Frame Discrimination Training of HMMs for Large Vocabulary Speech Recognition

Dan Povey, Phil Woodland

Cambridge University Engineering Department
Overview

Frame Discrimination:

- Discriminative technique developed by Kapadia at CUED
- Objective function related to Maximum Mutual Information (MMI) objective function
- Results on isolated digit and letter recognition were promising

Current work:

- Implement Frame Discrimination for LVCSR
- Computational problem
- Roadmap algorithm developed to speed up computation
- Computation 3 times faster than a previous implementation of MMI using lattices
- Results as good or better than MMI
MMI and Frame Discrimination

MMI:

\[ \mathcal{F}_\lambda = \sum_{r=1}^{R} \log \frac{P_\lambda(O_r | M^{wr}) P(w_r)}{\sum_{\tilde{w}} P_\lambda(O_r | M^{\tilde{w}}) P(\tilde{w})} \]

\[ = \sum_{r=1}^{R} \log \frac{P_\lambda(O_r | M^{wr})}{P_\lambda(O_r | M^{\text{gen}})} \]

Discriminate against general model of speech \( M^{\text{gen}} \) (recognition model).

FD:

\[ \mathcal{F}_\lambda = \sum_{r=1}^{R} \log \frac{P_\lambda(O_r | M^{wr})}{P_\lambda(O_r | \mathcal{N})} \]

- HMM \( \mathcal{N} \) is one state HMM: weighted sum of all states in \( M^{\text{gen}} \)
- Weights derived from Forward-Backward alignment data
Computation for Frame Discrimination training

- Extended Baum-Welch (EBW) update equations used
- Align speech data to HMMs
- Calculation dominated by computation of Gaussian probabilities

For transcription:

- Use Forward-Backward algorithm using beam pruning
- Perhaps only 100 Gaussians per time frame → fast.

For the denominator model $N$:

- Tens of thousands of Gaussians per time frame: impractical
Roadmap algorithm

The problem:

- Tens of thousands of Gaussians— but frame probability may be dominated by two or three

- Algorithm needed to find the most important Gaussians per frame, without calculating them all

The solution:

- Make list of “near” Gaussians for each Gaussian in system → need distance measure

- Navigate between Gaussians to find best set for current input frame

- Like a roadmap...
**Distance measure**

Overlap for one dimension is defined as $\int_{-\infty}^{\infty}$ of the minimum of the two Gaussians:

- Suitable distance measure is $-\log(\text{Overlap})$.
- Sum over all dimensions of diagonal-covariance Gaussians.
Constructing the roadmap

Use distance measure to construct a roadmap.

About 20 links to and from each Gaussian

1. Make a list of most similar Gaussians
   - Start with random list, iteratively improve it

2. Make lists “symmetric”

3. Remove redundant links
Performance of Roadmap algorithm

Evaluating performance:

- If we miss some important Gaussians, the probability of the speech file given the model $\mathcal{N}$ will decrease
- Measure the average decrease in log probability per frame

<table>
<thead>
<tr>
<th></th>
<th>% Gaussians calculated</th>
<th>Loss in log probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roadmap</td>
<td>3.7%</td>
<td>0.004</td>
</tr>
<tr>
<td>VQ</td>
<td>4%</td>
<td>0.3</td>
</tr>
</tbody>
</table>

(For a system with 9,500 Gaussians)
FD Results on Resource Management

- 1,000 word task
- 109 speakers, 4 hours of training data
- decision-tree state-clustered cross-word triphones, 1577 states
- word-pair grammar
- 4 iterations of FD training done starting from ML-trained models

<table>
<thead>
<tr>
<th></th>
<th>Initial Word Error</th>
<th>Final Word Error</th>
<th>Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Mixture</td>
<td>8.44</td>
<td>6.73</td>
<td>20.3 %</td>
</tr>
<tr>
<td>6 mixture</td>
<td>4.10</td>
<td>3.76</td>
<td>8.3 %</td>
</tr>
</tbody>
</table>
FD Results on Wall Street Journal

- 66 hours of training data (WSJ 0+1)
- decision-tree state-clustered cross-word triphones, 6399 states
- 65k word trigram LM

<table>
<thead>
<tr>
<th>Num mix</th>
<th>% WER</th>
<th>% WER reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comps</td>
<td>MLE</td>
<td>FD</td>
</tr>
<tr>
<td>1</td>
<td>14.64</td>
<td>13.14</td>
</tr>
<tr>
<td>2</td>
<td>12.52</td>
<td>11.30</td>
</tr>
<tr>
<td>4</td>
<td>10.96</td>
<td>10.30</td>
</tr>
<tr>
<td>12</td>
<td>9.63</td>
<td>9.42</td>
</tr>
</tbody>
</table>
Conclusions

- FD implemented on a large vocabulary task
- Used roadmap algorithm to greatly reduce computation
- Introduced overlap distance measure between Gaussians
- FD seems to work at least as well as MMI
- FD faster than lattice-based MMI: takes 6 times as long as ML training instead of 15 times as long
Smoothing in EBW equations

\[
\hat{\mu}_{j,m} = \frac{\{\theta_{j,m}^{\text{num}}(O) - \theta_{j,m}^{\text{den}}(O)\} + D \mu_{j,m}}{\{\gamma_{j,m}^{\text{num}} - \gamma_{j,m}^{\text{den}}\} + D}, \\
\hat{\sigma}_{j,m}^2 = \frac{\{\theta_{j,m}^{\text{num}}(O^2) - \theta_{j,m}^{\text{den}}(O^2)\} + D(\sigma_{j,m}^2 + \mu_{j,m}^2)}{\{\gamma_{j,m}^{\text{num}} - \gamma_{j,m}^{\text{den}}\} + D} - \hat{\mu}_{j,m}^2,
\]

- Set D at phone level
- Floor D at max in phone of any \(\gamma_{j,m}^{\text{num}}\) or \(\gamma_{j,m}^{\text{den}}\)
- Improved both objective function and recognition results
Frame Discrimination for LVCSR

- Iteration of FD training
- FD criterion
- Floored at max occ
- Floored at zero
- Accuracy on training set
Frame Discrimination for LVCSR