehat{\boldsymbol{\eta}}_{A|c}^T \mathbf{W}_\# \widehat{\boldsymbol{\eta}}_{B|c} \tag{26}$$

where $W_\#$ can be any symmetric positive definite matrix. Immediate candidates are $\mathbf{W}_A, \mathbf{W}_B$ and $\mathbf{W}_{UBM}$. Note that from the perspective of signal coding, the concepts of model and test segment are blurred since both utterances are represented in the same way. However, if we identify $\widehat{\boldsymbol{\eta}}_{A|c}$ as a model and set $\mathbf{W}_\# = \mathbf{W}_B$ the only difference between (26) and the linear approximation of the log-likelihood ratio in (18) is the way in which the test segment is encoded. Specifically, the test segment is represented by simply removing its encoding with respect to the atoms in $\mathbf{U}$ from $\boldsymbol{\eta}_B$. A comparison of both approaches is presented in the next section. Finally, another interesting idea that will be explored in the experiments is the effect of normalizing the scores (i.e., using the cosine of the angle between the compensated approximations as the similarity measure).

$$norm\_score = \frac{\langle \widehat{\boldsymbol{\eta}}_{A|c}, \widehat{\boldsymbol{\eta}}_{B|c} \rangle_{\mathbf{W}_\#}}{\langle \widehat{\boldsymbol{\eta}}_{A|c}, \widehat{\boldsymbol{\eta}}_{A|c} \rangle_{\mathbf{W}_\#}^{1/2} \langle \widehat{\boldsymbol{\eta}}_{B|c}, \widehat{\boldsymbol{\eta}}_{B|c} \rangle_{\mathbf{W}_\#}^{1/2}} \tag{27}$$

This normalization technique has already produced successful results when used as a kernel for SVMs on the speaker factor space spanned by the columns of $\mathbf{V}$ [12]. Moreover, an extension of that work into a new subspace—denoted as total variability space—has validated the excellent discriminative power of this similarity measure [13]. However, to the best of our knowledge, no use of this normalization has been directly studied in the mean supervector space.

4. Experimental setup

4.1. Switchboard-I database (SWB-I)

The Switchboard-I database is comprised of conversational speech between two speakers recorded over landline telephone channels with a sampling rate of 8 KHz. The average duration of each conversation is 5 minutes (approx. 2.5 min per speaker) and each conversation side is recorded in a different file. The total number of speakers in the database is 520 with a balance in gender and recorded into 4856 speech files. The telephone handsets were either electret or carbon button with an approximate proportion of 70% and 30% respectively.

4.2. Recognition system configuration

Each file in the database was parameterized into a sequence of 19-dimensional MFCC vectors using a 20ms Hamming window with a 10ms shift. The MFCC vectors were computed using a simulated triangular filterbank on the FFT spectrum. Prior to projecting the Mel-frequency band (MFB) energies into a DCT basis, bandlimiting was performed by discarding the filterbank outputs outside of the frequency range 300Hz-3138Hz. Finally, after projecting the MFB energies into a DCT basis and discarding C0, the 19-MFCC vectors were augmented with delta features resulting in $F = 39$ coefficients per frame.

SWB-I was partitioned into two sets, ***P1*** and ***P2***, of 260 speakers each with a balance in gender and handset type. A 2048-mixture gender-independent UBM with diagonal covariance matrices was trained on ***P2***. The data in ***P2*** was also used for hyperparameter/dictionary learning. In particular, we used an eigenchannel setup $\mathbf{\Phi}_{ECH} = [\mathbf{UD}]$ with $KF = 77824$, $U \in \mathbb{R}^{KF \times P}$ and the standard relevance-MAP diagonal matrix $\mathbf{D}$ was fixed to $\mathbf{D}^2 = \Sigma/\tau$ with $\tau = 16$. This configuration is general enough to validate our theoretical developments while avoiding unnecessary complexity in illustrating the underlying principles of the proposed techniques.

4.3. Experiments

In order to evaluate the theoretical exposition of the previous section we present three different sets of experiments. The first one is concerned with the effects of different DU steps in the learned dictionaries as well as the effect in speaker recognition accuracy. The second set of experiments is designed to evaluate the influence of different signal coding strategies along with various types of inner products for scoring. Finally, the third batch of experiments analyzes the influence of the normalization of the scores according to (27) in a verification task and compares our proposed similarity measure with the linear approximation of the log-likelihood introduced in [8].

4.3.1. Analysis of dictionary learning procedure

Equation (17) from the JFA model as well as equations (24) and (25) from the SC model provide three different DU mechanisms. We will refer to the updates in (17), (24) and (25) as Full, Zero and Average updates respectively. This notation stems from the fact that (17) takes a full account of the posterior covariance matrices; (24) can be understood as setting them to zero; and (25) considers a common and averaged covariance matrix for all utterances in the dictionary training set. The computational advantages of (24) and (25) over (17) were briefly discussed in section 3.2. However, the effects of this computational saving in the learned dictionaries are not evident and thus require some experimental analysis. We would like to know how the dynamics of the sequence of dictionaries generated by multiple iterations of the dictionary learning algorithm in figure 1 are affected. To study this, we apply the dictionary learning algorithm with the full, average and zero updates to obtain a sequence of eigenchannel subspaces $\mathbf{U}_F(i), \mathbf{U}_A(i)$ and $\mathbf{U}_z(i)$ with $i = 0, \ldots, 10$. The 2411 utterances coming from the 260 speakers of ***P2*** where used for each iteration. To quantify the similarity between two subspaces, we used a metric between the subspaces spanned by the columns of the matrices $\mathbf{A} \in \mathbb{R}^{FK \times P_A}$ and $\mathbf{B} \in \mathbb{R}^{KF \times P_B}$ known as the projection distance [14]

$$pdist(\mathbf{A},\mathbf{B}) = \|\mathbf{A}\mathbf{A}^T - \mathbf{B}\mathbf{B}^T\|_F^2. \quad (28)$$

Since the projection distance is at most the $\min(P_A, P_B)$, we normalized (28) to produce results between [0,1]. Figure 2 shows the projection distance of the average and zero updates with respect to the full update. The curves with the triangle markers refer to the distance between the full and average updates. The curves with the asterisk indicate the distance between the full and the zero update. Moreover, the color codes refer to the dimension of the subspaces computed (i.e., blue=128, green=64 and red=32 dimensions). A simple look at the y-axis shows that the normalized projection distances are very low for all configurations (since the maximum possible value is 1). Furthermore, the larger the dimensionality of the subspaces the larger the projection distances. As expected, the distance of the subspaces produced by the average update is smaller than those produced by the zero update. These results confirm that the three DU techniques produce very similar dictionaries.

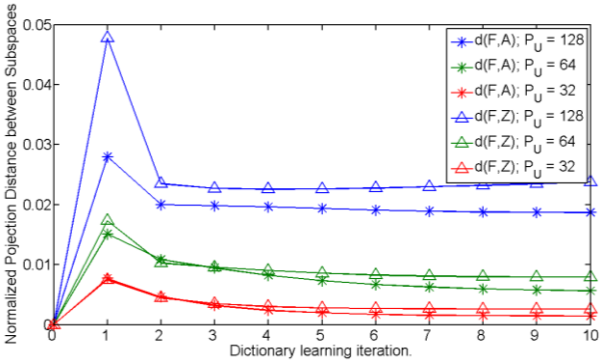

Figure 2. Normalized projection distance between subspaces of dimensions 128, 64 and 32 learned using different DU formulas.

Even though the distance between subspaces might not be too large, the effects in the recognition accuracy may not behave in the same way. To check this, a closed-set identification experiment was used. We coded each of the 2408 utterances from partition ***P1*** using the dictionaries obtained after the 6th iteration. The normalized score in equation (27) was used with the inner product defined by the weights and covariance matrices of the UBM. We obtained 33866 identification trials based on the 2408 utterances. The details about how we constructed these trials are provided in next section. Table 1 shows that the effect in identification accuracy is negligible. Hence, we can claim that for a scenario where enough utterances are available for dictionary training, the average and zero update rules provide computational advantages without any significant loss in performance. The robustness of these two techniques to amount of data is still an open issue for future research.

$\mathbf{P}_U$	Full $\mathbf{A}_R^{(k)}$	Avg. $\mathbf{A}_R^{(k)}$	Zero $\mathbf{A}_R^{(k)}$
128	95.0%	94.9%	94.9%
64	94.5%	94.5%	94.5%
32	93.3%	93.3%	93.3%

Table 1. Closed-set identification accuracy for dictionaries learned with three DU formulas (full, average and zero). Three dimensios of the eigenchannel space **U** are presented.

4.3.2. Closed-set speaker identification

This section explores the influence of different signal coding strategies along with various types of inner products in the context of speaker identification. We intentionally selected an identification setup in order to remove the influence of a verification threshold from the analysis. We obtained 33866 identification trials based on the 2408 utterances from 260 speakers in ***P1***. The protocol followed was as follows, for each speaker we picked one of its utterances and encode it to represent a model. Then, another utterance form that same speaker was selected as the test segment; and the remaining utterances from the rest of the speakers were used as models. This procedure was repeated exhaustively for all the utterances of each speaker and for all the speakers. The dimensionality of the eigenchannel space was explored and 128 dimensions produced the best results. Also, the average update rule was used in the learning process.

Figure 3 shows the influence of three different inner products in our SC formulation with normalized scoring (bottom left panel). The three inner products are defined by the matrices $\mathbf{W}_I = \mathbf{I}$, $\mathbf{W}_{ubm}$ and $\mathbf{W}_{test}$. The last two have already been discussed and the first one indicates the standard inner product. For comparison, we also analyze the influence of these inner products in other techniques such as: ML model training (top left), relevance MAP (top center), and the standard eigenchannel configuration with linear scoring (bottom center). A general trend is observed regardless of the modeling technique used; the use of the standard inner product performs much worse in all cases. This makes sense since not all the information is evenly distributed across the acoustic space. Therefore, penalizing by the amount of data (i.e., small value of the first order statistics) as well as the variability within the soft regions associated with each Gaussian (i.e., covariance of the UBM) is very effective. This concept is not new and has been exploited in the formulation of the KL-kernel (i.e., inner product defined by $\mathbf{W}_{ubm}$) in [15]. The results obtained with $\mathbf{W}_{ubm}$ and $\mathbf{W}_{test}$ change depending on the modeling strategy followed. For our SC approach, the use of both inner products produces comparable results. However, for the standard eigenchannel model with linear scoring, $\mathbf{W}_{test}$ produces significantly better results (and in the same range as the SC approach). The sensitivity with respect to the inner product is understandable since the linear scoring is an approximation of the log-likelihood ratio and by changing the inner product the approximation is less accurate.

After the first two iterations not much difference is obtained in the identification performance. This extremely fast convergence might be explained by the fact that the dictionary training data and the test set are very similar. Also, identification results based on the factors (bottom right) and the information in the eigenchannel subspace (top right) are included for diagnostic purposes. In particular we can observe that the eigenchannel subspace also contains speaker information since an accuracy of almost 70% is obtained. The factors **x** and **z** behave as expected. Finally, even though not explicitly shown in the paper, the performance of the normalized and un-normalized scoring techniques was assessed. No significant difference was observed for neither the SC approach nor the standard eigenchannel formulation. This makes sense since for identification purposes what matters is the relative positioning of scores and not their scaling. In the next section we explore this issue in the context

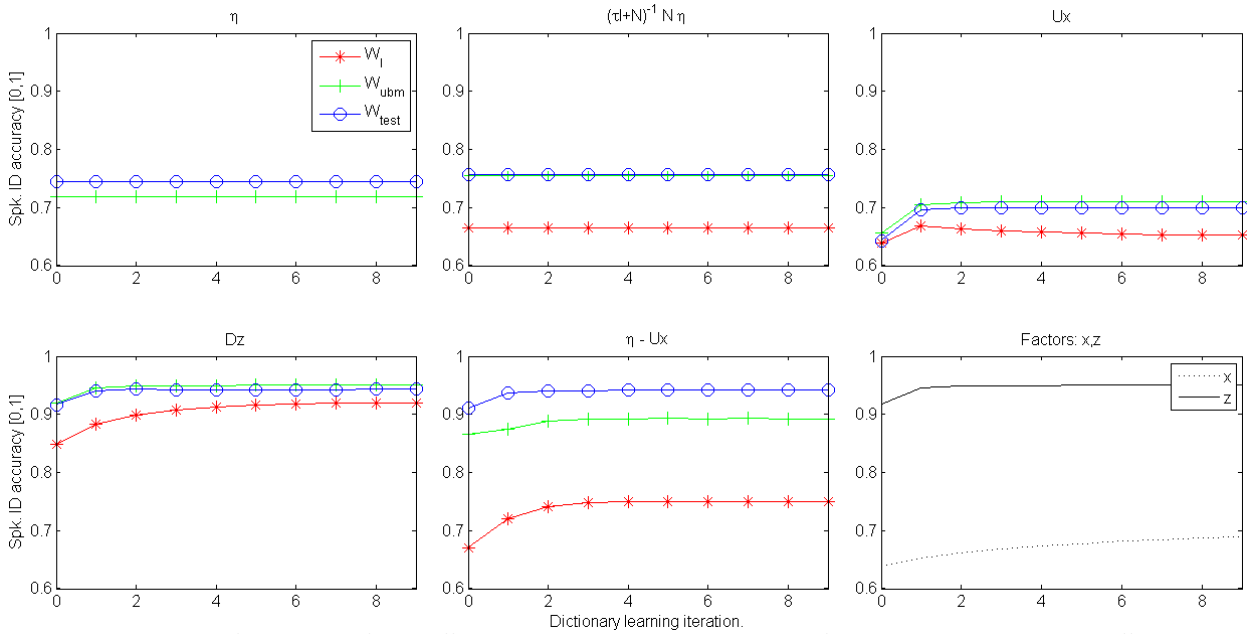

Figure 3. Closed-set identification results for six different modeling approaches (see main text for description) along with three different scoring techniques based on the inner products defined by the symmetric positive definite matrices $\mathrm{W}_I, W_{ubm}, W_{test}$.

of speaker verification where the scaling of the scores is critical.

4.3.3. Speaker verification

Based on the 2408 utterances from the 260 speakers in ***P1*** a verification experiment was designed. Specifically, a leave-one-out strategy was used. That is, each file was used as a model and the remaining 2407 utterances were used as test segments. This protocol produced a great number of trials (33,866 target and 5,764,598 non-target). However, since our proposed scoring as well as the linear scoring methods are simple inner products between high dimensional vectors, the entire set of trials was computed in a less than 5 minutes. The main purpose of this setup was to assess the influence of the score normalization proposed in (27).

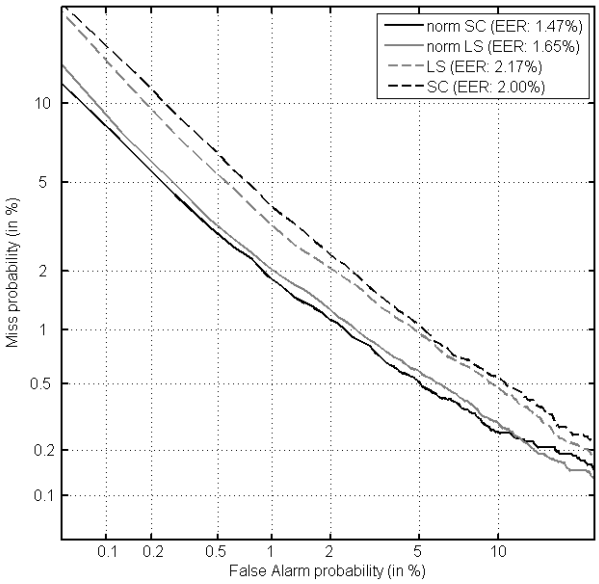

Figure 4. Verification results for different scoring methods.

Figure 4 shows the verification results. Three observations are in place. First, using the cosine of the angle between the vectors results in more than a 25% relative improvement in EER for both linear scoring in (17) and our proposed un-normalized inner product of (26). Second, the effects of the normalization are slightly better for the our approach. Finally, while the performance of the un-normalized scores is better for the linear scoring, the normalized SC scores produce slightly better performance under normalization.

5. Conclusions

We have established a connection between the Joint Factor Analysis paradigm for speaker recognition and signal coding using an overcomplete dictionary learned from data. The probabilistic concepts of model training, hyperparameter estimation and likelihood ratio computation were equated to the non-probabilistic notions of signal coding, dictionary learning and similarity computation respectively. Two novel ideas were proposed that resulted in faster hyperparameter estimation and improved scoring. Specifically, the proposed technique for hyperparameter estimation was able to avoid the need for explicit matrix inversions in the M-step of the ML estimation. This allowed the use of faster techniques such as Gauss-Seidel or Cholesky factorizations for the computation of the posterior means of the factors $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ during the E-step. Regarding the scoring, different similarity measures based on inner products—defined by symmetric positive definite matrices derived from data—were studied. A simple normalization technique of these inner products was shown to improve the verification performance of our recognition system using a dictionary comprised of eigenchannels and a fixed relevance-MAP matrix $\mathbf{D}$. Based on this experimental setup, slightly better results than those produced by the state-of-the-art linear scoring approach were reported. The experimental validation of these two novel techniques was presented using closed-set identification and speaker verification experiments over the Switchboard database.

6. Bibliography

[1] P. Kenny and P. Dumouchel, "Experiments in Speaker Verification using Factor Analysis Likelihood Ratios," in *Proceedings of Odyssey04 - Speaker and Language Recognition Workshop*, Toledo, Spain, 2004.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning* , 1st ed.: Springer, 2006.

[3] L. Burget, P. Matejka, H. Valiantsina, and J. Honza, "Investigation into variants of Joint Factor Analysis for Speaker Recognition," in *Interspeech 2009*, Brighton, 2009, pp. 1263-1266.

[4] P. Kenny, "Joint Factor Analysis of Speaker and Session Variability : Theory and Algorithms," CRIM, Montreal, (Report) CRIM-06/08-13, 2005.

[5] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, pp. 19-41, 2000.

[6] R. Vogt, B. Baker, and S. Sridharan, "Factor analysis subspace estimation for speaker verification with short utterances," in *Interspeech*, Brisbane, 2008.

[7] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A Study of Inter-Speaker Variability in Speaker Verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 5, pp. 980-988, July 2008.

[8] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny, "Comparison of scoring methods used in speaker recognition with Joint Factor Analysis," in *ICASSP*, 2009, pp. 4057-4060.

[9] R. Rubinstein, A.M. Bruckstein, and M. Elad, "Dictionaries for Sparse Representation Modeling," *Proceedings of the IEEE*, 2010 (to appear).

[10] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed.: Athena Scientific, 1999.

[11] R. Vogt and S. Sridharan, "Explicit Modelling of Session Variability for Speaker Verification," *Computer Speech and Language*, vol. 22, no. 1, pp. 17-38, 2008.

[12] N. Dehak et al., "Support Vector Machines and Joint Factor Analysis for Speaker Verification," in *ICASSP*, 2009, pp. 4237 - 4240.

[13] N. Dehak et al., "Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification," in *Interspeech*, Brighton, 2009, pp. 1559-1562.

[14] A. Edelman, T. Arias, and S. Smith, "The Geometry Of Algorithms With Orthogonality Constraints ," *SIAM J. Matrix Anal. Appl.*, 1999.

[15] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support Vector Machines Using GMM Supervectors for Speaker Verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308-311, May 2006.