Multi-level Deep Neural Network Adaptation for Speaker Verification Using MMD and Consistency Regularization

Weiwei Lin¹, Man-Wai MAK¹, Na Li², Dan Su² and Dong Yu²

¹Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR of China

²Tencent AI Lab, China

ICASSP’20
4-8 May 2020
Contents

1. Domain Mismatch in Speaker Recognition
2. Domain Adaptation
3. MMD-Based Speaker Embedding Adaptation
4. Experiments and Results
5. Conclusions
Domain Mismatch

- When training data and test data of speaker recognition systems have a severe mismatch, the performance degrades rapidly.
- The mismatch can be caused by languages, channels, noises, and genders.
- Collecting more data to retrain the system is time-consuming and computationally-expensive.
- We need to adapt existing systems to new environments or create a domain-invariant feature space.
Domain Adaptation

- Can be performed during system training by
  1. making the speaker embedding network domain-invariant
  2. transforming the speaker embedding to domain-invariant space
  3. adapting the PLDA model

Speech from multiple domains
Speaker-embedding Adaptation

- **Goal:** Train the speaker embedding network to produce domain-invariant feature vectors.
- Minimize domain discrepancy at both frame-level and utterance-level
- Apply consistency regularization to leverage unlabeled target-domain data.
X-Vector Network

- **Sliding Window**
  - 23 x 5 MFCC or Filter bank features

- **Conv**
  - 512 channels
  - 512 x 512 weights
  - \( \{t\} \) Conv

- **Conv**
  - (512 x 3) x 512 weights
  - 512 channels
  - \( \{t - 3, t, t + 3\} \) Conv

- **Conv**
  - (512 x 3) x 512 weights
  - 512 channels
  - \( \{t - 2, t, t + 2\} \) Conv

- **Conv**
  - 512 x 1500 weights
  - 512 channels
  - 1500 channels

- **Stat Pooling**
  - [mean, std]
  - \( \{0, \ldots, T - 1\} \)

- **FC**
  - 3000 x 512 weights
  - 512 x 3000 weights

- **K Softmax nodes**

X-vector
Speaker Embedding Adaptation

$$\frac{1}{I} \sum_{i=1}^{I} J(p\Theta(y|x^s_i, y^s_i))$$

$$\lambda \cdot D(H^7_s, H^7_t) + \alpha \cdot D(H^5_s, H^5_t)$$

$$\beta \cdot D(H^7_t, \hat{H}^7_t)$$
Maximum Mean Discrepancy (MMD)

- MMD is a nonparametric approach to measuring the distance between two distributions.
- The basic idea is to non-linearly map the input to an RKHS and compute the distance between the means of the two distributions in that space.
Maximum Mean Discrepancy (MMD)

\[
D_{\text{MMD}} = \left\| \frac{1}{N} \sum_{i=1}^{N} \phi(x_i) - \frac{1}{M} \sum_{j=1}^{M} \phi(y_j) \right\|^2 \\
= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \phi(x_i)^T \phi(x_{i'}) - \frac{2}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \phi(x_i)^T \phi(y_j) \\
\quad + \frac{1}{M^2} \sum_{j=1}^{M} \sum_{j'=1}^{M} \phi(y_j)^T \phi(y_{j'}). \\
= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} k(x_i, x_{i'}) - \frac{2}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} k(x_i, y_j) + \frac{1}{M^2} \sum_{j=1}^{M} \sum_{j'=1}^{M} k(y_j, y_{j'}).
Maximum Mean Discrepancy (MMD)

• Quadratic kernel:

\[ k(x, y) = \phi(x)^T \phi(y) = (x^T y + c)^2. \]

\[ D_{MMD} = 2c \left\| \frac{1}{N} \sum_{i=1}^{N} x_i - \frac{1}{M} \sum_{j=1}^{M} y_j \right\|^2 + \left\| \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T - \frac{1}{M} \sum_{j=1}^{M} y_j y_j^T \right\|_F^2 \]

• With a quadratic kernel, MMD can measure the distance between two distributions up to their second order stats.

• Multi-RBF kernels:

\[ k(x, y) = \sum_{q=1}^{K} \exp \left( -\frac{1}{2\sigma_q^2} \|x - y\|^2 \right) \]
Consistency Regularization

• Exploit the unlabeled data for domain adaptation by applying data augmentation on them.
• Consistency training is to regularize a network such that the predictions are consistent even if the network’s input is subjected to noise perturbation.
• Achieved by minimizing the KL divergence

$$\mathbb{E}_{q(\hat{x}_{\text{unlab}} | x_{\text{unlab}})} \left[ \text{KL}(p(\theta \mid y \mid x_{\text{unlab}}) \mid \mid p(\theta \mid y \mid \hat{x}_{\text{unlab}})) \right]$$

where $q(\ )$ is a data augmentation transformation, e.g., adding noise or reverb effect.
• We propose minimizing the discrepancy between the embeddings produced by the clean data and the embeddings produced by the augmented data.
Consistency Regularization

• Achieved by minimizing the MMD between target-domain data and unlabeled augmented data:

\[
\beta \cdot D(\mathcal{H}_t, \hat{\mathcal{H}}_t) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} k(h_i^7, h_{i'}^7) - \frac{2}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} k(h_i^7, \hat{h}_j^7) + \frac{1}{M^2} \sum_{j=1}^{M} \sum_{j'=1}^{M} k(\hat{h}_j^7, \hat{h}_{j'}^7)
\]
Experiments

• **Training data for DNN and PLDA:** 4808 speakers from SRE04-10 and Switchboard

• **Consistency Regularization:** SRE16 and SRE18 unlabeled

• **Test data:** SRE16-eval and SRE18-eval-cmn2

• **Kernel of MMD:** 19 RBF kernels with width ranges from $2^{-8} \sigma_m$ to $2^8 \sigma_m$, where $\sigma_m$ is the median pairwise distance from training data.

• **Acoustic vectors:** 23-dim MFCC with mean norm

• **VAD:** Kaldi’s energy-based VAD

• **PLDA adaptation and CORAL:** SRE16 and SRE18 unlabeled

• **Hyperparameters for DNN Objective:** $\alpha = \beta = \lambda = 1$
Experiments

SRE04-10 + Switchboard

SRE16 or SRE18 unlabeled

SRE16 or SRE18 augmented unlabeled
Experiments

- DNN Architecture

<table>
<thead>
<tr>
<th>Layer</th>
<th>Kernel</th>
<th>Channel in × Channel out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1</td>
<td>5,1,1</td>
<td>23 × 512</td>
</tr>
<tr>
<td>Conv2</td>
<td>3,1,2</td>
<td>512 × 512</td>
</tr>
<tr>
<td>Conv3</td>
<td>3,1,3</td>
<td>512 × 512</td>
</tr>
<tr>
<td>Conv4</td>
<td>1,1,1</td>
<td>512 × 512</td>
</tr>
<tr>
<td>Conv5</td>
<td>1,1,1</td>
<td>512 × 1536</td>
</tr>
<tr>
<td>Statistics pooling</td>
<td></td>
<td>1536 × 3072</td>
</tr>
<tr>
<td>FC6</td>
<td>–</td>
<td>3072 × 512</td>
</tr>
<tr>
<td>FC7</td>
<td>–</td>
<td>512 × 512</td>
</tr>
<tr>
<td>Am-softmax</td>
<td>–</td>
<td>512 × N</td>
</tr>
</tbody>
</table>

\[
\mathcal{L}_{AMS} = -\frac{1}{n} \sum_{i=1}^{n} \log \frac{e^{s \cdot (W_{y_i}^T x_i - m)}}{e^{s \cdot (W_{y_i}^T x_i - m)} + \sum_{j=1, j \neq y_i}^{c} e^{s \cdot W_{j}^T x_i}}
\]
Results

<table>
<thead>
<tr>
<th>Adapt Method</th>
<th>SRE16 EER (%)</th>
<th>SRE16 minDCF</th>
<th>SRE18 EER (%)</th>
<th>SRE18 minDCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>WGAN [12]</td>
<td>13.25</td>
<td>0.899</td>
<td>9.59</td>
<td>0.652</td>
</tr>
<tr>
<td>Sup. WGAN [12]</td>
<td>9.59</td>
<td>0.746</td>
<td>8.88</td>
<td>0.619</td>
</tr>
<tr>
<td>LSGAN [21]</td>
<td>11.74</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Our DNN Adapt.</td>
<td>9.03</td>
<td>0.585</td>
<td>8.33</td>
<td>0.520</td>
</tr>
</tbody>
</table>

- All the results are without backend adaptation.
- Our DNN adaptation performs significantly better than the previously proposed methods.
Results

<table>
<thead>
<tr>
<th>Adapt Method</th>
<th>SRE16 EER(%)</th>
<th>SRE16 minDCF</th>
<th>SRE18 EER(%)</th>
<th>SRE18 minDCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our DNN Adapt.</td>
<td>9.03</td>
<td>0.585</td>
<td>8.33</td>
<td>0.520</td>
</tr>
<tr>
<td>CORAL Adapt.</td>
<td>8.49</td>
<td>0.560</td>
<td>8.74</td>
<td>0.553</td>
</tr>
<tr>
<td>PLDA Adapt.</td>
<td>8.55</td>
<td>0.556</td>
<td>8.88</td>
<td>0.563</td>
</tr>
<tr>
<td>Ours+CORAL Adapt.</td>
<td>8.28</td>
<td>0.541</td>
<td>8.13</td>
<td>0.519</td>
</tr>
<tr>
<td>Ours+PLDA Adapt.</td>
<td>8.29</td>
<td>0.546</td>
<td><strong>8.09</strong></td>
<td><strong>0.521</strong></td>
</tr>
</tbody>
</table>

- Combining the proposed method with backend adaptation further improves the performance.
## Results

<table>
<thead>
<tr>
<th>Layer 7</th>
<th>Layer 6</th>
<th>Consis.</th>
<th>EER(%)</th>
<th>DCF</th>
<th>EER(%)</th>
<th>DCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td>×</td>
<td>×</td>
<td>12.02</td>
<td>0.99</td>
<td>11.59</td>
<td>0.72</td>
</tr>
<tr>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>9.79</td>
<td>0.62</td>
<td>9.08</td>
<td>0.58</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>9.63</td>
<td>0.60</td>
<td>8.77</td>
<td>0.55</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>9.03</td>
<td>0.58</td>
<td>8.33</td>
<td>0.52</td>
</tr>
</tbody>
</table>

- Multi-level adaptation significantly improves the performance in both SRE16 and SRE18.
- Consistency regularization also helps.
Conclusions

• Domain mismatch loss can be applied at both both frame-level and utterance-level

• Apply MMD at frame level performs significantly better than at utterance-level alone

• Data augmentation can be utilized in the unlabeled target-domain through consistency regularization.
Utterance- and Frame-level MMD

Utterance-level MMD

Frame-level MMD