JFA as a feature extractor

**JFA vs. i-vectors**
- The text-independent paradigm of i-vector/PLDA has not been successful in text-dependent speaker recognition. The speaker-phrase variability is hard to be confined into a low-dimensional subspace.
- JFA offers the flexibility of conforming the channel effects in a subspace while allowing the speaker-phrase factors to lie on the supervector space.

**Main JFA equation**
\[ S = m + Ux + V\eta + Dz \]  
(1)
- The hidden variable \( \eta \) varies from one recording to another and is intended to model channel effects.
- In text-independent speaker recognition, the term \( Dz \) is usually dropped and speakers are characterized by the low-dimensional vector \( \eta \). Here, we keep all terms.

**Left-to-Right Structure using HMMs**
- JFA can be extended to utterances that are segmented into HMM states.
- Variables can be global \( (x_1, x_2) \) or local \( (x_3, x_4) \) and model either speaker \( (x_1, x_2) \) or channel \( (x_3, x_4) \), \( z \) is the SV that corresponds to a segment.

**Joint-Density Backend**

**An Alternative to PLDA**
- PLDA is not well suited to our set-up (a single vector from all the enrollment utterances, the vectors are channel compensated, a.o.)
- Training: Use "target" trials from the training set \( t \rightarrow \{ c \} \).
- Estimate mean and covariance matrix \( (C) \) are estimated. Assuming zero mean, \( C \) is as follows:

\[ C = E[zt] = \begin{bmatrix} C_{xx} & C_{xz} \\ C_{zx} & C_{zz} \end{bmatrix} \]  
(2)
- Evaluation: \( LR = \frac{NDF(C)}{NDF(C_z)} \).
- \( C_{max} = C_z \) and \( C_{min} \) is obtained by setting \( C_z = 0 \).

**Experimental Results**

**Combining global and local features**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>G</th>
<th>EER (%)</th>
<th>DCF_F</th>
<th>DCF_B</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM-UBM</td>
<td>1</td>
<td>0.287</td>
<td>0.403</td>
<td>2.45</td>
</tr>
</tbody>
</table>

**Score Normalization**

<table>
<thead>
<tr>
<th>V</th>
<th>EER (%)</th>
<th>DCF_F</th>
<th>DCF_B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.63%</td>
<td>0.38</td>
<td>2.64</td>
</tr>
<tr>
<td>3</td>
<td>9.43%</td>
<td>0.44</td>
<td>2.76</td>
</tr>
<tr>
<td>4</td>
<td>9.68%</td>
<td>0.56</td>
<td>2.80</td>
</tr>
</tbody>
</table>

Features for speaker recognition extracted with various types of JFA model. Global v-tiers is trained on NIST MixSet telephone data.

**Fusing systems with different front-ends**

<table>
<thead>
<tr>
<th>Front-end</th>
<th>EER (%)</th>
<th>DCF_F</th>
<th>DCF_B</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLP</td>
<td>1.61%</td>
<td>0.34</td>
<td>2.22</td>
</tr>
<tr>
<td>Fusion</td>
<td>1.97%</td>
<td>0.48</td>
<td>2.58</td>
</tr>
<tr>
<td>PLP</td>
<td>1.62%</td>
<td>0.34</td>
<td>2.22</td>
</tr>
<tr>
<td>Fusion</td>
<td>1.62%</td>
<td>0.34</td>
<td>2.22</td>
</tr>
</tbody>
</table>

**Conclusions & References**

**Conclusions**
- An extension to JFA is proposed so that it can handle utterances that are segmented into HMM states.
- We extract local end global, subspace and supervector size features and use a trainable backend model (JDB) to extract LLRs.
- The method is capable of attaining about 2% EER on a very challenging dataset.

**References**