Score Stabilization for Speaker Recognition Trained on a Small Development Set

Hagai Aronowitz
IBM Research - Haifa, Israel
hagaia@il.ibm.com

Abstract

Nowadays state-of-the-art speaker recognition systems obtain quite accurate results for both text-independent and text-dependent tasks as long as they are trained on a fair amount of development data from the target domain (assuming clean speech). In this work, we address the challenge of building a speaker recognition system with a small development dataset from the target domain without using out-of-domain data whatsoever. When development data is limited, the Nuisance Attribute Projector (NAP) algorithm is (in general) superior to the i-vector approach. We have investigated the relative degradation observed from the different components of the NAP system trained on a small dataset and conclude that score normalization is a major source of degradation. We introduce a novel method for stabilizing the normalized scores. We explicitly estimate a low dimensional subspace in supervector space which accounts for high variability in score normalization parameters. We then compensate the estimated subspace. We report experiments on both text-dependent and text-independent tasks which validate our method and show large error reductions.

Index Terms: speaker verification, text-dependent, domain adaptation, score normalization

1. Introduction

The introduction of i-vectors [1] and Probabilistic Linear Discriminant Analysis (PLDA) [2] resulted in very low error rates in the recent NIST text-independent (TI) speaker recognition evaluations (SREs) [2]. However, the success of i-vector based PLDA is dependent on the availability of a large development set with thousands of multi session speakers, to estimate the PLDA hyper-parameters. Moreover, the development data must be matched to the target data.

When the target data is highly mismatched to the available development data, for instance due to channel mismatch or in text-dependent speaker recognition, a common strategy is to collect some data from the target domain. The collected in-domain data is then used to either train the speaker recognition system from scratch [3-6] or to adapt an already existing system [7-9, 4-6].

In this paper we investigate how to successfully train a speaker recognition system with limited in-domain data (no use of out-of-domain data whatsoever). We consider both text dependent and text independent tasks. Furthermore, we choose to use Nuisance Attribute Projection (NAP) with GMM-supervectors [10] as it outperforms i-vector based approaches when development data is limited [5, 6]. Nevertheless, our proposed methods would most probably be beneficial for the i-vector framework as well.

We address the problem of accurate estimation of score normalizing parameters from a small dataset. We propose a novel method that explicitly estimates a low dimensional subspace in the supervector space which accounts for high variability in the score normalization parameters. We then compensate the estimated subspace.

The remainder of this paper is organized as follows: Section 2 describes the baseline system. Section 3 describes the proposed methods. Section 4 describes the experiments and results. Finally, Section 5 concludes.

2. GMM-NAP-based System

In the GMM-NAP framework a GMM is adapted for each session (enrollment, testing and development) from a UBM using MAP-adaptation. A projection is estimated from the development set and is used to compensate intra-speaker intersession variability (such as channel variability).

2.1. Front-end

The front-end is based on Mel-frequency cepstral coefficients (MFCC). An energy based voice activity detector is used to locate and remove non-speech frames. The final feature set consists of 12 cepstral coefficients augmented by 12 delta and 12 double delta coefficients extracted every 10ms using a 25ms window. Feature warping is applied with a 300 frame window before computing the delta and double delta features.

2.2. GMM supervector extraction

A 512-Gaussian Universal Background Model (UBM) with diagonal covariance matrices is trained on the development set and is used for extracting the supervectors. The means of the GMMs are stacked into a supervector after normalization with the corresponding standard deviations of the UBM and multiplication by the square root of the corresponding weight from the UBM:

$$ s = \sum_{\ell=1}^{12} \left( \frac{\lambda_{\ell}^{1/2}}{\lambda_{\ell,UBM}} \otimes I_F \right) \mu $$  (1)

where $\mu$ stands for the concatenated GMM means, $\lambda_{\ell,UBM}$ stands for the vectorized UBM weights, $\sum$ is a block diagonal matrix with covariance matrices from the UBM on its diagonal, $F$ is the feature vector dimension, $\otimes$ is the Kronecker product, and $I_F$ is the identity matrix of rank $F$.

2.3. NAP estimation

A low rank projection $P$ is estimated as follows. First, we remove from each supervector in the development its corresponding speaker supervector mean. The resulting supervectors are named nuisance supervectors. We compute the covariance matrix of the nuisance supervectors and apply PCA to find a basis to the nuisance space. Projection $P$ is created by stacking the top $k$ eigenvectors as columns in matrix $V$:

$$ P = I - V V^T. $$  (2)

In [12] a novel technique named Gaussian-based smoothing (GBS) was introduced to better estimate the NAP projection with limited data. We report results using this technique as
well as using the method described above. Basically, GBS 
exploits the structural nature of GMM supervectors. 
Knowledge on the way GMM supervectors are created 
(namely a concatenation of statistics obtained for a set of 
Gaussians over the feature space) is used to guide and 
constrain modeling in high dimensional supervector space.

2.4. NAP compensation

The enrollment supervectors are compensated by applying projection $P$. There is no need to project the test supervectors because we use dot-product scoring (see next subsection):

$$ (P_x y) = x' P' P y = (P x)' y . $$  \hfill (3)

2.5. Scoring and score normalization

Scoring is performed using a dot-product between the 
compensated enrollment and test supervectors. We apply ZT-
score normalization [10] using the sessions from development 
data. Therefore, given a raw scoring function $\varphi(s, x)$ between 
an enrolled supervector $s$ and a test supervector $x$, ZT-norm 
aims at standardization of the distribution of $\varphi(s, x)$ on 
impostor trials. The Z-norm method estimates the mean and 
variance of $\varphi(s, x)$ and uses them to standardize $\varphi(s, x)$.

$$ \varphi_{\text{Znorm}}(s, x) = \frac{\varphi(s, x) - \mu_Z(s, x)}{\sigma_Z(s, x)} $$

$$ \mu_Z(s, x) = E_s \varphi(s, x') $$

$$ \sigma_Z(s, x) = \sqrt{Var_s \varphi(s, x')} $$

(4)

Equivalent descriptions for T-norm and ZT-norm can be found 
in [13].

3. Score Stabilization using Subspace Removal

The GMM-NAP based system uses the development data for 
three purposes: UBM training, NAP projection estimation and 
scoring normalization. In [4, 5] it was observed that UBM 
training is less sensitive to the size of the development set 
compared to the degradation observed for the two other 
components. We therefore focus in this paper on score 
normalization, and address the NAP projection estimation 
issue in [12].

3.1. Analysis

We limit our analysis to Z-norm but claim it can be extended 
trivially to T-norm and (less trivially) to ZT-norm. Given 
development data of $n$ sessions with the corresponding 
supervectors $X_0 = \{x_1, \ldots, x_n\}$. The unbiased estimates for the Z-
norm parameters are

$$ \hat{\mu}_Z(s, X) = \frac{1}{n} \left( \varphi(s, x) \right)_{i \in X} $$

$$ \hat{\sigma}_Z^2(s, X) = \frac{1}{n-1} \left( \varphi^2(s, x) \right)_{i \in X} - \left( \varphi(s, x) \right)^2_{i \in X} $$

(5)

We aim at stabilizing the normalized scores by minimizing the 
expected variances of $\hat{\mu}_Z(s, X) / \hat{\sigma}_Z(s, X)$ and $\hat{\sigma}_Z(s, X)$ 
over the distributions of $X$ and $s$.

Without loss of generality we assume in the next 
subsections that the mean of the supervector is 0 and that the 
covariance matrix of the supervector population is diagonal 
with its eigenvalues $\{\lambda_i\}$ on its diagonal.

3.1.1. Stabilizing $\hat{\sigma}_Z(s, X)$

Assuming that the impostor scores for a speaker $s$ are 
independently drawn from a normal distribution, the variance 
of $\hat{\sigma}_Z(s, X)$ with respect to development data $X$ is

$$ \text{Var}_X \left( \hat{\sigma}_Z^2(s, X) \right) = \frac{2 \sigma_{z(s)}^4}{n-1} = \frac{1}{n-1} \left( s' Cov(X)s \right)^2 $$ \hfill (6)

and the expected variance (with respect to $s$) is

$$ E_s \left[ \text{Var}_X \left( \hat{\sigma}_Z^2(s, X) \right) \right] = \frac{1}{n-1} tr \left( Cov(x) \right)^2 $$ \hfill (7)

Therefore, in order to minimize the expected variance of $\hat{\sigma}_Z(s, X)$ a low dimensional subspace spanning the top 
eigenvectors of $Cov(s)$ (which is the total variability 
covariance matrix) should be removed from the supervector 
space.

3.1.2. Stabilizing $\hat{\mu}_Z(s, X) / \hat{\sigma}_Z(s, X)$

Assuming that $\hat{\sigma}_Z(s, X)$ has already been stabilized (in the 
previous subsection) we approximate $\hat{\mu}_Z(s, X) / \hat{\sigma}_Z(s, X)$ with

$$ \hat{\mu}_Z(s, X) / \hat{\sigma}_Z(s, X) = \frac{1}{n} s' Cov(X)s / \hat{\sigma}_Z^2(s, X) $$ \hfill (8)

Therefore, we cannot stabilize $\hat{\mu}_Z(s, X) / \hat{\sigma}_Z(s, X)$ further 
using a subspace removal technique.

3.2. Proposed method

We propose to seek for a low dimensional subspace in the 
high-level vector space (supervector, i-vector / etc.) which 
upon removal decreases substantially the expected variance of 
the score normalization parameters.

For the case of dot-product scoring, the optimal subspace to 
be removed is spanned by the top eigenvectors of the total 
variability covariance matrix.

3.3. Related work

Removal of the span of the top eigenvectors of the total 
variability covariance matrix was used in [14] for speaker 
recognition using kernel-PCA. In [15] it was used for speaker 
recognition in two-wire (summed) conversations. In [16, 4] it 
was used instead of standard intra-speaker variability removal 
due to lack of multi-session (per speaker) development data. 
Lately, it was used in [7] in the i-vector PLDA framework for 
robustness to training on mismatched development data (the 
domain adaptation challenge [7]). It is important to note that 
removal of this subspace does not improve the state-of-the-art 
with enough development data.

4. Experiments and Results

We describe the full datasets and reduced subsets in 
subsections 4.1 and 4.2 respectively. Then we describe the TD 
and TI experiments in subsections 4.3 and 4.4 respectively.
4.1. Datasets

4.1.1. TD task: the WF dataset

The WF dataset consists of 750 speakers which are partitioned into a development set (200 speakers) and an evaluation dataset (550 speakers). Each speaker has 2 sessions using a landline phone and 2 sessions using a cellular phone. The data collection was accomplished over a period of 4 weeks.

Four authentication conditions were defined and collected (global, speaker-dependent and prompted passphrases as well as free text), and experimental results were reported for them in [3]. In this work we limit ourselves to the global (passphrase is shared among all speakers) condition for which the same passphrase is used for both development, enrollment and verification. We report results for the 10-digit pass phrase 0-1-2-3-4-5-6-7-8-9 which we name ZN.

In the WF dataset each session contains 3 repetitions of ZN. For each enrollment session we use all 3 repetitions for enrollment, and for each verification session we use only a single repetition.

A comprehensive description of the WF dataset can be found in [3].

4.1.2. TI task: NIST-2010

We use the NIST 2010 SRE [11] for evaluation. We use the NIST 2010 SRE male core trial list with telephone conditions (5, 6 and 8) for evaluation. The dataset consists of 355, 178 and 119 target trials and 13746, 12825 and 10997 impostor trials respectively.

The development dataset consists of male sessions from NIST 2004 and 2006 SREs (telephone data only). In total we use 4374 sessions from 521 speakers.

4.2. Reduced developments datasets

4.2.1. TD task

We define the following subsets of the WF dataset (Table 1). In Table 1 L stands for a landline sessions and C for a cellular session. For instance, LLCC stands for 4 sessions (2 landline + 2 cellular), and LC stands for 2 sessions (1 landline + 1 cellular). Except for the last row (30RR), subsets are gender balanced. The last row describes a subset for which the genders are highly imbalances and the two sessions per speaker are selected randomly. The purpose of the 30RR subset is to simulate a realistic condition when the actual data collected in not balanced as planned.

Table 1. Reduced development sets for the TD task.

<table>
<thead>
<tr>
<th>Name</th>
<th>Number of speakers</th>
<th>Sessions per speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>200</td>
<td>LLCC</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>LLCC</td>
</tr>
<tr>
<td>50LC</td>
<td>50</td>
<td>LC</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>LLCC</td>
</tr>
<tr>
<td>30LC</td>
<td>30</td>
<td>LC</td>
</tr>
<tr>
<td>30LL</td>
<td>30</td>
<td>LL</td>
</tr>
<tr>
<td>30CC</td>
<td>30</td>
<td>CC</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>LLCC</td>
</tr>
<tr>
<td>20LC</td>
<td>20</td>
<td>LC</td>
</tr>
<tr>
<td>20RR</td>
<td>30</td>
<td>RR</td>
</tr>
</tbody>
</table>

4.2.2. TI task

We define the following subsets of the TI development set. The number of speakers is varied between 20 and 500. We create 3 different subsets for every chosen number of speakers. The first subset consists of 2 sessions per speaker. The second subset consists of 4 sessions per speaker.

4.3. TD Experiments and results

Table 2 reports results for the TD task using different subsets for development. The baseline system (with NAP subspace dimension of 10 which was found optimal in [12]) is contrasted to the proposed method. We use subspace dimensions 10, 25 and 50 for score stabilization (SS). We also report results for score normalization with the full development set (but using a subset for NAP training) in order to assess how far can we get with our technique. In order to reduce the variance of our measured EERs, we repeat each experiment 10 times with randomly selected subsets. Figure 1 shows a bar plot of the results for selected systems.

Note that for all subsets the proposed system outperforms the baseline system (except for full development data). The last two rows in Table 2 report the relative error reduction and EER recovery rate (SS 25).

Table 2. Results for the TD task using different subsets for development. The baseline NAP system method is contrasted to the proposed method and to score normalization with the full development dataset. Results are averaged over 10 randomly selected subsets. Best result for each subset is in bold.

<table>
<thead>
<tr>
<th>System</th>
<th>20LC</th>
<th>20</th>
<th>30CC</th>
<th>30LL</th>
<th>30RR</th>
<th>30LC</th>
<th>30</th>
<th>50LC</th>
<th>50</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAP 10</td>
<td>2.8</td>
<td>2.5</td>
<td>3.2</td>
<td>3.3</td>
<td>3.5</td>
<td>2.4</td>
<td>2.1</td>
<td>1.8</td>
<td>1.6</td>
<td>1.0</td>
</tr>
<tr>
<td>NAP 10 SS 10</td>
<td>2.4</td>
<td>2.0</td>
<td>2.7</td>
<td>2.4</td>
<td>2.4</td>
<td>2.4</td>
<td>2.0</td>
<td>1.8</td>
<td>1.6</td>
<td>1.4</td>
</tr>
<tr>
<td>NAP 10 SS 25</td>
<td>2.3</td>
<td>2.0</td>
<td>2.4</td>
<td>2.4</td>
<td>2.4</td>
<td>2.4</td>
<td>2.1</td>
<td>1.8</td>
<td>1.7</td>
<td>1.5</td>
</tr>
<tr>
<td>NAP 10 SS 50</td>
<td>2.3</td>
<td>1.9</td>
<td>2.4</td>
<td>2.6</td>
<td>2.5</td>
<td>2.1</td>
<td>1.8</td>
<td>1.8</td>
<td>1.5</td>
<td>1.1</td>
</tr>
<tr>
<td>NAP 10 Norm-full</td>
<td>1.7</td>
<td>1.6</td>
<td>2.0</td>
<td>2.0</td>
<td>1.9</td>
<td>1.5</td>
<td>1.4</td>
<td>1.5</td>
<td>1.2</td>
<td>1.0</td>
</tr>
<tr>
<td>EER reduction (SS 25)</td>
<td>18%</td>
<td>20%</td>
<td>25%</td>
<td>27%</td>
<td>31%</td>
<td>13%</td>
<td>14%</td>
<td>6%</td>
<td>6%</td>
<td>-10%</td>
</tr>
<tr>
<td>Recovery rate (SS 25)</td>
<td>45%</td>
<td>56%</td>
<td>67%</td>
<td>69%</td>
<td>69%</td>
<td>33%</td>
<td>43%</td>
<td>33%</td>
<td>25%</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3. Results for the TD task using different subsets for development. The GBS-NAP system [12] is contrasted to the proposed method. Results are averaged over 10 randomly selected subsets. Best result for each subset is in bold.

<table>
<thead>
<tr>
<th>System</th>
<th>20LC</th>
<th>20</th>
<th>30CC</th>
<th>30LL</th>
<th>30RR</th>
<th>30LC</th>
<th>30</th>
<th>50LC</th>
<th>50</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBS</td>
<td>2.5</td>
<td>2.3</td>
<td>2.7</td>
<td>2.7</td>
<td>2.7</td>
<td>2.2</td>
<td>2.1</td>
<td>1.8</td>
<td>1.8</td>
<td>1.6</td>
</tr>
<tr>
<td>GBS SS 10</td>
<td>2.1</td>
<td>2.0</td>
<td>2.2</td>
<td>2.1</td>
<td>2.1</td>
<td>1.9</td>
<td>1.8</td>
<td>1.7</td>
<td>1.6</td>
<td>1.3</td>
</tr>
<tr>
<td>GBS SS 25</td>
<td>2.1</td>
<td>1.9</td>
<td>2.2</td>
<td>2.1</td>
<td>2.0</td>
<td>1.9</td>
<td>1.8</td>
<td>1.7</td>
<td>1.6</td>
<td>1.3</td>
</tr>
<tr>
<td>GBS SS 50</td>
<td>2.1</td>
<td>1.8</td>
<td>2.2</td>
<td>2.3</td>
<td>2.4</td>
<td>1.9</td>
<td>1.7</td>
<td>1.7</td>
<td>1.6</td>
<td>1.4</td>
</tr>
<tr>
<td>EER reduction (SS 25)</td>
<td>16%</td>
<td>17%</td>
<td>19%</td>
<td>22%</td>
<td>26%</td>
<td>14%</td>
<td>14%</td>
<td>6%</td>
<td>11%</td>
<td>19%</td>
</tr>
</tbody>
</table>
the percentage of the error due to estimating the score normalization parameters on limited data that is recovered by the proposed method.

We report similar experiments using the GBS [12] method for NAP projection estimation. The results are reported in Table 3 and Figure 1. For all evaluated subsets (including full dataset) score stabilization improves accuracy.

4.4. TI Experiments and results

Tables 4 and 5 report results for the TI task using different subsets of the development dataset. Table 4 reports results for two sessions per speaker, and Table 5 reports results for four sessions per speaker. Score stabilization (with a subspace dimension of 25) is evaluated on the baseline NAP method (with a subspace dimension of 100) and GBS-NAP (with a subspace dimension of 1000). In order to reduce the variance of our measured EERs, we repeat each experiment 10 times with different randomly selected subsets.

Note that for 108 experiments, score stabilization improves accuracy for 80 experiments and degraded accuracy in only 17 (usually for 20 and 30 speakers).

Table 5. Results for the TI task as a function of number of speakers in subset. Subsets contain four randomly selected subsets. Best result for each subset is in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cond.</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAP 100</td>
<td>5</td>
<td>12.4</td>
<td>12.1</td>
<td>10.5</td>
<td>9.0</td>
<td>7.0</td>
<td>4.8</td>
<td>4.4</td>
<td>3.9</td>
<td>3.9</td>
</tr>
<tr>
<td>NAP+SS 10</td>
<td>5</td>
<td>12.4</td>
<td>11.5</td>
<td>9.5</td>
<td>9.3</td>
<td>6.5</td>
<td>4.8</td>
<td>4.2</td>
<td>3.9</td>
<td>3.9</td>
</tr>
<tr>
<td>GBS+SS 10</td>
<td>6</td>
<td>11.3</td>
<td>10.4</td>
<td>9.3</td>
<td>9.0</td>
<td>7.9</td>
<td>7.0</td>
<td>6.7</td>
<td>6.5</td>
<td>6.5</td>
</tr>
<tr>
<td>NAP+SS 10</td>
<td>6</td>
<td>11.6</td>
<td>12.5</td>
<td>11.7</td>
<td>10.1</td>
<td>6.2</td>
<td>5.6</td>
<td>5.1</td>
<td>5.0</td>
<td>5.0</td>
</tr>
<tr>
<td>NAP-100</td>
<td>6</td>
<td>13.8</td>
<td>11.8</td>
<td>11.8</td>
<td>10.1</td>
<td>8.4</td>
<td>6.6</td>
<td>5.7</td>
<td>5.1</td>
<td>5.0</td>
</tr>
<tr>
<td>GBS</td>
<td>8</td>
<td>11.8</td>
<td>10.1</td>
<td>10.1</td>
<td>10.1</td>
<td>9.0</td>
<td>7.9</td>
<td>7.3</td>
<td>7.8</td>
<td>7.8</td>
</tr>
<tr>
<td>GBS+SS 10</td>
<td>8</td>
<td>5.1</td>
<td>4.3</td>
<td>3.5</td>
<td>3.4</td>
<td>2.3</td>
<td>2.0</td>
<td>1.6</td>
<td>1.0</td>
<td>0.8</td>
</tr>
<tr>
<td>GBS</td>
<td>5</td>
<td>5.2</td>
<td>4.2</td>
<td>4.2</td>
<td>4.1</td>
<td>4.1</td>
<td>4.5</td>
<td>4.2</td>
<td>4.2</td>
<td>4.2</td>
</tr>
<tr>
<td>GBS+SS 10</td>
<td>5</td>
<td>5.8</td>
<td>5.0</td>
<td>4.2</td>
<td>4.1</td>
<td>3.4</td>
<td>2.6</td>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
</tr>
</tbody>
</table>

5. Conclusions

In this work we have explored the possibility of building speaker recognition systems with small development datasets. We investigated the NAP framework as it has been found in the past to outperform the i-vector framework when development data is limited [3-6].

We propose to stabilize score normalization parameters by removing a low dimensional subspace from the supervector space. We have proved that in the dot-product framework the optimal subspace to remove is spanned by the top eigenvectors of the total variability covariance matrix.

For the TD task, ~50% (in average) of the error due to score normalization with limited data is recovered by the proposed method (~20% relative error reduction). For the TI task, the proposed method reduced error by 9% relative (in average).

For future work, we plan to explore the framework in the context of an i-vector PLDA system.

6. Acknowledgements

This work was part of the DEM@CARE EU project, partly funded by the European Commission in the scope of the 7th ICT framework.

7. References


H. Aronowitz, "Inter dataset Variability compensation for speaker recognition", in Proc. ICASSP, 2014.

H. Aronowitz, "Compensating Inter-Dataset Variability in PLDA Hyper-Parameters for Robust Speaker Recognition", in Proc. Speaker Odyssey, 2014.


