COMBINATION STRATEGIES FOR A FACTOR ANALYSIS PHONE-CONDITIONED SPEAKER VERIFICATION SYSTEM

Nicolas Scheffer, Robbie Vogt†, Sachin Kajarekar, Jason Pelecanos‡

SRI International, Queensland University of Technology†, IBM T.J. Watson Research Center‡

ABSTRACT

This work aims to take advantage of recent developments in Joint Factor Analysis (JFA) in the context of a phonetically conditioned GMM speaker verification system. Previous work has shown performance advantages through phonetic conditioning, but this has not been shown to date with the JFA framework. Our focus is particularly on strategies for combining the phone-conditioned systems. We show that the classic fusion of the scores is suboptimal when using multiple GMM systems. We investigate several combination strategies in the model space, and demonstrate improvement over score-level combination as well as over a non-phonetic baseline system. This work was conducted during the 2008 CLSP Workshop at Johns Hopkins University.

Index Terms— joint factor analysis, robust speaker ID, phonetic GMM, JHU workshop.

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 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.

This work aims to take advantage of developments in JFA in the context of a phonetically conditioned system. Previous work with phonetic systems has shown the ability to extract additional performance through phonetic conditioning [2, 3], although this advantage was not observed for a full factor analysis model.

The particular focus of this work is to investigate strategies for combining each of the phone-conditioned JFA systems. Our hypothesis is that score level combination is suboptimal and does not fully realize the potential advantages of a conditioned JFA system. Options for model-level combination are presented and compared.

We term the model combination strategies as *supervector concatenation* and *subspace stacking*, both illustrated in Figure 1. The motivation behind the supervector concatenation approach is to simultaneously present all the phone-conditioned statistics to the JFA model so that correlations and relationships between the phonetic conditions, as well as the differences, can be observed and modeled. This approach results in an increase in the dimension of the speaker model mean by a factor of the number of phonetic classes with no increase in the latent variable dimension.

Alternatively, the subspace stacking approach combines subspace transforms from each phonetic context resulting in an increased dimension of the speaker, channel or both latent variables. It is hypothesized that this approach provides the flexibility for the observed data to select the most relevant subspace dimensions and has previously proven useful in the auxiliary microphone conditions of recent NIST SREs [4].

While the focus of this work is on phone-conditioned JFA systems, the implications may reach beyond this scope. We expect that investigating several possibilities using phonetic-events will lead to a better understanding of the JFA model and a methodology that can be applied to increase robustness to other kinds of conditions such as language, gender and microphone types.

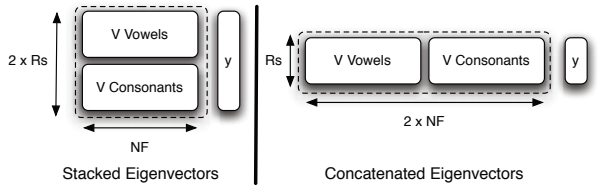

Fig. 1. *Stacked vs. concatenated eigenvectors for 2 phonetic classes. The former enrich the model by projecting statistics on both classes, thus increasing the rank. The latter produces a more robust latent variable by tying the classes together, thus increasing the model size.*

2. SYSTEMS AND PROTOCOL

We describe the JFA framework, as well as the system and the phonetic decoder used for the experiments, before presenting the experimental protocol.

2.1. Joint Factor Analysis

Let us define the notations that will be used throughout this discussion. The JFA framework uses the distribution of an underlying GMM, the universal background model (UBM) of mean m_0 and diagonal covariance Σ_0. Let the number of Gaussians of this model be N and the feature dimension in each Gaussian be F. A supervector is a vector of the concatenation of the means of a GMM: its dimension is NF. 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,

$$m_s = m_0 + Vy + Dz \tag{1}$$

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

low-dimensional subspace of rank R_C. The supervector for an utterance h with speaker s is

$$m_h = m_s + Ux \quad (2)$$

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

2.2. Baseline System Description

The speaker recognition system from Brno University of Technology (BUT) [5] is used for the experiments. The baseline system employs a 512-Gaussian UBM. The features are warped Mel-frequency cepstral coefficients (MFCCs) composed of 19 cepstrum features and one energy feature. First and second order derivatives are appended for a total dimension of 60. The rank of the speaker space is 120 while the channel space rank is 60. A lower number of Gaussian as well as lower subspace ranks were selected to accommodate for the multiple phone classes.

To train the matrices, 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 a $N(0,1)$ 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 [6]).

The verification score for each trial was a scalar product between the speaker model mean offset and the channel compensated first order Baum-Welch statistics centered around the UBM. This scalar product was found to be simple yet very effective [7] and was subsequently adopted by the JHU fast scoring group [8].

The speaker verification system is gender-independent with a gender-dependent score normalization (ZT norm).

2.3. Phonetic Decoder

The phonetic decoder used for these experiments is an open-loop Hungarian phone decoder from BUT, Brno [9]. The Hungarian language possesses a large phone set and enables the modeling of more nuances than an English set. This has been particularly useful in language identification tasks. For this work, we chose to cluster the phonemes into broader phonetic events. We used two different clusterings obtained in a supervised way by expertise:

- 2-class set: vowels (V), consonants (C)
- 4-class set: vowels (V), sibilants (Si), stops (St), non-vowels (NV).

To build a phonetically conditioned system, for example a vowel system, we first extract the feature vectors from an utterance corresponding to the occurrences of vowels in the phone transcription to obtain phone-conditioned Baum-Welch statistics for the utterance. These statistics are used in exactly the same fashion as described above to build a full JFA model with phone-conditioned speaker and channel subspace matrices. The speaker and channel loadings will be subscripted by the notation adopted for each event in Table 1 (for instance, V_V will be the speaker loading for the vowel set).

2.4. Experimental Protocol

All experiments were performed based on the all trials condition from the NIST-SRE-2006 dataset. The data set consists of 3616 target trials and 47452 non-target trials. Results are given in terms of equal error rate (EER) and minimum detection cost function (mDCF) given by NIST.

The factor analysis model uses the following data sets for training:

- The UBM is trained on Switchboard and Mixer data. For simplicity we fixed the UBM for all phonetic events.
- The eigenvoices and eigenchannels are trained in a gender-independent fashion on the NIST SRE 04 data set, consisting of 304 speakers and 4353 sessions. The diagonal model is trained on 359 utterances coming from 57 speakers from SRE 04 and 05.
- The score normalization data (Z- and Tnorm) was drawn from SRE 04 and 05 with around 300 utterances for each gender.

3. COMBINATION STRATEGIES

In this section, we evaluate the performance of the score-level combination strategy for the phonetic-system. We will then investigate techniques in the model space that will robustly estimate the speaker by taking into account all phonetic classes.

3.1. Baseline and Score-level Fusion Results

Score-level combination is a frequently used technique for gaining robustness on different conditions. For a phonetic GMM system, the usual strategy is to have as many systems as the number of phonetic events. The combination of information is done at the score level by fusing the scores. In this experiment, an optimistic system combination is used, as the logistic regression is trained and tested on the same data. The FoCal toolkit [10] is used for this process.

Table 1. *Results for the baseline system, as well as for each phonetic group are included. The results of fusions across phonetic groupings are also shown. Results show that score-level combinations for the two phonetic sets are similar, but fail to outperform the baseline. [SRE 06, all trials, DCF×10, EER(%)].*

System	% Data	EER (%)	mDCF
Vowels (V)	60	6.17	0.296
Consonants (C)	40	7.91	0.391
Consonant subsets...			
Non-Vowels (NV)	15	10.7	0.502
Sibilants (Si)	15	14.14	0.647
Stops (St)	10	15.27	0.685
V+ C	100	5.20	0.262
V + NV + Si + S.	100	5.42	0.272
Baseline	100	5.12	0.241

Table 1 presents the results for the baseline system, as well as for each broad phonetic event of our set. There is a clear advantage of the system using vowels alone, but it also represents 60% of the entire data used. The score-level fusion on the 2-class is better than for the 4-class set. However, while using the same amount of data, the 2-class fusion performance is worse than the baseline system. In the following paragraphs, we show how to improve the subsystem combination.

3.2. Concatenation

The first model space approach investigated consists of concatenating parameters of the speaker from different phone sets. The following experiments investigate at which level this concatenation should occur. Let us consider the 2-class phone-set $\{V, C\}$ for this approach. The resulting model supervector length will thus increase

to $2NF$. The main advantage of this method is that a single system is used for the entire phone set.

3.2.1. Eigenvector concatenation

We first concatenate the eigenvectors from different phonetic events during training and testing of the speaker models. Under this model, the system will estimate a single set of latent variables x, y, z per utterance, each of them being independent of the class.

$$m_s = \begin{pmatrix} m_0 \\ m_0 \end{pmatrix} + \begin{pmatrix} V_V \\ V_C \end{pmatrix} y + \begin{pmatrix} U_V \\ U_C \end{pmatrix} x + \begin{pmatrix} D_G & 0 \\ 0 & D_G \end{pmatrix} z \quad (3)$$

Here, the ranks of the subspaces are the same as in the baseline system and the D_G matrix is a copy of the D matrix from the baseline system.

The results in Table 2 (first three rows) show a significant degradation of the model concatenation style combination. It seems that if the subspaces are trained separately, the projection on the resulting concatenated subspace does not reflect the classes appropriately. This leads to the need to retrain subspaces explicitly to be tied together. It is important to note that the concatenation of the channel eigenvectors decreases the performance much more compared to the speaker eigenvectors. This supports the hypothesis that eigenvoices should be the main focus when using a phonetic GMM system.

3.2.2. Baum-Welch statistics concatenation

For this experiment, the speaker and channel subspaces are retrained using the concatenated first- and zero-order statistics from each phonetic event. The results in Table 2 show that this approach performs close to the score-level combination, but fails to outperform it. However, the subspaces are effectively tied so that a robust estimate of the latent variable can be produced. Consequently, a gain is observed compared to the systems taken separately.

3.2.3. Tied factor analysis

Tied factor analysis has been used successfully in other fields such as face recognition [11]. For this approach, the model is the same as in Equation 3, but the eigenvectors for each phonetic event are trained so that the latent variables are tied between the phonetic events. This approach should be successful for a phonetic system, as the amount of data for each event can vary, especially for very short conditions. We applied the following algorithm until convergence:

- Estimate the latent variables for the concatenated Baum-Welch statistics (like in 3.2.2).
- Estimate the matrices separately, on their respective statistics, by maximizing the likelihood of the data with respect to the latent variables of the previous step.

Table 2 shows that retraining the subspaces by concatenating the statistics from each phone set or by using tied factor analysis leads to similar performance. It seems the EM algorithm used for the factor analysis model tends to tie the different phonetic events naturally.

3.3. Stacking

Another approach in the model space consists in stacking the eigenvectors of the subspaces together. In this approach, the dimension of the model remains constant while the rank of the subspaces increases. This leads to running one system per event before combining them at the score-level.

Table 2. *Eigenvector concatenation on the 2-class set. The speaker and the channel subspace used are shown along with the concatenation type. Results show that the subspaces have to be retrained to obtain decent performance, using the standard EM or a Tied Factor Analysis approach. [SRE 06, all trials, DCF×10, EER(%)]*

System	Speaker	Channel	EER (%)	mDCF
Baseline	V_G	U_G	5.12	0.241
Eig. Concat.	V_V, V_C	U_V, U_C	13.4	0.573
Eig. Concat.	V_G	U_V, U_C	11.3	0.531
Eig. Concat.	V_V, V_C	U_G	7.02	0.378
BW Concat.	V_V, V_C	U_V, U_C	5.45	0.266
Tied FA	$\{V_V, V_C\}_{Tied}$	U_V, U_C	5.32	0.268

3.3.1. Eigenvector stacking

The advantage of this method is its robustness to different stacking configurations. Indeed, the latent variable estimation is enriched with the information of other events while keeping a good estimate for the current event. Let us consider two matrices from the 2-class phone set V_V and V_C, and their respective latent variables y_v, y_c. This approach captures cross-correlation between phonetic events when estimating the latent components. Stacking the eigenvectors for different events is equivalent to performing a sum in the supervector space. For the 2-class set, the system is expressed as:

$$m_h = m_0 + \begin{pmatrix} V_V & V_C \end{pmatrix} \begin{pmatrix} y_V \\ y_C \end{pmatrix} + \begin{pmatrix} U_V & U_C \end{pmatrix} \begin{pmatrix} x_V \\ x_C \end{pmatrix} + D_G z \quad (4)$$

The D_G matrix is the one from the baseline system. The ranks of the resulting stacked matrices are 240 and 120, for the speaker and the channel respectively.

3.3.2. Stacking in the speaker space and channel space

Stacking the channel eigenvectors was already demonstrated to be successful for a different set of microphones [4]. Stacking the speaker eigenvectors should be suitable for a phonetic GMM system for two reasons. Firstly, speaker modeling should profit from correlations between phonetic events. Secondly, using subspaces from all phonetic events when evaluating a single phonetic event should increase robustness to errors of the phonetic decoder.

Table 3. *System combination using stacked eigenvectors for the speaker space, channel space or both. The matrices selected in each configuration are specified. Results tend to show that the relevant information is contained in the speaker space, as stacking the speaker loadings gives better results than the score-level fusion. [SRE 06, all trials, DCF×10, EER(%)]*

System	Speaker	Channel	EER	mDCF
Baseline	V_G	U_G	5.12	0.241
Unstacked	V_V, V_C	U_V, U_C	5.20	0.262
Stacked	V_G	U_V, U_C	5.34	0.260
Stacked	V_V, V_C	U_G	5.09	0.247
Stacked	V_V, V_C	U_V, U_C	5.28	0.251
Stacked	$V_V, V_{St}, V_{Si}, V_{NV}$	U_G	5.03	0.250

Similarly to the concatenation experiments, results in Table 3 tend to show that the relevant information is contained in the speaker space as stacking in the channel space degrades the results. This means that a global channel matrix can be estimated and successfully applied to all events. Therefore, we only present this configuration

for the 4-class set. Stacking the speaker eigenvectors is a strategy that outperforms the score-level combination and gives the results similar to the baseline non-phonetic system. There is no observed improvement by using the 4-class set over the 2-class one.

3.3.3. Stacked eigenvoices for the baseline system

In section 3.3, we showed that stacking the matrices for each phonetic event was a successful approach for a phonetic-based system. One disadvantage of this method, compared to the method of Section 3.2, is the need to run one system for each event.
The phonetic subspaces can, however, be used to generate large factor loading matrices. In the protocol, around 300 speakers are used to train the eigenvoice matrix. This is also the maximum number of eigenvoices that can be estimated. For the 4-class phone set, the system has a rank of 480 for the speaker space. This number of eigenvectors cannot be estimated from our data set. However, it is interesting to use this large eigenvoice matrix for the baseline non-phonetic system (channel matrices are not used here following the results in Table 3). Under this scenario, the standard (non-phonetic) statistics will be presented to the system while the stacked matrices coming from different phonetic events are used as eigenvoices. The channel matrix used is the one from the baseline system.

Table 4. *Performance of the stacked eigenvoices generated from different phonetic events on a non-phonetic system. Stacked eigenvoices from the 4-class set outperform the baseline. [SRE 06, all trials, DCF×10, EER(%)]*

System	Speaker	EER (%)	mDCF
Baseline	V_G	5.12	0.241
Stacked	V_V, V_C	5.14	0.243
Stacked	$V_V, V_{NV}, V_{St}, V_{Si}$	4.76	0.234

Results in Table 4 show that stacking eigenvoices derived from different phonetic events can be useful for improving performance over the standard baseline system. It may also be that using more classes may better the performance of the stacked system. Indeed, using the stacked eigenvoices from the 4-class set outperforms the baseline non-phonetic system and the 2-class system.

4. CONCLUSION

This work aims to take advantage of the recent developments in Joint Factor Analysis in the context of a phonetically conditioned GMM speaker verification system. We focused on strategies for combining the phone-conditioned systems. Our first approach was to perform JFA per class and combine the systems at the score-level. Our hypothesis is that this approach does not use the data efficiently as the performance is worse than the baseline. We later employed strategies in the model space that more robustly estimate the latent variables by taking into account all phonetic events. In section 3.2, we showed that the concatenation of eigenvectors could lead to decent performance provided that the subspaces are explicitly retrained on the concatenated statistics. In section 3.3, we showed that both factor concatenation and score-level fusion could be outperformed by stacking eigenvectors from different phonetic events. For the phonetic system, stacking the eigenvoices leads to the greatest improvement. We also proposed to use this large set of eigenvoices on the baseline system and showed that it could result in a slight improvement over the traditional baseline system.

While the focus of this work is on phone-conditioned JFA systems, the implications may lead to a better understanding of the JFA model and a methodology that can be applied to increase robustness to other kinds of conditions such as language, gender and microphones. Future work will focus on understanding the differences and overlaps between the global and per-class estimates, in the channel and the speaker space, and methods to extract more information for a more robust estimate of speaker models.

5. ACKNOWLEDGMENTS

The authors thank JHU for the summer workshop of 2008 as well as all the Robust Speaker ID group. We particularly thank Lukas Burget and Ondrej Glembek from Brno University of Technology[1] for their tireless technical support and openness during the workshop that made this work possible.

The work, 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.

6. 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] S. Kajarekar, "Phone-based cepstral polynomial SVM system for speaker recognition," *Proceedings of Interspeech 2008*, 2008.

[3] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, "Compensation of nuisance factors for speaker and language recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 1969–1978, 2007.

[4] 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.

[5] L. Burget, P. Matejka, and O. Glembek, "BUT system description for the NIST SRE 2008 evaluation," 2008, Montreal, Canada.

[6] R. Vogt, B. Baker, and S. Sridharan, "Modelling session variability in text-independent speaker verification," in *Ninth European Conference on Speech Communication and Technology*. ISCA, 2005.

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

[8] "Johns Hopkins University, Summer workshop, Robust Speaker ID, Fast scoring team," 2008, Baltimore, MD.

[9] P. Matejka, L. Burget, P. Schwarz, and J. Cernocky, "Brno University of Technology System for NIST 2005 Language Recognition Evaluation," *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006*, pp. 1–7, 2006.

[10] N. Brümmer and J. du Preez, "Application-independent evaluation of speaker detection," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 230–275, 2006.

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

[1] http://www.fit.vutbr.cz/