Leveraging from the NIST i-vector machine learning challenge

Elie Khoury

Biometrics group, Idiap Research Institute, Switzerland

Lausanne, October 24, 2014
AUTOMATIC SPEAKER RECOGNITION

APPLICATIONS

- Access control and surveillance
- Indexing and retrieval systems, etc.
AUTOMATIC SPEAKER RECOGNITION

CHALLENGES:

- High variability in data acquisition (background noise, microphone, overlap speech, etc.)

- Short duration utterance
- Emotional state, age, etc.
OUTLINE

INTRODUCTION

TOTAL VARIABILITY MODELING (i-VECTORS)

NIST i-VECTOR CHALLENGE

IDIAP PARTICIPATION

CONCLUSIONS
OUTLINE

INTRODUCTION

TOTAL VARIABILITY MODELING (i-VECTORS)

NIST I-VECTOR CHALLENGE

IDIAP PARTICIPATION

CONCLUSIONS
Total Variability Modeling

- Current *state-of-the-art* in speaker recognition
- Acts as *front-end feature extractor* of the so-called *i-vectors*
**Audio Features**

**Extraction**

![Diagram of audio feature extraction process]

**Distribution: Gaussian Mixture Model (GMM)**

\[
P(o \mid \Theta) = \sum_{c=1}^{C} \omega_c \mathcal{N}[o \mid \mu_c, \Sigma_c]
\]  

(1)

with \( \Theta = \{\omega_c, \mu_c, \Sigma_c\}_{c=1,...,C} \): parameters

and \( o \): feature vector of dimensionality \( D_o \)
GMM

- Audio features (MFCC, LFCC, PLP, etc.)
- Speaker-dependent GMM model
- Speaker-independent GMM model (UBM)
- MAP adaptation

Problem: Maximum-a-posteriori (MAP) adapts to not only speaker-specific characters of the speech, but also channel (background noise, microphone, etc.).
GMM

SUPERVECTOR REPRESENTATION

\[ s_i = m + d_i \]

- Speaker supervector
- UBM supervector
- Speaker-specific offset
**GMM**

**SUPERVECTOR REPRESENTATION**

$$s_i = m + d_i$$

- Speaker supervector
- UBM supervector
- Speaker-specific offset

**Problem:**

*Maximum-a-posteriori* (MAP) adapts to not only speaker-specific characters of the speech, but also channel (background noise, microphone, etc.)!
JOINT FACTOR ANALYSIS (JFA)

\[ s_{i,j} = m + V y_i + U x_{i,j} + D z_i \]

- **V**: Speaker subspace (also know as eigenvoice matrix)
- **U**: Channel subspace (also know as eigenchannel matrix)
- **D**: Residual matrix

Estimating **V**, **U** and **D** is done using EM on the training set.
**Joint Factor Analysis (JFA)**

\[ s_{i,j} = m + V y_i + U x_{i,j} + D z_i \]

- **V**: Speaker subspace (also known as eigenvoice matrix)
- **U**: Channel subspace (also known as eigenchannel matrix)
- **D**: Residual matrix

Estimating **V**, **U** and **D** is done using EM on the training set.

**Problem:**
In practice, the estimated channel factor \( x_{i,j} \) (which models only the channel effects) contains also information about the speaker!
**TOTAL VARIABILITY MODELING (TV)**

\[ s_{i,j} = m + T v_{i,j} \]

- **Speaker supervector**
- **UBM supervector**
- **Total-variability matrix**
- **i-vector**

**i-vectors**

- **Low-dimensional** (e.g. \( dim = 600 \))
- **Fixed-length**: independent from the duration of the speech utterance
- **Very discriminative**: a simple cosine distance between i-vectors can achieve good performance
OUTLINE

INTRODUCTION

TOTAL VARIABILITY MODELING (I-VECTORS)

NIST I-VECTOR CHALLENGE

IDIAP PARTICIPATION

CONCLUSIONS
NIST i-vector machine learning challenge

Motivation

- Improving technology performance over an established baseline (provided by NIST)
- Making the field more accessible to ML researchers by providing directly the i-vectors instead of the original speech signal

Task: Speaker detection

- Target speaker (client) model: 5 i-vectors
- Test speech utterance: 1 single i-vector
- Goal: compute the likelihood that the speaker in the test utterance is the target speaker
NIST I-VECTOR MACHINE LEARNING CHALLENGE

EVALUATION METRIC

- Minimum detection cost function (DCF)
  \[
  \text{minDCF} = \min_t (\text{FRR}(t) + 100 \cdot \text{FAR}(t))
  \]

- FRR: False Rejection Rate
- FAR: False Acceptance Rate

The lower the minDCF => the better the system
NIST i-vector machine learning challenge

Development Data

- 36,572 unlabeled i-vectors (their identity is unknown)

Enrol/Test Data

- 1,306 target models (6,530 i-vectors)
- 9,634 test i-vectors
- Total of 12,582,004 trials
  - 40% for the progress set
  - 60% for the evaluation set

Side Information

- Duration of speech after voice activity detection
OUTLINE

INTRODUCTION

TOTAL VARIABILITY MODELING (i-VECTORS)

NIST I-VECTOR CHALLENGE

IDIAP PARTICIPATION

CONCLUSIONS
**PROGRESS AND BASELINE**

- Baseline
  - 0: $\text{minDCF} = 0.386$
  - 1: $\text{minDCF} = 0.372$
  - 2: $\text{minDCF} = 0.356$
  - 3: $\text{minDCF} = 0.302$
  - 4: $\text{minDCF} = 0.302$
  - 5: $\text{minDCF} = 0.292$
  - 6: $\text{minDCF} = 0.286$
  - 7: $\text{minDCF} = 0.258$
  - 8: $\text{minDCF} = 0.247$

elie.khoury@idiap.ch 4/21
PROGRESS AND BASELINE

Progress highlights

- Clustering
- Score normalization

Conclusions

0. Baseline

\[ \text{minDCF} = 0.386 \]
\[ \text{minDCF} = 0.372 \]
\[ \text{minDCF} = 0.356 \]
\[ \text{minDCF} = 0.302 \]
\[ \text{minDCF} = 0.292 \]
\[ \text{minDCF} = 0.286 \]
\[ \text{minDCF} = 0.258 \]
\[ \text{minDCF} = 0.247 \]

elie.khoury@idiap.ch

4/21
**BASELINE**

**WHITENING**

- **Goal:** Normalize the i-vector space such that the covariance matrix of the i-vectors $\bar{\Sigma} = I$

$$\nu_{i,j}^{(\text{whitened})} = W^T (\nu_{i,j} - \bar{\nu})$$  \hspace{0.5cm} (3)

$$\bar{\Sigma}^{-1} = WW^T$$  \hspace{0.5cm} (4)

**LENGTH-NORMALIZATION**

- **Goal:** Reduce the impact of a mismatch between training and test i-vectors

$$\nu_{i,j}^{(1\text{-norm})} = \frac{\nu_{i,j}}{\|\nu_{i,j}\|}$$  \hspace{0.5cm} (5)
**Probabilistic Linear Discriminant Analysis**

- **Goal:** Incorporate both *between-speaker* and *within-speaker* information

\[ v_{i,j} = Fh_i + Gk_{i,j} + \epsilon_{i,j} \]  \hspace{1cm} (6)

- **Pros:**
  - Generation of LLR scores => suitable for speaker detection task
  - **Better performance** than cosine scoring

- **Cons:**
  - It requires *labeled* training data! Not the case for NIST i-vector challenge :( 

- **Solution:**
  - Provide synthetic labels for the training data, so let’s cluster them!
Cosine-PLDA clustering

- **1\textsuperscript{st} step** of clustering uses Cosine measure
  - After each merge, the similarity matrix is updated by re-computing the cosine measure between average i-vectors of the resulting clusters

- **2\textsuperscript{nd} step** of clustering uses PLDA where automatically labeled i-vectors are used to train the PLDA model
  - After each merge, the PLDA model and the similarity matrix could be updated
CLUSTERING RESULTS

Figure: MinDCF values on the progress set in terms of the number of clusters for both clustering methods using the PLDA recognition system
OUTLINE

INTRODUCTION

TOTAL VARIABILITY MODELING (i-vectors)

NIST i-vector challenge

IDIAP participation

CONCLUSIONS
CONCLUSIONS

- Quick overview of state-of-the-art speaker recognition systems
- Description of the NIST i-vector ML challenge
- Successful Idiap participation
  - Robust Cosine-PLDA algorithm for Speaker clustering
  - Top ranking in the challenge ($N_{\text{participants}} = 105$ and $N_{\text{submissions}} = 8192$)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Affiliation</th>
<th>Best Score on progress set</th>
<th>Number of Submissions</th>
<th>Time of Best Submission</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sébastien Marcel</td>
<td>Idiap research institute</td>
<td>0.214</td>
<td>38 submissions</td>
<td>2014-07-31 13:36:26 UTC</td>
</tr>
<tr>
<td>2</td>
<td>Elie Khoury</td>
<td>Idiap Research Institute</td>
<td>0.230</td>
<td>686 submissions</td>
<td>2014-07-31 12:17:35 UTC</td>
</tr>
<tr>
<td>3</td>
<td>Anonymized</td>
<td></td>
<td>0.239</td>
<td>411 submissions</td>
<td>2014-04-03 17:27:00 UTC</td>
</tr>
<tr>
<td>4</td>
<td>Anonymized</td>
<td></td>
<td>0.239</td>
<td>134 submissions</td>
<td>2014-04-04 11:19:57 UTC</td>
</tr>
</tbody>
</table>
Reproducible Research

http://www.idiap.ch/software/bob

- Signal processing and machine learning toolbox
- Open source project
- Integrate implementations of modeling techniques

Satellite package

https://pypi.python.org/pypi/bob.spearr

- Speaker recognition toolbox
- Relies on Bob