Modeling Intra-Speaker Variability for Speaker Recognition

Hagai Aronowitz[1], Dror Irony[2] and David Burshtein[3]

[1]Department of Computer Science, Bar-Ilan University, Israel
[2]School of Computer Science, Tel-Aviv University, Israel
[3]School of Electrical Engineering, Tel-Aviv University, Israel
aronowc@cs.biu.ac.il, irony@tau.ac.il, burstyn@eng.tau.ac.il

Abstract

In this paper we present a speaker recognition algorithm that models explicitly intra-speaker inter-session variability. Such variability may be caused by changing speaker characteristics (mood, fatigue, etc.), channel variability or noise variability. We define a session-space in which each session (either train or test session) is a vector. We then calculate a rotation of the session-space for which the estimated intra-speaker subspace is isolated and can be modeled explicitly. We evaluated our technique on the NIST-2004 speaker recognition evaluation corpus, and compared it to a GMM baseline system. Results indicate significant reduction in error rate.

1. Introduction

The Gaussian mixtures model (GMM) algorithm [1-3] has been the state-of-the-art automatic speaker recognition algorithm for many years. The GMM algorithm first fits a parametric model to the target training data and then calculates the log-likelihood of a test utterance given a target speaker assuming frame independence. In [6] it was claimed that the GMM based algorithm has a severe drawback because it assumes there is no intra-session dependency. However, considerable intra-session dependency does exist. This dependency may be attributed to channel, noise, and changing speaker characteristics (mood, fatigue, etc.). It is reasonable to assume that these factors are constant during a single session but change between sessions. The focus of this work is to model explicitly this intra-speaker variability.

In [4, 5] a new speaker recognition technique named TUP was presented. The idea is to train GMMs not only for target speakers but also for the test sessions, hence the name TUP (test utterance parameterization). The likelihood of a test sessions is calculated using only the GMM of the target speaker and the GMM of the test session.

In [6] a novel model for generation of test sessions was presented in which each speaker is modeled by a prior distribution over all possible GMMs instead of being modeled by a single GMM. This model is based on an assumption that at the beginning of a spoken session, a GMM is selected from the speaker's prior distribution, and the frames for the session are generated independently using the selected GMM. This new generative model is not naturally incorporated under the classical GMM framework but is naturally incorporated under the TUP framework. In [6], a simple prior distribution over the GMM space was proposed, and both training and testing algorithms were presented. In [13], intra-speaker variability was modeled using factor analysis.

In this paper we extend the work in [6] by assuming more realistic assumptions on the process of GMM generation that lead to two alternative prior distributions over the GMM space. We present algorithms to train these distributions and to compute the likelihood of a test utterance given a target speaker. More specifically, we factor the GMM space into two subspaces. One subspace is of low dimension and includes the estimated intra-speaker inter-session variability. The second subspace is of high dimension and is modeled by a simple distribution.

The organization of this paper is as follows: we overview the TUP framework in section 2. We present the generative model and the corresponding training and testing algorithms in section 3. Section 4 describes the experimental setup and the results. Section 5 analyzes the complexity of the test algorithm. Finally, section 6 presents conclusions and proposed future work.

2. Test utterance parameterization (TUP)

The basic idea of the TUP framework [4, 5] is to view a GMM not as a classifier but only as a representation for speech sessions. Therefore, estimating a GMM is actually a feature extraction process and should be done for both train and test sessions. The TUP framework is summarized by the following procedure:

1. Estimate GMM Q for target speaker.
2. Estimate GMM P for test session.
3. Compute score $S=S(P, Q)$.
4. Normalize score (T-norm, Z-norm, H-norm etc.) using P, Q, and possibly other GMMs (universal background model – UBM, cohort speakers, etc.).

In [4, 5] it was shown that there exists a function S in the form of $S(P,Q)$ that approximates the log-likelihood of a test utterance given a GMM fitted to a target speaker. The motivation for using the TUP framework in [4, 5] was the task of speaker retrieval in large audio archives. For this task the TUP framework achieved a considerable speedup. In [6] the motivation for using the TUP framework was to be able to exploit a new model for generation of speech by speakers, and it was found that the TUP framework is flexible and is suitable for implementing complex generative models.

3. Session-GMM generative model

In [6] a model for the generation of test sessions was presented. We present an outline of the model in subsection 3.1 and present our new results in the following subsections.

3.1 The Generative model

The classic GMM algorithm assumes that every speaker can

be modeled by a single GMM. The generative model implied by the GMM algorithm is that every frame is emitted by that single GMM independently from other frames. Consequently, if 2 utterances are spoken by the same speaker and are long enough, they should have identical empirical distributions (when length approaches infinity). Unfortunately, this is not the case. In reality there exist session-dependent factors that cause the distribution of different sessions of the same speaker to deviate from each other. A generative model that models explicitly such variability is the following:

Generate session:

1. Generate GMM P for current session according to a speaker dependent prior distribution over the GMM space.
2. Generate a sequence of frames by independent generation of feature vectors according to GMM P.

A session-GMM is the GMM distribution used to generate the frames of a single session. Each speaker is modeled as a prior distribution over session-GMMs. We define G as a GMM in the GMM-space, and derive the likelihood of an observed session X given speaker S as:

$$\Pr(X|S) = \int_{GMM-space} \Pr(X|G)\Pr(G|S)dG \quad (1)$$

In order to develop simple and tractable training and scoring algorithms equation (1) is approximated by assuming that the distribution $\Pr(X|G)$ is much sharper than distribution $\Pr(G|S)$. Therefore, defining P as (equation (2)):

$$P = \arg\max_G \{\Pr(X|G)\} \quad (2)$$

the likelihood of a test session X given speaker S can be approximated by:

$$\Pr(X|S) \cong \Pr(P|S) \quad (3)$$

3.2. The prior distribution Pr(session-GMM | speaker)

We verified empirically [6] that the covariance matrices and the weights of the GMMs can be shared among speakers and sessions. Therefore, the speaker dependent prior distribution over the GMM space $\Pr(P|S)$ needs only to model the means of GMM P. We embed GMM P into a high dimensional Euclidean space by concatenating the means of GMM P into a single high dimensional vector μ. We assume that the distribution of μ is multivariate Gaussian. For every speaker the mean of the distribution of μ can be easily estimated from the training data of the speaker.

The covariance of the distribution is an nxn matrix Σ. A typical size of n is 50,000. In order to estimate Σ robustly we assume all speakers share a global Σ. In order to train Σ we take pairs of same speaker sessions from a development corpus. For each pair we train two GMMs and calculate the difference of the corresponding means of the GMMs: $\bar{\delta} = \bar{\mu}^1 - \bar{\mu}^2$ .The mean of the random vector $\bar{\delta}$ equals $\bar{0}$ and the covariance of $\bar{\delta}$ equals 2Σ. Therefore we can estimate Σ from a collection of difference vectors $\{\bar{\delta}\}$ calculated over pairs of same-speaker sessions pooled from different speakers. Obviously, a full covariance matrix cannot be estimated robustly from the training sessions. One feasible possibility of assuming a diagonal covariance matrix was explored in [6]. However, there is empirical evidence that the elements of μ are highly correlated. We suggest two alternative algorithms to estimate the covariance matrix Σ robustly.

The first algorithm is based on an assumption that $\Sigma = Q^{-1}\Sigma' Q$ where Q is a rotation matrix and Σ' is an nxn matrix which is diagonal excluding its full upper left mxm block. The upper left mxm block of Σ' is supposed to represent the intra-speaker inter-session variability. m is chosen to be the dimension of the intra-speaker inter-session subspace spanned by the training data. Therefore we choose Q to transform the basis of the original GMM space into a new basis in which the first m vectors span the estimated subspace of intra-speaker inter-session variability. The algorithm for computing Q is detailed in the following subsection. Σ' can be robustly estimated because it is mostly diagonal and the upper left mxm full block is guaranteed to be non-singular due to the definitions of Q and m. The elements of the diagonal beneath the upper-left block are set to a small value ε. We found out that the algorithm is not sensitive to the actual value of ε. The algorithm is outlined in figure (1).

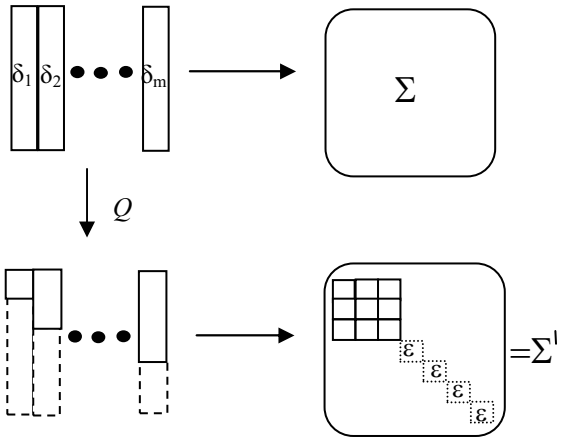

Figure 1: Computing a rotation matrix Q for which the covariance matrix is singular in the upper mxm block and the diagonal elements beneath are set to ε.

The second algorithm is a modification of the previous one but exploits pre-knowledge about the structure of the GMM space. The random variable μ being modeled is actually a concatenation of low dimensional (26 dimensional) vector means representing Cepstral and delta-Cepstral coefficients. We verified empirically the hypothesis that most of the significant correlations between elements of μ are between elements of the same index in the low dimensional (26) space. Therefore we factor the GMM space into 26 disjoint subspaces and apply the algorithm described above separately to each subspace.

3.3. Computing rotation matrix *Q*

We use QR factorization based on Givens rotations, which is known to be stable [10]. Moreover, it may be implemented

efficiently both in terms of time and memory [11]. Using this technique rotation matrix Q can be found in $O(nk^2)$ time, stored in $O(nm)$ memory, and can be applied on a vector in $O(nm)$ time, where n is size of the GMM, k is the number of training vectors $\bar{\delta}$, and m is the dimension of the space spanned by the training vectors $\bar{\delta}$, possibly reduced by a dimension reduction technique such as PCA.

4. Experimental results

4.1 The SPIDRE and the NIST-2004 datasets

Experiments were done on the NIST-2004 speaker evaluation data set using the core condition [9]. Detailed description of the experimental setup can be found in [6] .The SPIDRE corpus [8] was used for training the UBM, and for estimating the speaker-independent covariance matrices of the GMM prior distribution models and as a development set.

4.2 The baseline GMM system

The baseline GMM system in this paper was inspired by the GMM-UBM system described in [1-3]. A detailed description of the baseline system can be found in [4-5]. The baseline system is based on an ETSI-MFCC [7] + derivatives and an energy based voice activity detector. In the verification stage, the log likelihood of each conversation side given a target speaker is divided by the length of the conversation and normalized by the UBM score.

4.3 Normalization techniques

The resulting scores are normalized (independently) by the following techniques: non-parametric Z-norm, T-norm [12], and TZ-norm. Non-parametric Z-norm is similar to Z-norm [2] but uses a histogram to estimate scores distribution instead of fitting a normal distribution. TZ-norm is a combined version of both T-norm and non-parametric Z-norm: a score is first normalized using T-norm and then by non-parametric Z-norm.

4.4 Results

In tables (1, 2) we present results for our two algorithms compared to the baseline GMM. The systems reported in table (1) use non-parametric Z-norm, while the systems reported in table (2) use TZ-norm. For each system we report the equal error rate (EER) and the standard min–DCF as defined in [9]. The corresponding DET curves are presented in figures (2, 3).

	EER (%)	min-DCF
Baseline GMM	15.1	0.053
Session-GMM: single rotation	13.5	0.047
Session-GMM: block diagonal	12.6	0.044
Error reduction	16.6%	17.0%

Table 1: Results of the session-GMM generative model compared to the baseline GMM – using non-parametric Z-norm.

	EER (%)	min-DCF
Baseline GMM	12.4	0.048
Session-GMM: single rotation	12.2	0.047
Session-GMM: block diagonal	10.8	0.042
Error reduction	12.9%	12.5%

Table 2: Results of the session-GMM generative model compared to the baseline GMM - using TZ-norm.

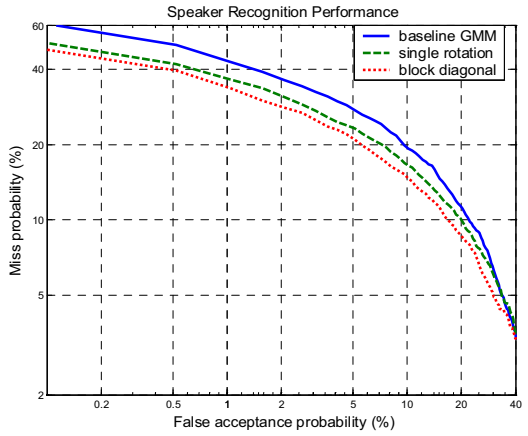

Figure 2: Comparison of the performance of the session-GMM based algorithms compared to the baseline GMM using non-parametric Z-norm.

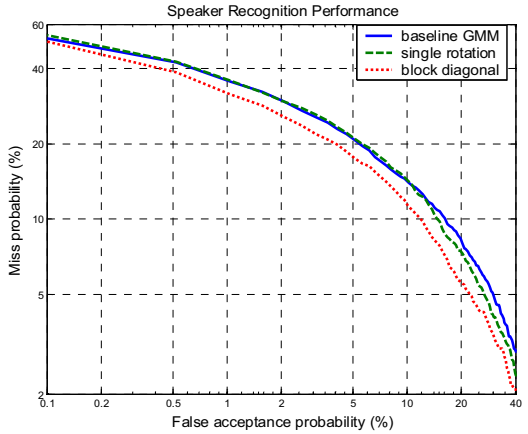

Figure 3: Comparison of the performance of the session-GMM based algorithms compared to the baseline GMM on the NIST-2004 evaluation, using TZ-norm.

From analyzing the results we notice that when using TZ-norm, we get a smaller improvement from the session-GMM algorithms compared to when using non-parametric Z-norm. In figure (4) we show the sensitivity of the performance of the classic GMM algorithm to the various normalization techniques. We conclude from these results that TZ-norm is better than both non-parametric Z-norm and T-norm. In figure (5) we show the sensitivity of the second session-GMM algorithm (using a block-diagonal covariance matrix). Surprisingly, we see that T-norm is not a good normalization technique for the session-GMM algorithm. This observation explains why we get only 12.9% reduction in EER when

using TZ-norm compared to 16.6% reduction when using non-parametric Z-norm.

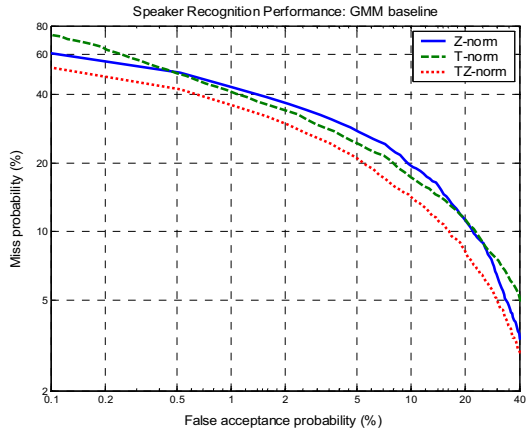

Figure 4: Comparison of the performance of the baseline GMM using Z-norm, T-norm and TZ-norm.

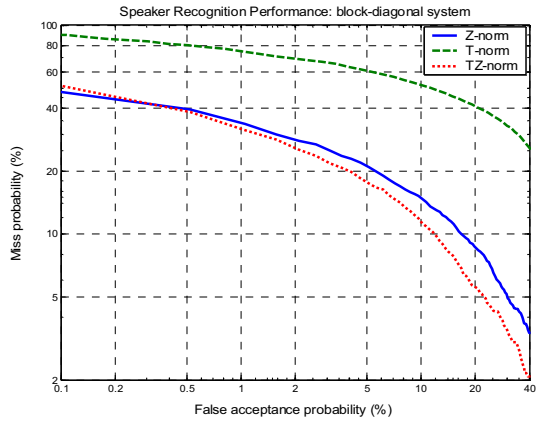

Figure 5: Comparison of the performance of the session-GMM algorithm (block-diagonal covariance matrix) using Z-norm, T-norm and TZ-norm.

5. Time complexity

The time complexity of computing the rotation matrix is analyzed in subsection 3.3.

In order to train a speaker first we train a GMM for the speaker and then rotate it. The time complexity of the rotation algorithm is O(gdm) (g – number of Gaussians in GMM, d – dimension of feature space, m – dimension of intra-speaker inter-session variability space).

In order to test a test-session first a GMM is trained for the speaker and then rotated using Givens rotations. The time complexity of the rotation algorithm is again O(gdm). For every target speaker a score is computed in gd calculations compared to $5dT$ calculations for the GMM algorithm (T – length (in frames) of test data). For $T>g/5$ and many target speakers our algorithm is faster than the GMM algorithm. For a typical g=2048, our technique would be faster than the GMM algorithm for sessions longer that 4 seconds.

6. Conclusions

We have proposed an algorithm for estimating robustly intra-speaker inter-session variability. The results indicate that with appropriate score normalizations, the proposed algorithm outperforms the classic GMM approach. On the NIST-2004 speaker evaluation recognition EER was reduced by 12.9% and the min-DCF was reduced by 12.5%. We hypothesize that a suitable normalization technique may further improve performance.

Our future plan is to estimate the intra-speaker inter-session variability from a larger corpus with channel variability.

7. Acknowledgements

This research was supported by Muscle, a European network of excellence funded by the EC 6th framework IST programme.

8. References

[1] Reynolds D. A., Quatieri T. F. and Dunn R. B., "Speaker verification using adapted Gaussian mixture models", *Digital Signal Processing*, Vol. 10, No.1-3, pp. 19-41, 2000.

[2] Reynolds, D. A., "Comparison of background normalization methods for text-independent speaker verification", in Proc. *Eurospeech*, pp.963-966, 1997.

[3] McLaughlin J., Reynolds D. A., and Gleason T., "A study of computation speed-ups of the GMM-UBM speaker recognition system", in Proc. *Eurospeech*, pp.1215-1218, 1999.

[4] Aronowitz H., Burshtein D. and Amir A., "Speaker indexing in audio archives using Gaussian mixture scoring simulation", in *MLMI:* Proceedings of the Workshop on Machine Learning for Multimodal Interaction. Springer-Verlag LNCS, 2004.

[5] Aronowitz H., Burshtein D. and Amir A., "Speaker indexing in audio archives using test utterance Gaussian mixture modeling", in Proc. *ICSLP*, pp. 609-612, 2004.

[6] Aronowitz H., Burshtein D. and Amir A., "Speaker indexing in audio archives using test utterance Gaussian mixture modeling", in Proc. *ICASSP* 2005.

[7] "Speech processing, transmission and quality aspects (stq); distributed speech recognition; front-end feature extraction algorithm; compression algorithms," ETSI Standard: ETSI-ES-201-108-v1.1.2, 2000, http://www.etsi.org/stq.

[8] Linguistic Data Consortium, SPIDRE documentation file, http://www.ldc.upenn.edu/Catalog/readme_files/spidre.readme.html.

[9] "The NIST Year 2004 Speaker Recognition Evaluation Plan", http://www.nist.gov/speech/tests/spk/2004/.

[10] Higham N.J., "Accuracy and Stability of Numerical Algorithms", Second edition, *SIAM* Press, 2002.

[11] Stewart G.W., "The economical storage of plane rotations", *Numerical Mathematics*, 25: 137-138, 1976.

[12] Auckenthaler R., Carey M., and Lloyd-Thomas H., "Score normalization for text-independent speaker verification systems", *Digital Signal Processing*, vol. 10, pp. 42-54, 2000.

[13] Kenny P., Boulianne G., Ouellet P., Dumouchel P., "Factor Analysis Simplified", in Proc. *ICASSP* 2005.