MULTICONDITION TRAINING OF GAUSSIAN PLDA MODELS IN I-VECTOR SPACE FOR NOISE AND REVERBERATION ROBUST SPEAKER RECOGNITION

Daniel Garcia-Romero, Xinhui Zhou and Carol Y. Espy-Wilson

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

ABSTRACT

We present a multicondition training strategy for Gaussian Probabilistic Linear Discriminant Analysis (PLDA) modeling of i-vector representations of speech utterances. The proposed approach uses a multicondition set to train a collection of individual subsystems that are tuned to specific conditions. A final verification score is obtained by combining the individual scores according to the posterior probability of each condition given the trial at hand. The performance of our approach is demonstrated on a subset of the interview data of NIST SRE 2010. Significant robustness to the adverse noise and reverberation conditions included in the multicondition training set are obtained. The system is also shown to generalize to unseen conditions.

Index Terms: *Robust speaker recognition, i-vector, multicondition training, PLDA.*

1. INTRODUCTION

Speaker recognition systems built in the lab with clean speech recordings can provide very high accuracies when tested on clean conditions. However, the performance rapidly degrades when the systems are used in the real world where channel/handset mismatch as well as environmental noise and reverberation are present [1]. The recently developed paradigm of i-vector extraction [2] provides an elegant way to obtain a low dimensional fixed-length representation of an entire speech utterance. The low-dimensional nature of the i-vector space facilitates the use of large amounts of data to remove/attenuate the effects of adverse conditions. Due to the emphasis that the series of NIST Speaker Recognition Evaluations [3] has placed on channel/handset mismatch, most of the focus on modeling i-vectors has been directed towards robustness to channel/handset mismatch. In this paper we focus on the use of generative models of i-vectors that are also robust to noise and reverberation.

Particularly, we are interested in situations where the recognition system is going to be deployed in an environment for which we can anticipate the most likely types of distortions that the system is going to be subjected to. Examples of this include systems deployed in cars, helicopters, office environments, etc. For these scenarios, a simple approach is to collect samples from the expected conditions (e.g., car noise at different speeds) and then create a multicondition dataset by electronically distorting the original clean data. This new augmented dataset can be used to train robust back-ends; as in [1] where multicondition training was successfully applied to a classical GMM-UBM architecture. In this paper we propose the use of multicondition training in the more advanced speaker representation based on i-vectors.

In the remainder of this paper we present a formal description of the proposed recognition architecture and analyze its behavior on a multicondition set created from a portion of NIST SRE 2010.

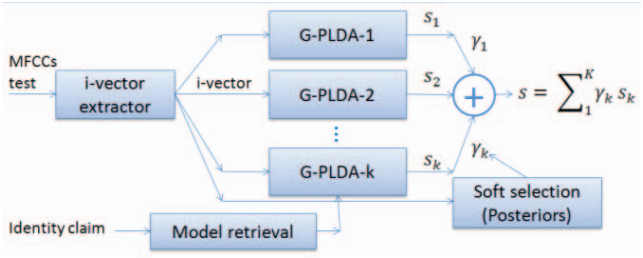

Figure 1. Mixture of K multicondition verification systems.

2. MULTICONDITION RECOGNITION SYSTEMS

All the multicondition training approaches proposed in this paper are based on a baseline state-of-the-art architecture that uses an i-vector extractor front-end followed by a Gaussian probabilistic generative model back-end [4]. In particular, as shown in Figure 1, we consider an extension in which the final verification score is a convex mixture of the scores produced by a collection of K susbsytems. Moreover, each of these subsystems is trained according to a multicondition training scheme.

In the following we provide the details of each of the three basic building blocks of our recognition system, namely: i-vector extraction, modeling and score computation.

2.1. I-vector extraction

An i-vector extractor [2] maps a sequence of cepstral coefficients from a speech utterance into a low-dimensional fixed-size representation $\eta \in \mathbb{R}^D$. This is accomplished by projecting a supervector of Baum-Welch statistics—collected with a Gaussian Mixture Model denoted as Universal Background Model (UBM)—into a low-dimensional subspace (i-vector space). The projection is learned from a large collection of data using a ML criterion.

2.2. Multicondition Gaussian PLDA (G-PLDA) modeling

The basic G-PLDA model ignores the abovementioned i-vector extraction process and considers the i-vector as an observed variable following the generative model [4]:

$$\eta = m + \Phi\beta + \epsilon. \quad (1)$$

In particular, m is a global offset; the columns of Φ provide a basis for the speaker-specific subspace (eigenvoices); β is a latent identity vector having a standard normal distribution; and ϵ is a noise term assumed to be Gaussian with zero mean and full covariance Σ. Maximum Likelihood point estimates of the model parameters $\{m, \Phi, \Sigma\}$ are obtained from a large collection of development data using an EM algorithm as in [5].

In a multicondition setup, we have access to K versions of the development data. Therefore, we can estimate a collection of hyperparameters $\{m_k, \Phi_k, \Sigma_k\}_1^K$. In the following, we present three alternatives to accomplish this.

2.2.1. Independent modeling

The easiest way to take advantage of the multicondition setup is to consider each condition k independent of each other and obtain hyperparameters as in the basic G-PLDA model [4].

2.2.2. Tied modeling

Another alternative, denoted as Tied-PLDA [5], assumes that for a subset of conditions (or even all conditions) the i-vectors of speaker j are generated using the same latent identity variable. However, the hyperparameters of each condition are different. This results in a new generative model of the form:

$$\eta' = m' + \Phi'\beta + \epsilon'. \quad (2)$$

where $\eta' = [\eta_1^T, \dots, \eta_K^T]^T$, $m' = [m_1^T, \dots, m_k^T]^T$, $\Phi' = [\Phi_1^T, \dots, \Phi_k^T]^T$ and $\epsilon' = [\epsilon_1^T, \dots, \epsilon_K^T]^T$. Note that the latent identity variable β is the same across all conditions (i.e., it ties the hyperparameters). By constraining the latent identity to be the same, the hyperparameter sets are not independent of each other and can leverage the data from all the tied conditions to obtain better estimates.

2.2.3. Pooled modeling

This approach assumes that a subset of (or all) conditions were generated from the same set of hyperparameters. Therefore, by simply pooling the data of those conditions together the hyperparameter learning stage is the same as for the basic G-PLDA model [5]. In this approach, sharing data across conditions is explicitly controlled by the pooling mechanism.

2.3. Verification score

As shown in Figure 1, a final verification score is obtained as a convex mixture of a collection of K scores $\{s_k\}$ according to the weights $\{\gamma_k\}$. The multicondition training scenario assumes that a speaker model is represented by K i-vectors $\{\eta_M^k\}$. Moreover, given a query test segment η_T, a score for each subsystem s_k can be computed as a likelihood ratio of two Gaussian distributions [5]:

$$s_k = p\left(\{\eta_M^k\}, \eta_T | k, \mathcal{H}_s\right) / p\left(\{\eta_M^k\}, \eta_T | k, \mathcal{H}_d\right) \quad (3)$$

where $\mathcal{H}_s$ and $\mathcal{H}_d$ represent the same- and different-speaker hypothesis respectively. The mean and covariance of these two distributions are defined by the hyperparameters $\{m_k, \Phi_k, \Sigma_k\}_1^K$. Also, the mixing weights $\{\gamma_k\}$ correspond to the posterior probability of each condition given the test i-vector and regardless of the claimed identity. They are therefore obtained by Bayes theorem as:

$$\gamma_k = \alpha_k \, p(\eta_T | k, \mathcal{H}_d) \; / \; \sum_j \alpha_j \, p(\eta_T | j, \mathcal{H}_d). \quad (4)$$

The prior probability of condition k is denoted by α_k. In all our experiments we consider each condition equally probable.

3. EXPERIMENTS

3.1. Experimental setup

All our experiments were conducted on the male part of condition 2 of the extended NIST SRE 2010 evaluation (i.e., interview data). Throughout the experiments we refer to this set as *evaluation data*. This subset comprises 1,108 models and 3,328 test segments from which 6,932 target trials and 1,215,586 non-target trials were obtained. Verification performance is reported in terms of Equal Error Rate (EER). We have used 400 dimensional i-vectors in all experiments. They were computed using a gender-dependent i-vector extractor trained from NIST SRE 04, 05, 06, 08-followup, Switchboard and Fisher. The necessary Baum-Welch sufficient statistics were collected using a diagonal-covariance gender-dependent UBM with 2048 mixtures trained on the same data. A subset of these data comprising 907 male speakers with a total of 10,695 files was used to train the hyperparameters of the PLDA models. We refer to this set as *development data*.

All speech files were parameterized using 38 MFCCs (i.e., 19 base coefficients without c0 plus deltas) obtained every 10 ms from a 20 ms Hamming window. Mean and variance normalization was applied to the entire utterances (i.e., not short time).

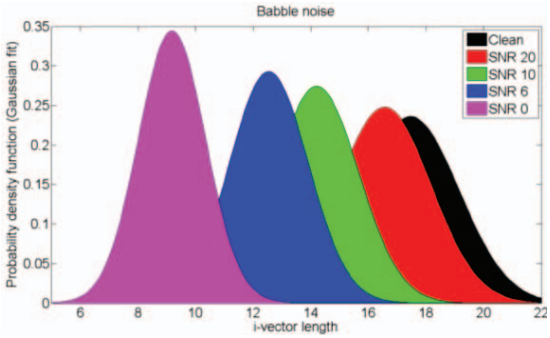

Figure 2. Gaussian fits to the distributions of i-vector lengths of the evaluation data for different SNRs in babble noise.

3.2. Multicondition data generation

In order to create a set for *multicondition* training and testing, 16 copies of each file from the evaluation and development sets were created by electronically adding 4 different types of noises: white, babble, car and helicopter at 20 dB, 10 dB, 6 dB and 0 dB SNRs. Additionally, 3 other copies with reverberation were created by convolving the original files with simulated impulse responses from rooms with reverberation times (i.e., RT30) of 100, 300 and 500 ms. Subsequently, i-vectors were computed for each file in the development and evaluation sets (i.e., original plus 19 corrupted versions). Note that neither the UBM nor the i-vector extractor were exposed to the noisy conditions since they were trained only on the original "clean" data.

During the experiments, the white noise subset was set aside to assess the behavior of our system to unanticipated conditions. Hence, the multicondition training set for our experiments was comprised of the clean data along with the other three noisy versions (babble, car, helicopter) and reverberation; a total of $K = 16$ conditions. For evaluation, we had access to the corrupted test segment in the expected deployment conditions (i.e., the training conditions) plus the unanticipated condition involving white noise.

3.3. I-vector length normalization

In [4] it was shown that the current strategy (e.g., [2]) used to extract i-vectors induces a severe mismatch between the length of the development and evaluation i-vectors. This was identified as a major source of non-Gaussian behavior. Moreover, a length normalization was proposed to reduce this mismatch and allow for effective Gaussian modeling. Here we further extend those observations by looking at the distribution of i-vector lengths of the evaluation data as a function of the SNR. Figure 2 shows the results of fitting Gaussians to the length distributions for different SNRs of babble noise. As a general trend we can observe that the lower the SNR the smaller the length and the dispersion of the data. The same relative trend happens for the development data and

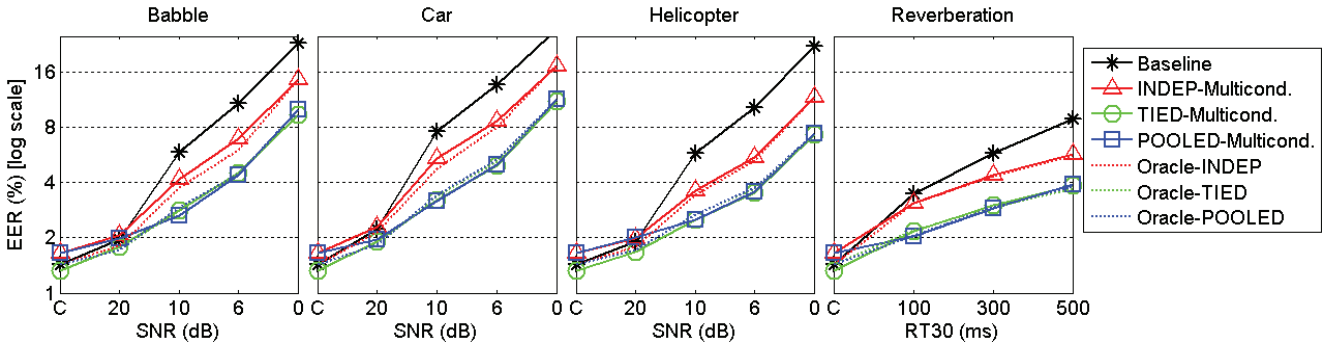

Figure 3. Performance of multicondition training approaches for different noise types and SNRs. (See section 3.5 for details)

other types of noise. Thus, when considering a collection of data with a wide range of SNRs, either heavy-tailed modeling is used (e.g., [6]) or Gaussian modeling must be preceded by a compensation technique such as length normalization. For this reason, all the experiments in this paper were carried out using length-normalized i-vectors.

3.4. Tying structure

In this section we explore the effects of different tying/pooling structures in the verification performance. That is, if we have development data for K conditions, we want to know how to best tie/pool them together to maximize performance. The reason to explore this is that tying/pooling conditions reflects our belief of how the data was generated. Note that the larger the number of conditions tied/pooled together, the more data we have to estimate the hyper-parameters. However, this comes at the risk of stronger assumptions about the data generating process (which may not be true and therefore can harm the performance).

Table 1 shows the verification performance averaged across babble, car and helicopter noise using three different tying structures (same trends are true for pooling approach). In all three structures, the performance was always better when individualized global offset hyper-parameters were used (i.e., no tying/pooling of the data for the $\{m_k\}_1^{K=16}$). Therefore, the difference among the three structures came from the way the speaker and channel subspaces were treated. In particular, for the "Pair" structure, a pairwise tying of each noisy condition with the clean data was used. For the "Block" case, the blocks of conditions from the same noise type were tied together and also with the clean data. Finally, the "All" structure tied together all 16 conditions, thus a unique pair of $(\Phi_{\text{ALL}}, \Sigma_{\text{ALL}})$ was computed and only the offsets $\{m_k\}_1^{K=16}$ were different.

The results indicate that the pairwise structure is the best for the high SNR conditions and very close to the "All" structure for the lower SNRs. Using a unique pair of $(\Phi_{\text{ALL}}, \Sigma_{\text{ALL}})$ seems detrimental for the clean and 20 dB conditions. The "Block" structure is still competitive but not as good as the other two. Even though the number of structures explored was not exhaustive, the results presented here provide a sense of the aforementioned tradeoff between leveraging large amounts of data and imposing suboptimal assumptions on the data.

3.5. Performance analysis

Figure 3 shows the performance of three multicondition training approaches—along with a baseline system trained only on clean data—for different noise types and SNRs. Also, the oracle counterparts are depicted (i.e., oracle selection of training condition that matches the test segment instead of using posterior probabilities). Only the results on the expected deployment conditions are presented here (deferring the analysis in unanticipated conditions to section 3.7). The baseline system performs very well in the clean condition (1.43% EER). However, the performance degrades rather quickly with small amounts of noise and reverberation (e.g., at 10dB the performance is between 6 and 8 times worse than in clean). The improved performance of the independently trained system (i.e., not tying or pooling) with respect to the baseline is solely due to the reduction of the mismatch between training and testing. Moreover, the performance improvement of the Tied- and Pooled-PLDA approaches—which are based on the pairwise tying/pooling structure described in the previous section—is much better than the one obtained by independently training K PLDA systems. Specifically, in the Tied-PLDA approach, forcing the latent identity variable to be the same (pairwise) for clean and each degraded condition allows the system to leverage the clean data to obtain better estimates of the speaker and channel subspaces in noisy situations. The Pooled-PLDA system also exhibits an impressive performance but it is slightly worse than the Tied-PLDA approach for the high SNRs. In summary, when using Tied- and Pooled-PLDA training for the anticipated noisy conditions, the performance improvement is between 2.5 and 3 times better than the baseline system and about 1.5 times better than the independent multicondition training.

3.6. Score combination based on posteriors

As described in section 2.3, the final score for a verification trial is obtained by combining the scores of each sub-system based on the posterior probability of the condition given the trial at hand. Therefore, the success of this approach heavily relies on the quality of these mixing coefficients.

Figure 4 depicts the average behavior of these coefficients for the pairwise Tied-PLDA approach. In particular, each column corresponds to a 16-dimensional vector of posterior probabilities (averaged across more than 1.2 million trials). Also, the magnitude of the coefficients—which adds up to one for each column—is proportional to the side of the gray squares. For example, the first column indicates that on average the scores for trials involving clean test segments are produced by an equal mixture of the scores

Tying structure	EER (%) averaged across test segment noise types				
	Clean	20 dB	10 dB	6 dB	0dB
Pair	**1.32**	**1.78**	2.85	4.32	9.23
Block	1.66	1.96	3.00	4.43	9.46
All	1.76	1.91	**2.80**	**4.23**	**9.08**

Table 1. Speaker verification performance averaged across babble, car and helicopter noise for 3 different tying structures.

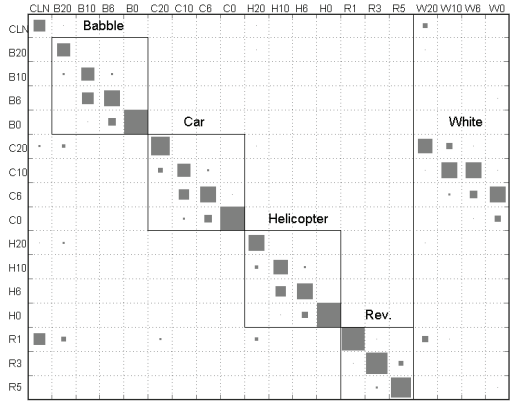

Figure 4. Average posterior probabilities for pairwise Tied-PLDA

form the clean and 100 ms reverberation sub-systems. In this particular case, the (averaged) equal weights are mostly due to the fact that 50% of the time either one of them is responsible for 100% of the final score (i.e., posterior equal 1). Thus, the posteriors are very sharp. For other cases, the actual mixing mass is spread across two or at most three sub-systems. However, the diagonal dominance of the posterior probability matrix (left part of Figure 4) along with its sparsity is a clear indication of the correct behavior of the proposed sub-system selection mechanism.

A more direct way to assess the quality of the selection mechanism is to compare the verification performance of the actual system with that of a system based on oracle selection (i.e., final score comes from the sub-system matching the test segment condition). Figure 3 facilitates this comparison. Particularly, the performance of the oracle and actual systems is almost indistinguishable. The individual multicondition approach tends to be a little bit worse than its oracle counterpart. However, for the Tied- and Pooled-PLDA cases the actual performance is always equal or even better than that of the oracle selection. As an example, for the clean condition, the Tied-PLDA system produces an EER of 1.32% whereas the oracle performance is 1.43%.

3.7. Unanticipated conditions

So far we have assessed the performance of multicondition training when the testing condition matches one of the development conditions. However, it is not realistic to assume that we will be able to anticipate all the potential situations that the system is going to encounter during deployment. Hence, it is important to know how the system is going to behave in those situations. Ideally, the system would approximate the test condition with the closest development condition and produce a score no worse than what the baseline system would produce. Therefore, the system will perform much better than the baseline in the expected conditions and not worse (or better) in unanticipated scenarios. This is exactly how our system behaves. The last four columns of Figure 4 show the average posteriors for test segments corrupted with white noise (recall that white noise was not included in the development set). As expected, the system selects the closest candidate (i.e., car noise resembles white noise much better than the remaining competitors). Also, the posteriors seem to concentrate around the correct SNR levels. More importantly, as shown in Table 2, the performance of any of the multicondition approaches outperforms the baseline system; with Tied- and Pooled-PLDA producing a significantly superior performance over the independent approach. However, the relative improvement for this unanticipated condition is not as good as the one observed for the expected conditions.

EER (%)	White noise SNR			
	20 dB	10 dB	6 dB	0 dB
Baseline	3.62	15.08	23.06	35.98
Indep. Multi.	3.29	12.56	19.01	32.37
Tied Multi.	**2.92**	10.76	16.46	28.13
Pooled Multi.	3.08	**9.64**	**15.48**	**27.79**

Table 2. Performance in unanticipated white noise

4. CONCLUSIONS

A multicondition training strategy for Gaussian PLDA modeling of i-vector representations of speech utterances was presented. Three different multicondition strategies were evaluated on a subset of the NIST SRE 2010 database. The Tied- and Pooled-PLDA approaches were shown to be superior to the independent training of the condition-dependent subsystems. The key to this superior performance was attributed to the ability to leverage data from multiple conditions to improve the estimation of the PLDA hyperparameters. For the noisy conditions included in the development set, the performance improvement of Tied- and Pooled-PLDA was between 2.5 and 3 times better than that of a baseline system trained only on clean data and about 1.5 times better than the independent multicondition training. The posterior probabilities used to combine the individual scores into a final verification score were shown to be quite sparse and localized around the testing condition at hand. Also, the system was tested against unseen noise conditions and the performance of the multicondition strategies was better than that of the baseline system. Finally, since the performance of the Tied- and Pooled-PLDA systems was very similar, the lower number of parameters of the Pooled-PLDA system renders it more appealing.

5. ACKNOWLEDGMENTS

This work has been supported by NSF grant 0917104. The authors would like to thank Alan McCree and Jonas Borgstrom for their insightful feedback.

6. REFERENCES

[1] J. Ming, T. J. Hazen, J. R. Glass, and D. A. Reynolds, "Robust Speaker Recognition in Noisy Conditions," *IEEE Trans.on ASLP*, vol. 15, no. 5, pp. 1711-1723, July 2007.

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis For Speaker Verification," *IEEE Transactions on ASLP,* vol. 19, no. 4, pp. 788 - 798, May 2010.

[3] A. F. Martin and C.S. Greenberg, "NIST 2008 speaker recognition evaluation: Performance across telephone and room microphone channels," in *Proceedings of Interspeech*, Brighton, UK, Sep. 2009, pp. 2579-2582.

[4] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of I-vector Length Normalization in Speaker Recognition Systems," in *Proceedings of Interspeech*, Florence, Italy, August 2011.

[5] S. J. D. Prince, "Probabilistic Linear Discriminant Analysis for Inferences About Identity," in *Proceedings of ICCV*, Rio de Janeiro, 2007.

[6] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in *Proc. Odyssey-2010*, Brno, Czech Republic, 2010.