Impact of Prior Channel Information for Speaker Identification

C. Vaquero[1,2,⋆], N. Scheffer[2], and S. Karajekar[2]

[1] University of Zaragoza
Maria de Luna 1, 50018 Zaragoza, Spain
cvaquero@unizar.es
[2] SRI International,
333 Ravenswood Avenue, Menlo Park, CA 94025-3493, USA
nicolas.scheffer@sri.com,
sachin@speech.sri.com

Abstract. Joint factor analysis (JFA) has been very successful in speaker recognition but its success depends on the choice of development data. In this work, we apply JFA to a very diverse set of recording conditions and conversation modes in NIST 2008 SRE, showing that having channel matched development data will give improvements of about 50% in terms of Equal Error Rate against a Maximum a Posteriori (MAP) system, while not having it will not give significant improvement. To provide robustness to the system, we estimate eigenchannels in two ways. First, we estimate the eigenchannels separately for each condition and stack them. Second, we pool all the relevant development data and obtain a single estimate. Both techniques show good performance, but the former leads to lower performance when working with low-dimension channel subspaces, due to the correlation between those subspaces.

1 Introduction

Cepstral features with the Gaussian mixture model (GMM) is a very commonly used configuration for a speaker recognition system in NIST speaker recognition evaluations. Channel mismatch is a major problem in performance degradation and the most successful approach to addressing this problem has been joint factor analysis (JFA) [1]. JFA models the variability in the features into speaker and channel variability. As with any other statistical modeling technique, the choice of development data used for JFA is crucial for the best performance.

In this work we explore different approaches to estimating channel variability, testing them with NIST 2008 SRE data. The data contains telephone conversations recorded over telephone and over different microphones as well as interviews recorded over different types of microphones. There is a lot of development data for telephone conversations, but there is very limited data for interviews. In

⋆ This work has been supported in part by the program FPU from MEC of the Spanish Government.

M. Tistarelli and M.S. Nixon (Eds.): ICB 2009, LNCS 5558, pp. 443–453, 2009.
© Springer-Verlag Berlin Heidelberg 2009

addition, the microphones used in telephone conversations and interviews are similar, so the development data for the former can be reused for the latter.

In our first approach we divide the development data into different sets, each belonging to a particular microphone and communication mode. JFA is performed separately for each set and the resulting estimates are stacked as a single estimate of channel variability.

In our second approach we use all the development data to obtain a single estimate of channel variability. Both approaches work well when using a dimension high enough to model all channel subspace variability, but with a lower dimension the single estimate obtains better results, since the first approach shows correlation between different channel subspaces. Further analysis on correlation is done to show the importance of selecting the dimension for every channel subspace when stacking channel estimates.

2 System and Experimental Protocol

We describe the JFA system and the experimental protocol.

2.1 Joint Factor Analysis System

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 be F. A supervector is a vector of the concatenation of the means of a GMM: its dimension is $N \times F$. 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), that spans a subspace of low-rank R_S. D is the diagonal matrix 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, 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.

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 [2]).

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 [3] and was subsequently adopted by the JHU fast scoring group [4].

2.2 System Description

We have used a gender-dependent JFA system for all experiments. The UBM consists of a GMM of 512 gaussians trained on telephone conversations of about 3 minutes duration each from the NIST SRE04 database [5]. The male UBM was trained on 1970 conversations, while the female UBM was trained on 2741 conversations. A 300-rank eigenvoice matrix was trained to model the speaker subspace, using telephone conversations from NIST SRE04 and Switchboard II databases. Male speaker subspace was trained on recordings from 1368 speakers, while female speaker subspace was trained on recordings from 992 speakers.

For training channel subspace, three different datasets were considered: for tackling conditions involving telephone data, 1675 conversations from 118 male speakers and 2409 conversations from 183 female speakers from the NIST SRE04 were used. For dealing with conditions involving microphone data, 1207 and 1414 telephone conversations from 44 male and 52 female speakers, respectively, recorded over different microphones from the alternate microphone data of the NIST SRE05 were used. To have some prior information of interview data type, the interview development data from NIST SRE08 was used. This dataset contains long interview recordings over several microphones from three male (138 recordings) and three female speakers (141 recordings).

For scoring normalization zt-norm was used. We used 207 male and 292 female speaker models from the NIST SRE04 database for t-norm and 1374 male and 1770 female speaker segments for z-norm from the same database.

2.3 Experimental Protocol

Results are reported on the NIST SRE08 required condition, short2-short3. This condition takes just one session of the target speaker for enrollment and one session for testing. Short2-short3 is divided into several conditions, and we are interested in four of them:

- Two conditions using interview sessions for enrollment:
  - Condition 1, involving 34181 microphone channel matched trials, testing on interview sessions
  - Condition 4, involving 11741 microphone-telephone trials, testing on telephone calls

– Two conditions using telephone calls for enrollment:
  - Condition 5, involving 8454 telephone-microphone trials, testing on telephone calls recorded over different microphones.
  - Condition 6, involving 35896 telephone channel matched trials, testing on telephone calls.

Results are reported in terms of both equal error rate (EER) and detection cost function (DCF) as described in NIST SRE.

3 Channel Compensation Using Different Types of Data

We investigate the importance of having matching data to perform channel compensation in a JFA system.

3.1 Baseline System without Channel Compensation

To analyze the improvement obtained in performance when using channel compensation, we consider as baseline the JFA system without channel compensation. For this purpose, a classic Maximum a Posteriori (MAP) system was used.

The results for interview and telephone conditions are shown in Table 1 where, for two reasons, the former are worse than the latter. First, the interview condition uses microphones that are different from telephone and second, the background data is made up entirely of telephone conversation.

Table 1. Results of a classic MAP on NIST-SRE-2008, where results with interview data (condition 1) are worse than telephone data (condition 6)

Train-Test (Condition)	DCF	EER
Interview-Interview (1)	0.639	16.898
Interview-Telephone (4)	0.793	19.276
Telephone-Interview (5)	0.404	11.549
Telephone-Telephone (6)	0.523	11.725

3.2 JFA with Telephone Data

Table 2 show the results for four conditions with JFA performed with telephone development data. The results show that condition 1, which has the biggest mismatch with the development data, improves by only about 5% and is almost insensitive to the rank of the eigenchannel matrix. On the other hand, condition 6, which is the most matched condition, improves by almost 40% and increased the rank of about 500 gives the best performance on this condition.

The mixed data conditions show an interesting trend. When telephone data is used in testing (condition 4), the improvement is about 15%. In addition, the effect of using more eigenchannels is negligible. When telephone data is used in training (condition 5), the improvement is more. It is around 30% with improvement in performance with as many as 200 eigenchannels. There is no improvement from adding more eigenchannels. We discuss the mixed results further in Section 5.

Table 2. Results for several ranks on all conditions, using telephone development data for training channel compensation. Best results according to a trade-off between performance and matrix rank are emphasized. DCF (x10)/EER(%).

Rank	Condition 1	Condition 4	Condition 5	Condition 6
50	0.609/15.771	0.824/16.742	0.348/8.560	0.356/7.506
100	**0.582/15.355**	0.827/17.104	0.309/7.609	**0.350/6.908**
200	0.579/15.572	**0.821/16.471**	**0.298/6.997**	0.347/6.834
300	0.578/15.607	0.821/17.195	0.305/7.201	0.337/6.796
500	0.576/15.797	0.821/17.376	0.311/7.337	0.336/6.684

3.3 JFA with Microphone Data

Table 3 shows the results for four conditions with JFA performed with microphone data. Note that this data is more matched to condition 1 than condition 6. The difference between condition 1 and the development data is that the former is an interview and the latter is a conversation. As expected, the performance of condition 1 improves by almost 75% and it gets better with higher rank. The performance of condition 6 improves only slightly and with smaller-rank eigenchannels.

Table 3. Results for several ranks on all conditions, using microphone development data for training channel compensation. Best results according to a trade-off between performance and matrix rank are emphasized. DCF (x10)/EER(%).

Rank	Condition 1	Condition 4	Condition 5	Condition 6
50	0.341/6.352	0.652/13.394	0.346/8.288	0.537/11.202
100	0.319/5.789	0.612/13.032	0.321/8.084	0.536/11.016
200	**0.278/5.199**	**0.582/11.493**	0.302/7.473	**0.521/10.680**
300	0.269/5.017	0.575/11.222	0.290/7.745	0.52/10.792
500	0.270/4.991	0.568/10.679	**0.279/7.405**	0.518/10.754

The same trend is seen with mixed conditions. This trend is opposite of the trend seen with JFA using only telephone data (Table 2). The results show that there is about a 50% improvement in the performance when interview data is used in training (condition 4) with the results getting better with more eigenchannels. The latter can be also seen with condition 5 (interview data used in testing only), but the gains are about 30%. As mentioned earlier we will elaborate on the mixed conditions in Section 5.

3.4 JFA with Interview Data

Table 4 shows the results for four conditions with JFA performed with interview development data. Although it is matched to condition 1, it was very sparse. It had only six speakers (three male and three female) with different microphones. This is reflected in the experiments by the reduced rank of eigenchannels that

Table 4. Results for several ranks on all conditions, using interview development data for training channel compensation. Best results according to a trade-off between performance and matrix rank are emphasized. DCF (x10)/EER(%).

Rank	Condition 1	Condition 4	Condition 5	Condition 6
50	**0.446/9.194**	**0.728/15.656**	**0.428/10.734**	**0.558/12.360**
75	0.441/9.09	0.739/15.837	0.416/10.598	0.559/12.397
100	0.435/9.004	0.731/15.747	0.417/10.734	0.555/12.360
125	0.434/8.917	0.732/15.656	0.415/10.598	0.552/12.360

could be estimated from this data. Results are similar to Table 3; performance of condition 1 improves with increased rank. Performance of condition 6 is worse with these eigenchannels. Among mixed conditions, there is about a 20% improvement when interview data is used for training. When interview data is used for testing there is a very small improvement.

4 Importance of Matched Development Data

Some important conclusions can be made by comparing different results from the previous section. First is the importance of matched development data for getting the best performance. The results show that the best results on the telephone data are obtained with JFA trained with telephone data. It is the same with interview data where JFA trained with microphone and interview data gives better performance than JFA trained with telephone data. The results are not obvious in the sense that any statistical technique relies on the match between development and evaluation data. The results are interesting in the case of mismatch. These results show that JFA is not very effective with the mismatched development data.

Second, the mixed conditions (4 and 5) prefer matched development data for training condition. For example, when telephone development data is used, the best performance is obtained on condition 5. When microphone development data is used, the best performance is obtained on condition 4. The results on mixed conditions also show an interesting trend. These results improve with the addition of eigenchannels but only up to about 200 of them.

Third, note that interview data differs from telephone data in two aspects: recording microphones and communication mode. As mentioned earlier, the interview development data is very small compared to microphone development data. However, the results show that the microphone data is more representative of the interview condition than the interview data. One hypothesis about these results is that interview does not differ from conversation as a communication mode but the difference between microphones is what makes two conditions different. This effect was seen in FRTIV data collection [6]. It will be interesting to investigate further with this hypothesis.

5 Building a Robust System to Compensate for Unseen Channels

The JFA approach can take advantage of having different datasets in order to compensate for unseen channels as linear combinations of known channels. This can be achieved in two ways: it is possible to train different eigenchannel matrices on every dataset separately and then stack them as a single matrix, or channel compensation can be trained on the whole merged dataset.

5.1 Stacking Channel Matrices

Stacking channel matrices is an interesting approach to modeling different channels with one eigenchannel matrix, as it is a modular way to deal with many datasets. In this approach, one eigenchannel matrix is trained for every dataset, and all these matrices are stacked together. This approach is nondestructive and enables the model to select the dimension that best fits the data.

However, selecting for every database the dimension that best fits the data will lead us to obtain large eigenchannel matrices, so in order to avoid impacting the system speed when stacking many matrices, we choose for every data set that dimension obtaining best performance according to a trade-off between performance and matrix rank. Finally, we choose a 100-rank telephone (see Table 2), a 200-rank microphone (see Table 3) and a 50-rank interview eigenchannel matrix (see Table 4) to be stacked, and they will be stacked in two steps: first we will stack telephone and microphone matrices, obtaining a matrix of rank 300, and then we will add the interview matrix, obtaining a final rank of 350.

Table 5. Results obtained stacking eigenchannel matrices trained on three different datasets: telephone (phn), microphone (mic), interview (int). Results in terms of DCF(x10)/EER(%).

System trained on	Condition 1	Condition 4	Condition 5	Condition 6
Best single dataset	0.278/5.199	0.582/11.493	0.309/7.609	0.336/6.684
Phn, mic	0.264/5.277	0.483/9.321	0.203/4.959	0.339/6.311
Phn, mic, int	**0.211/4.263**	**0.460/9.050**	**0.198/4.775**	**0.332/6.263**

Table 5 shows results obtained for channel-matched conditions as channel matrices are stacked. For every condition, results for the best-performing system among the 100-rank telephone, 200-rank microphone and 50-rank interview are shown for comparison. We can see that stacking matrices does not decrease performance for matched conditions, while it brings a great improvement in mismatched conditions.

We can see as well that the interview data provides additional information, which helps to improve performance on conditions involving interview. Indeed, a 1% absolute gain is observed at the EER for condition 1. This highlights again the importance of having prior knowledge on the data for NIST evaluation and for realistic scenarios.

5.2 Retraining Channel Compensation

Retraining channel compensation is the easiest way to assure good performance across every channel, but it has an important disadvantage: it takes a lot of time to retrain an eigenchannel matrix. Indeed, matrices with very high rank should be trained to model all variability contained in several databases.

Table 6. Results obtained retraining channel compensation merging different datasets: telephone (phn), microphone (mic), interview (int). Results in terms of DCF(x10)/EER(%).

System trained on	Condition 1	Condition 4	Condition 5	Condition 6
Best single dataset	0.278/5.199	0.582/11.493	0.309/7.609	0.336/6.684
Phn, mic	0.248/5.442	0.476/9.683	**0.202**/5.095	0.334/6.161
Phn, mic, int	**0.212/4.255**	**0.444/9.321**	0.206/**4.823**	**0.330/6.049**

Table 6 shows results when retraining a 300-eigenchannel matrix using telephone and microphone datasets, and a 350-eigenchannel matrix on all data available. Comparing these results with those obtained in Table 5 we can see that it is possible to obtain similar performance by stacking eigenchannel matrices as by retraining a new eigenchannel matrix. Moreover, stacking matrices is a practical answer for enriching channel modeling without having to retrain on all data.

However, when low-rank matrices are used, stacking is not as good as retraining, especially on cross-channel conditions. This can be seen on Table 7, where results of stacking 50 phone, 50 microphone, and 50 interview eigenchannels are compared to those obtained by retraining 150 eigenchannels.These results show the importance of rank selection for every channel subspace before stacking, as has been done in this work.

This behavior is explained by the correlation between channel subspaces.

Table 7. Comparison between stacking and retraining for low dimension channel subspaces. Results in terms of DCF(x10)/EER(%).

Channel compensation	Condition 1	Condition 4	Condition 5	Condition 6
Stacking (all datasets)	0.229/4.835	0.501/10.226	0.226/5.571	**0.325**/6.647
Retraining (all datasets)	**0.228/4.645**	**0.463/9.140**	**0.212/4.891**	0.339/**6.497**

5.3 Correlation between Channel Subspaces

When stacking eigenchannel matrices, some of the vectors stacked may be correlated, thus providing redundant information as well as very high ranks. To analyze the correlation between channel subspaces, an orthonormal base of the subspace is estimated for every eigenchannel matrix. We call these subspaces *phone*, *microphone*, and *interview* subspace. We use Singular Value Decomposition (SVD), as every low-rank $m \times n$ matrix M can be represented as:

$$M = U\Sigma V^* \tag{3}$$

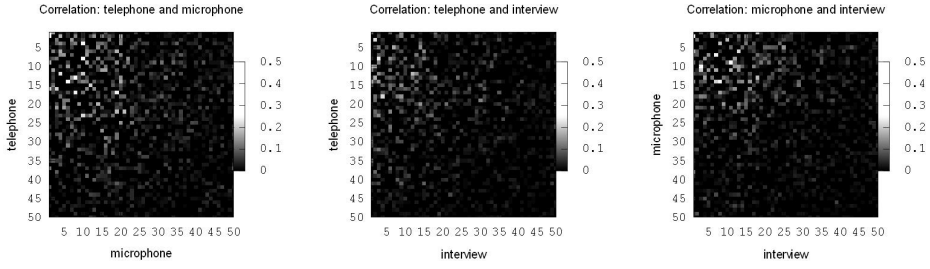

Fig. 1. Correlation between channel subspaces, trained on male data. The importance of the eigenvectors is ranked with respect to their singular values.

where U is an $m \times m$ unitary matrix containing a set of orthonormal basis vectors for the output space, Σ is a diagonal $m \times n$ matrix containing the singular values of the transformation defined by M, and V is an $n \times n$ matrix containing a set of orthonormal basis vectors for the input space.

Modeling the channel in this way allows us to get the correlation between channel matrices by just projecting one orthonormal channel matrix onto another. Figure 1 shows the correlation between phone, microphone, and interview subspaces, trained on male data, on a dimension reduced to 50. This correlation is higher between microphone and interview, and between telephone and microphone, while lower between telephone and interview. We can see that correlation appears mainly in those directions corresponding to the highest singular values.

When pooling all data together, the training method can take advantage of this correlation and model the channel directions more precisely. However, as more variability directions are modeled (i.e., rank is increased), the stacked approach tends to perform as well as the retraining approach.

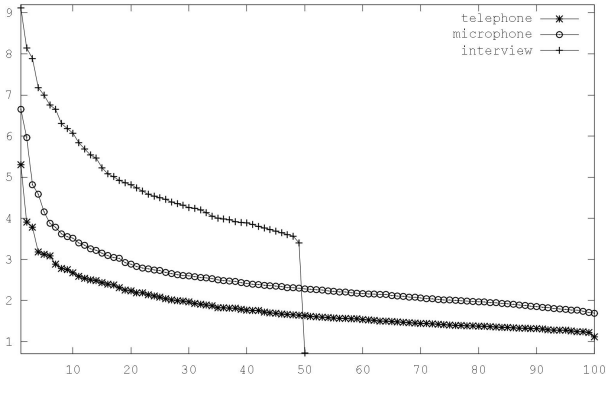

Fig. 2. First 100 singular values for every channel subspace, trained on male data. Interview and microphone subspaces shadow the telephone subspace.

5.4 Variability of Channel Subspaces

To remove correlation when stacking different matrices to obtain a low-rank stacked matrix, we can stack higher-rank matrices that are known to obtain good performance and then reduce the rank by means of SVD, keeping those directions showing higher variability (higher singular values).

However, analyzing Figure 2, which shows the first 100 singular values for every channel subspace, trained on male data, it can be noticed that microphone and interview channels have more variability, so that telephone variability directions will be shadowed and removed when reducing the rank, decreasing performance on those conditions involving telephone channel.

6 Conclusions

The problem of missing prior knowledge on the channels involved in the speaker verification task was studied. We showed the major impact of this information as without any information. A state-of-the art system would have the performance of a system not using channel compensation.

Then, we investigated two approaches for merging information from several datasets to build a more robust system. The first approach consists of stacking matrices trained with different datasets, which is a modular solution, allowing us to deal with multiple datasets with different types of recording for training channel compensation. The second approach consists of retraining the whole eigenchannel matrix.

The stacking approach was as good as the retraining approach when using high-rank eigenchannel matrices, but worse, especially on channel mismatched conditions, when using low-rank eigenchannel matrices. A further analysis has shown that this effect is probably due to the correlation present between different channel subspaces, correlation that the retraining approach may detect and take into account to obtain the final channel variability directions but that the stacking approach cannot get rid of.

Finally, a study on the singular values of telephone, microphone, and interview channel subspaces has shown that it is not possible to take advantage of the correlation for reducing high-rank stacked matrices to a desired rank. Directions from channels having high variability will shadow directions from channels showing less variability, losing performance in conditions having low variability.

As a final conclusion, we can assert that in order to assure performance as good for stacking matrices as for retraining channel compensation, a previous analysis on the optimal rank for every matrix to be stacked is needed. Stacking will lead to higher rank matrices than retraining, but it will assure good performance for every condition.

References

1. Kenny, P., Ouellet, P., Dehak, N., Gupta, V., Dumouchel, P.: A Study of Inter-Speaker Variability in Speaker Verification. IEEE Trans. Audio, Speech and Language Processing 16(5), 980–988 (2008)

2. Vogt, R., Baker, B., Sridharan, S.: Modelling session variability in text-independent speaker verification. In: Ninth European Conference on Speech Communication and Technology, ISCA (2005)
3. Brümmer, N.: SUN SDV system description for the NIST SRE 2008 evaluation, Montreal, Canada (2008)
4. JHU: Johns Hopkins University, Summer workshop, Robust Speaker ID, Fast scoring team, Baltimore, MD (2008)
5. NIST: The NIST year 2005 speaker recognition evaluation plan (April 2004), `http://www.nist.gov/speech/tests/spk/2004/SRE-04_evalplan-v1a.pdf`
6. Shriberg, E., Graciarena, M., Bratt, H., Kathol, A., Kajarekar, S., Jameel, H., Richey, C., Goodman, F.: Effects of Vocal Effort and Speaking Style on Text-Independent Speaker Verification. In: Proceedings of Interspeech, Brisbane, Australia (2008)