ON THE USE OF SPEAKER SUPERFACTORS FOR SPEAKER RECOGNITION

Nicolas Scheffer and Robbie Vogt

[1]SRI International
[2] Queensland University of Technology
nicolas.scheffer@sri.com, r.vogt@qut.edu.au

ABSTRACT

We propose a new method to characterize a speaker within the Joint Factor Analysis (JFA) framework. Scoring within the JFA framework can be costly and a new method was proposed to produce an accurate score in a fast manner. However, this method is nonsymmetric and performs badly without any score normalization. We propose a new JFA scoring method that is both symmetrical and efficient. In the same way as means of Gaussians can be concatenated to form a supervector, we use several estimates of speaker factors from the eigenvoice space to build a supervector of factors that we call *superfactors*. We motivate the use of such factors in the current JFA model through comparison with a Tied Factor Analysis model. We show that this method substantially improves the performance of a system that uses only the standard speaker factors to produce scores, and usually outperforms the baseline system. We also show that this method is relatively effective even when score normalization is not an option.

Index Terms: speaker recognition

1. INTRODUCTION

Modeling variability in the model space is a major focus of the speaker recognition community. This work has shown to be particularly useful for channel compensation of speaker models. One of the most developed frameworks for tackling this problem is Joint Factor Analysis (JFA), introduced by Patrick Kenny in [1]. This framework aims at factoring out two components for an utterance: the speaker and the nuisance component (usually called *channel* or *session variability*). The latter is commonly removed for training a speaker model.

In this work, we investigate a promising new method to characterize a speaker within the JFA framework. In [2], the authors show the multitude of methods that one can use to produce scores for the JFA system. The method currently most used is a linear approximation of the log likelihood ratio using a dot product formulation.

The first disadvantage of this technique is its asymmetry. Indeed, while previously successful, the symmetrical Kullback Leibler (KL) kernel does not work using an eigenvoice-based system. In [3], the authors propose the use of the latent variables of the JFA model as a feature for a support vector machine (SVM) classifier where the speaker factors characterize the speaker entirely. One advantage of such a method, apart from its speed, is the symmetry of the extracted features. Indeed, the symmetry enables the use of an SVM classifier to produce speaker ID scores. It also, by nature, treats the training and the testing utterances in the same manner, avoiding a potential loss in information. However, this method does not reach the performance of a baseline system. The authors claim that a secondary channel compensation technique is required in the SVM kernel space for competitive performance. The present work proposes a simpler approach.

For both methods and particularly when using a dot product score between speaker factors, the performance is very poor without any score normalization. There exists multiple scenarios when one wants to produce speaker ID scores but cannot perform score normalization, e.g., for speaker diarization, language identification systems, when differentiating an optimization function, or simply when appropriate data for normalization is not available.. In this scenario, the frame-by-frame scoring strategy gives better performance but at a very low speed.

This work attempts to improve the characterization of the speaker in the eigenvoice domain. We give more insight on the use of the speaker factor by proposing the construction of speaker factor supervectors, that we call **superfactors**. In the same way that Gaussian mixture model (GMM) mean supervectors can be built, we propose to derive several speaker factor estimates to build a supervector. To reach this goal, we look at the eigenvoice system in a new way by showing a close relationship to the Tied Factor Analysis paradigm. We will then present a new method to build speaker superfactors based on Gaussian clusters of the universal background model (UBM).

2. SYSTEMS AND PROTOCOL

We describe the JFA framework and the system configuration used for the experiments, and then present the experimental protocol.

2.1. Joint Factor Analysis

Under the JFA framework, speakers and utterances are modeled as mean offsets from an underlying GMM distribution of a UBM represented as a *supervector*. Let the number of Gaussians of the GMM be N and the feature dimension be F. Then, a supervector is a vector of the concatenation of the GMM component means vectors with dimensions NF. The UBM has a supervector mean $\mathbf{m}$ and diagonal covariance $\mathbf{\Sigma}$. The speaker component of the JFA model is a factor analysis model on the speaker GMM supervector. It is composed of a set of eigenvoices and a diagonal model. Precisely, the supervector m_s of a speaker s is governed by

$$m_s = \mathbf{m} + Vy + Dz \tag{1}$$

where V is a tall matrix of dimension $NF \times R_S$, the eigenvoices (or speaker loadings), spanning a subspace of low-rank R_S. D is the diagonal vector of the factor analysis model of dimension NF. Two latent variables y and z entirely describe the speaker and are subjected to the prior $N(0, 1)$. The nuisance (or channel) supervector distribution also lies in a low-dimensional subspace of rank R_C.

The supervector for an utterance h with speaker s is

$$m_h = m_s + Ux \tag{2}$$

The matrix U, the eigenchannels (or channel loadings), has a dimension of $NF \times R_C$. The loadings U, V, D are estimated from a sufficiently large data set while the latent variables x, y, z are estimated for each utterance.

2.2. Baseline System and JFA Model Estimation

The baseline system employs gender-dependent 512-Gaussian UBMs. Cepstral features are mel frequency cepstrum coefficients (MFCCs) composed of 12 cepstrums, adding first order derivatives, for a total dimension of $F = 24$. The rank of the speaker space is $R_S = 300$ and for the channel space is $R_C = 100$.

To train the matrices U, V and D, several iterations of the expectation maximization (EM) algorithm of the factor analysis framework are used. An alternative minimum divergence estimation (MDE) is used at the second iteration to scale the latent variables to an $N(0, 1)$ distribution. To train a speaker model, the posteriors of x, y, z are computed using a single iteration (via the Gauss-Seidel method as in [10]).

The UBM parameters $\mathbf{m}$ and $\mathbf{\Sigma}$, the eigenvoices V, and the diagonal model D were trained in a gender-dependent fashion on a large selection of Mixer data. The eigenchannels U were trained on the NIST SRE 04 data set and on *alternate microphone* data from NIST-SRE-2005.

The speaker verification systems additionally have gender-dependent score normalization (ZT-norm) applied using a selection of utterances per gender drawn from SRE 04 and 05. All experiments were performed on the NIST SRE 2008 data set, Conditions 6 and 7. Results are given in terms of equal error rate (EER).

2.3. Baseline results

The results of a baseline system, shown in Table 1, are included as a point of comparison. For this system the verification score for each trial was a scalar product between the speaker model mean offset and the channel-compensated first-order Baum-Welch (BW)) statistics centered around the UBM. This scalar product was found to be simple yet very effective [5] and was subsequently adopted by the community [6]. We also show the baseline results without the diagonal component Dz of the factor analysis model as this component is also omitted in the proposed approach. This system gets a slight degradation. The KL kernel scoring approach is also shown. We can clearly see that the addition of eigenvoices prevents the use of this method.

While we compare our results to a baseline system, this approach aims at improving the methods developed in [3] where the speaker factors were used for scoring. In this work, however, we did not find any advantage of using SVMs to produce models and perform recognition. The speaker recognition score is thus produced using a dot product between the vectors (precisely, we used a cosine kernel, where each vector is divided by its norm). Also shown are the results of the cosine kernel on the speaker factor as in [3]. We can see that this method is not suitable to replace the baseline system.

In this paper, we aim at defining a symmetric scoring method, that we named *superfactor scoring*, that performs as well as the baseline system.

Table 1: *Baseline experiments with different types of scoring. EER results on NIST-SRE-2008, Conditions 6 and 7. The performance of methods using a symmetrical scoring is much worse than the baseline.*

System	Cond. 6/7 (EER (%))
baseline	8.03/3.91
baseline $D = 0$	8.44/4.48
KL kernel	10.08/6.03
Speaker factor cosine kernel	10.01/5.54

3. TIED ESTIMATION OF EIGENVOICES

Tied Factor Analysis [7] (Tied FA) constrains data from differing views or classes to produce the same latent variables in the "identity space" by training a distinct loading matrix for each view. The advantage of this approach is that the estimates of the latent variables are still valid even if the data from one or several classes is missing.

In [7], the authors apply this methodology for face recognition among large variations in poses, particularly to enable face recognition from a pose, for which we do not have a training example. While Tied FA was developed to address a missing data problem, the tied approach can be seen as an attempt to gain from observing multiple "views" of a speaker.

Figure 1 illustrates the process of training such a system using the gender of the speaker as classes. In algorithm 1, we explain how to train the tied matrices to produce a single estimate of y.

Data: Concatenated Baum-Welch statistics for each utterance across genders
Result: Tied estimate of the eigenvoice space
while *convergence is not met* **do**
  Estimate tied speaker factors y_{tied} from the concatenated eigenvoice matrices $(V_m V_f)$;
  for *each gender g* **do**
    find the maximum likelihood of V'_g using y_{tied};
  end
end

Algorithm 1: Training a tied eigenvoice matrix across genders.

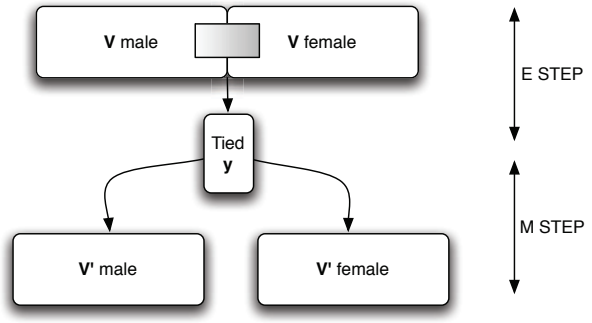

Fig. 1: *Diagram of the EM algorithm for the Tied Factor Analysis system. The estimate of the speaker factors y_{tied} is tied across genders.*

In [8], we showed that a Tied FA algorithm would give the same performance as an "augmented" FA system, where the standard FA algorithm is applied on concatenated BW statistics. Some further analysis indicates that these two algorithms behave very similarly and produce eigenvoice matrices and speaker factors that are extremely similar.

Technically, if the estimation of eigenvoices can be expressed per Gaussian and if the factors are tied across the classes, then the reestimation will be identical. As a result, the JFA model can be interpreted as a Tied Factor Analysis model where each class is a Gaussian, and a single estimate of the speaker factor is to be considered as tied across all Gaussians. The factor analysis system (without the channel matrix) can be rewritten as

$$m_s = \mathbf{m} + \begin{bmatrix} V_1 \\ \vdots \\ V_G \end{bmatrix} \mathbf{y}, \tag{3}$$

where $\mathbf{y}$ is a single estimate of the speaker factors produced by all Gaussians and $V_1, \ldots V_G$ are blocks of V corresponding to the Gaussians. The single estimate of the speaker factor can be considered as tied across all Gaussians. The main idea underlying the *superfactors* method is that, while the estimations of the matrices are tied, the individual matrices tend to produce speaker factor estimates that are *comparable* but not necessarily *identical*. This remaining distance is essentially a measure of the error between the speaker factor estimates produced by the individual classes or Gaussians. In our case, the discrepancy between these estimates is potentially a source of information to exploit.

4. SUPERFACTORS

We define *speaker superfactors* as a supervector of speaker factors. This idea is related to the discussion in the previous section. Considering the eigenvoices matrix as tied matrices across Gaussians, the different classes tend to produce different individual speaker factors. We propose that instead of using the regular – tied – estimate of the speaker factor, we should keep the individual estimates and build a supervector by concatenating them.

4.1. Construction of Gaussian-based Superfactors

If considered as a tied model, then each Gaussian (or cluster of Gaussians) can estimate its own "view" of y. Precisely, we would like to evaluate how many of these "local" estimates of speaker factors we should use to get the best performance. We know that the single estimate of speaker factor for the eigenvoice matrix is a blend of each estimate coming from each Gaussian.

For all the following experiments, we cluster the Gaussians to form our classes. We show experiments from one cluster (i.e., the original estimate of the speaker factors) to the number of Gaussian in the UBM, where each cluster is a Gaussian.

To estimate the speaker factors for each cluster, we first obtain an estimate for the channel factors on all clusters. To reach this goal, we compute the posterior estimate of the speaker and channel estimate at the same time as suggested in [9, 10], precisely

$$\begin{bmatrix} y \\ x \end{bmatrix} = L_{yx}^{-1} \begin{bmatrix} V^T \\ U^T \end{bmatrix} \mathbf{\Sigma}^{-1} \mathbf{F} \tag{4}$$

where L_{yx}^{-1} is the covariance of the posterior distribution and $\mathbf{F}$ is the first-order BW statistics centered around the UBM mean for the utterance. This first-order statistic is then channel compensated by centering with respect to the channel factor, called $\hat{\mathbf{F}}$. Then for each cluster of Gaussians c, we can obtain the speaker factor estimate y_c

$$y_c = L_c^{-1} V_c^T \mathbf{\Sigma}_c^{-1} \hat{\mathbf{F}}_c \tag{5}$$

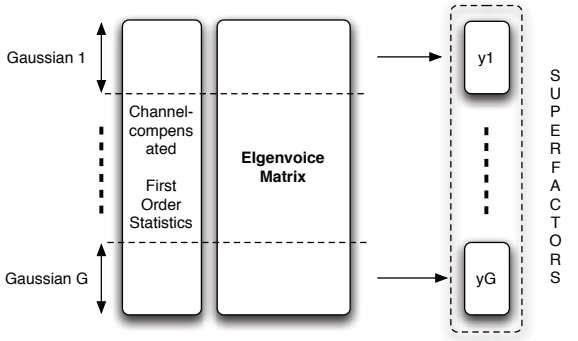

Fig. 2: *Construction of Gaussian-based speaker superfactors.*

where V_c, L_c^{-1}, and $\hat{\mathbf{F}}_c$ are the eigenvoice matrix, the covariance of the posterior distribution, and channel-compensated statistics related to the Gaussian in the cluster, respectively. The superfactors are the concatenation of each y_c, i.e., $y_{\text{super}} = [y_1^T \ldots y_C^T]^T$. This procedure is illustrated in Figure 2.

We perform experiments in order to pick the best method to estimate the covariance matrices. We report results on two different types of estimates. One is global, it is the covariance estimate for the standard speaker factor, making this covariance tied across clusters. The other, called local, uses only the counts from the Gaussians in the cluster, as described below:

$$L_c = \mathbf{I} + V_c^T \mathbf{\Sigma}^{-1} \mathbf{N}_c V_c \tag{6}$$

where $\mathbf{N}_c$ and V_c are the counts and the subspace relative to a particular cluster.

4.2. Results

Table 2 shows the results of this method obtained on NIST-SRE-2008 with Gaussian clusters formed using Gaussian indexes, e.g. for two splits the first cluster is composed of Gaussians 1 to 256, the second of Gaussians 257 to 512, and so on. The UBM was trained using an LBG-type algorithm such that the Gaussian indexes will somewhat reflect proximity of the Gaussians in feature space. Figure 3 illustrates the trend of the EER with respect to the number of splits and to the baseline system.

In these results, using only one split is strictly equivalent to using a cosine kernel between speaker factors, a result already found in the literature. However, when using superfactors, as the number of clusters grows, the performance of the system improves. Starting from using two clusters, using superfactors outperforms the simple cosine kernel on speaker factors. And using an optimal number of clusters permits reaching the performance of the baseline, with even a modest improvement. Moreover, while a global estimate of the L_c matrices works for a small number of clusters, using a localized estimate for each cluster seems to be critical, but comes at a higher computational cost. Compared to using only the speaker factor, superfactors give a relatively good performance without applying any score normalization technique. This is helpful when optimizing a function, for instance when using a kernel function or an optimization routine, as well as in other tasks, such as language identification.

There is, however, a lower bound that seems to be around 32 clusters resulting in 16 Gaussians per cluster. Using more clusters results in a degradation of performance. There are several reasons explaining the presence of this lower bound. The first one is that as the number of Gaussians in the cluster decrease, the data needed to estimate the L_c matrix becomes too sparse. Second, estimating

Table 2: *Superfactors on a varying number of Gaussian clusters. EER results on NIST-SRE-2008, Conditions 6 and 7 (resp. telephone and telephone English only). The second column indicates the superfactor dimension for each experiment.*

# Clusters	Superfactor dimension	Nonorm (EER) Cond 6/7	Ztnorm (EER) Cond 6/7
1	300	10.75/6.66	10.01/5.54
2	600	10.05/5.86	9.93/5.52
4	1200	9.11/5.39	8.78/4.88
8	2400	8.63/4.80	8.22/4.32
16	4800	8.51/4.79	7.66/3.83
32	9600	8.37/4.65	7.58/3.58
64	19200	8.85/5.06	7.73/3.83
128	38400	9.71/5.70	7.73/4.06
256	76800	10.16/6.35	7.80/4.07
512	153600	10.42/6.66	7.77/3.99

speaker factors per cluster partially removes the correlation information in order to expose it to the classifier (in our case a dot product). While this seems to perform well for a small number of clusters, decorrelating each local speaker factor from another will inevitably result in reduced accuracy of the resulting y estimates.

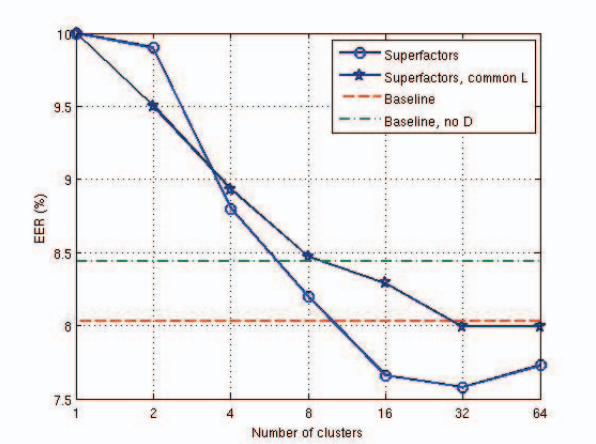

Fig. 3: *Superfactor system performance compared to the baseline system (with and without the diagonal component). Superfactors computed with their respective covariance reached baseline performance using 16 clusters.*

One interesting interpretation of the superfactors is that each y_c vector from each Gaussian or cluster brings information about part of the data (where each frame belongs to each Gaussian proportionally to its occupation). Thus we argue that the variability reflected in the differences between the y_c estimates for an utterance is one possible way of modeling within-session variability. More work needs to be done to be able to model that distribution, and this should guide us in building more efficient clusters based on Gaussians.

5. CONCLUSIONS

We propose an innovative method to characterize a speaker in the JFA framework. While the baseline system usually recreates a GMM supervector, new techniques are using the latent space to perform speaker verification. We propose to go further by building superfactors, i.e., vectors composed of several estimates of speaker factors from clusters of Gaussians. We investigate this framework by analyzing the Tied Factor Analysis system where the different estimates of the factors never converge to a single estimate, resulting in a blended estimate where information is lost. We propose to interpret the current JFA system as a Tied Factor Analysis system where the classes are clusters of Gaussians. In this approach, we keep the speaker factor estimate of each cluster to build a superfactor vector that will be used for speaker verification. We show improvement over using a single tied speaker factor estimate and also improvement over the standard baseline system. There are two main trends for continuing this work. The first will be to produce more relevant clusters of Gaussians for this model. We believe that with more work to understand the model, we can eventually get rid of the lower bound in the number of clusters. The second will be to investigate how to use the variance of the local factors and relate it to within-session variability modeling.

6. ACKNOWLEDGMENTS

The authors thank the Johns Hopkins University for the 2008 summer workshop, as part of this work was accomplished during that time. The research by authors at SRI International was funded through a development contract with Sandia National Laboratories (#DE-AC04-94AL85000). The views herein are those of the authors and do not necessarily represent the views of the funding agencies. The research by authors at Queensland University of Technology was supported by the Australian Research Council Discovery Grant No. DP0877835.

7. REFERENCES

[1] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, 2005.

[2] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny, "Comparison of scoring methods used in speaker recognition with joint factor analysis," *Proceedings of ICASSP 2009*, 2009.

[3] N. Dehak, P. Kenny, R. Dehak, O. Glembek, P. Dumouchel, L. Burget, H. Valiantsina, and F. Castaldo, "Support vector machines and joint factor analysis for speaker verification," *Proceedings of ICASSP 2009*, 2009.

[4] R. Vogt and S. Sridharan, "Experiments in session variability modeling for speaker verification," *Proceedings of ICASSP 2006*, 2006.

[5] N. Brümmer, "SUN SDV system description for the NIST SRE 2008 evaluation," 2008.

[6] "Johns Hopkins University, Summer Workshop, Robust Speaker ID, Fast Scoring Team," 2008, Baltimore, MD.

[7] S. Prince and J. Elder, "Tied factor analysis for face recognition across large pose changes," *Proceedings of the British Machine Vision Conference*, vol. 3, pp. 889–898, 2006.

[8] N. Scheffer, R. Vogt, S. Kajarekar, and J. Pelecanos, "Combination strategies for a factor analysis phone-conditioned speaker verification system," *Proceedings of ICASSP 2009*, 2009.

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