Minimum Phone Error and I-Smoothing for Improved Discriminative Training

Dan Povey & Phil Woodland

May 2002

Cambridge University Engineering Department

IEEE ICASSP’2002
Overview

- MPE Objective Function
- MPE & Other Discriminative Criteria
- Lattice Implementation of MMIE: Review
- Lattice Implementation of MPE
- Optimising the MPE criterion: Extended Baum-Welch
- I-smoothing for Improved Generalization
- Switchboard Experiments
- Summary & Conclusions
MPE Objective Function

- Maximise the following function:

\[ \mathcal{F}_{\text{MPE}}(\lambda) = \sum_r \frac{\sum_s p_\lambda(O_r|s)^\kappa P(s) \text{RawAccuracy}(s)}{\sum_s p_\lambda(O_r|s)^\kappa P(s)} \]

where \( \lambda \) are the HMM parameters, \( O_r \) the speech data for file \( r \), \( \kappa \) a probability scale and \( P(s) \) the LM probability of \( s \)

- \( \text{RawAccuracy}(s) \) measures the number of phones correctly transcribed in sentence \( s \) (derived from word recognition).
  i.e. the number of correct phones in \( s \) — inserted phones in \( s \)

- \( \mathcal{F}_{\text{MPE}}(\lambda) \) is weighted average of \( \text{RawAccuracy}(s) \) over all \( s \)

- Scale down log-likelihoods by scale \( \kappa \). As \( \kappa \to \infty \), criterion approaches phone accuracy on data

- Criterion is to be maximised, not minimised (for compatibility with MMI)
MPE & Other Discriminative Criteria

- MMI maximises the posterior probability of the correct sentence
  Problem: sensitive to outliers, e.g. mistranscribed or confusing utterances

- MCE maximises a smoothed approximation to the sentence accuracy
  Problem: cannot easily be implemented with lattices; scales poorly to long sentences

- Criterion we evaluate in testing is word error rate: makes sense to maximise something similar to it

- MPE uses smoothed approximation to phone error but can use lattice-based implementation developed for MMI

- Note that MPE is an approximation to phone error in a word recognition context i.e. uses word-level recognition, but scoring is on a phone error basis.

- Can directly maximise a smoothed word error rate → Minimum Word Error (MWE). Performance for MWE slightly worse than MPE, so main focus here on MPE
Lattice Implementation of MMI: Review

- Generate lattices marked with time information at HMM level
  - Numerator (num) from correct transcription
  - Denominator (den) from confusable hypotheses from recognition

- Use Extended Baum-Welch (Gopalakrishnan et al, Normandin) updates e.g. for means

\[
\hat{\mu}_{jm} = \frac{\{\theta_{jm}^{num}(O) - \theta_{jm}^{den}(O)\}}{\{\gamma_{jm}^{num} - \gamma_{jm}^{den}\}} + D
\]

  - Gaussian occupancies (summed over time) are \(\gamma_{jm}\) from forward-backward
  - \(\theta_{jm}(O)\) is sum of data, weighted by occupancy.

- For rapid convergence use Gaussian-specific D-constant

- For better generalisation broaden posterior probability distribution
  - Acoustic scaling
  - Weakened language model (unigram)
Lattice Implementation of MPE

- Problem: RawAccuracy(s), defined on sentence level as
  (\#correct - \#inserted) requires alignment with correct transcription

- Express RawAccuracy(s) as a sum of PhoneAcc(q) for all phones q in the sentence hypothesis s:

  \[
  \text{PhoneAcc}(q) = \begin{cases} 
  1 & \text{if correct phone} \\
  0 & \text{if substitution} \\
  -1 & \text{if insertion} 
  \end{cases}.
  \]

- Calculating PhoneAcc(q) still requires alignment to reference transcription

- Use an approximation to PhoneAcc(q) based on time-alignment information
  - compute the proportion \( e \) that each hypothesis phone overlaps the reference
  - gives a lower-bound on true value of RawAccuracy(s)
### Approximating PhoneAcc using Time Information

PhoneAcc\((q)\) = \(\begin{cases} 
-1 + 2e & \text{if same phone} \\
-1 + e & \text{if different phone} 
\end{cases}\)

<table>
<thead>
<tr>
<th>Reference</th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypothesis</td>
<td>a</td>
<td>b</td>
<td>b</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Proportion (e)</th>
<th>1.0</th>
<th>0.8</th>
<th>0.2</th>
<th>0.15</th>
<th>0.85</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1 + (\text{correct:}2e, \text{incorrect:}e))</td>
<td>1.0</td>
<td>0.6</td>
<td>-0.6</td>
<td>-0.85</td>
<td>-0.15</td>
</tr>
<tr>
<td>Max of above</td>
<td>1.0</td>
<td>0.6</td>
<td>-0.6</td>
<td>-0.15</td>
<td></td>
</tr>
</tbody>
</table>

Approximated sentence raw accuracy from above = 0.85

Exact value of raw accuracy: 2 corr – 1 ins = 1
PhoneAcc Approximation For Lattices

Calc PhoneAcc(q) for each phone q, then find $\frac{\partial F_{MPE}(\lambda)}{\partial \log p(q)}$ (forward-backward)
Applying Extended Baum-Welch to MPE

- Use EBW update formulae as for MMI but with modified MPE statistics

- For MMI, the occupation probability for an arc \( q \) equals \( \frac{1}{\kappa} \frac{\partial F_{\text{MMIE}}(\lambda)}{\partial \log p(q)} \) for numerator (\( \times -1 \) for the denominator). The denominator occupancy-weighted statistics are subtracted from the numerator in the update formulae.

- Statistics for MPE update use \( \frac{1}{\kappa} \frac{\partial F_{\text{MPE}}(\lambda)}{\partial \log p(q)} \) of the criterion w.r.t. the phone arc log likelihood which can be calculated efficiently.

- Either MPE numerator or denominator statistics are updated depending on the sign of \( \frac{\partial F_{\text{MPE}}(\lambda)}{\partial \log p(q)} \), which is the “MPE arc occupancy”

- After accumulating statistics, apply EBW equations.

- EBW is viewed as a gradient descent technique and can be shown to be a valid update for MPE.
Improved Generalisation using I-smoothing

- Use of discriminative criteria can easily cause over-training

- Get smoothed estimates of parameters by combining Maximum Likelihood (ML) and MPE objective functions for each Gaussian

- Rather than globally interpolate (H-criterion), amount of ML depends on the occupancy for each Gaussian

- I-smoothing adds $\tau$ samples of the average ML statistics for each Gaussian. Typically $\tau = 50$.
  - For MMI scale numerator counts appropriately
  - For MPE need ML counts in addition to other MPE statistics

- I-smoothing essential for MPE (& helps a little for MMI)
Experimental Setup

- PLP cepstral features + first/second derivatives (39 dims)
- Cepstral mean/variance normalisation
- Vocal tract length normalisation
- Training on h5train00 (265 hours) or h5train00sub (68 hours)
- Decision tree-clustered triphone HMMs with 6165 states
  - 16 mix comps for h5train00
  - 12 mix comps for h5train00sub
- Testing on 1998 Hub5 evaluation data: about 3 hours (Swbd2/Call Home)
- Need more training iterations for MPE than MMI (e.g. 8 vs 4)
## Switchboard Results (I)

<table>
<thead>
<tr>
<th>Method</th>
<th>% WER Train</th>
<th>% WER eval98</th>
<th>% WER redn (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>41.8</td>
<td>46.6</td>
<td>–</td>
</tr>
<tr>
<td>MMIE</td>
<td>30.1</td>
<td>44.3</td>
<td>2.3</td>
</tr>
<tr>
<td>MMIE ($\tau=200$)</td>
<td>32.2</td>
<td>43.8</td>
<td>2.8</td>
</tr>
<tr>
<td>MPE ($\tau=50$)</td>
<td>27.9</td>
<td>43.1</td>
<td>3.5</td>
</tr>
</tbody>
</table>

HMMs trained on h5train00sub (68h train). Train use lattice unigram

<table>
<thead>
<tr>
<th>Method</th>
<th>% WER Train</th>
<th>% WER eval98</th>
<th>% WER redn (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE baseline</td>
<td>47.2</td>
<td>45.6</td>
<td>–</td>
</tr>
<tr>
<td>MMIE</td>
<td>37.7</td>
<td>41.8</td>
<td>3.8%</td>
</tr>
<tr>
<td>MMIE ($\tau=200$)</td>
<td>35.8</td>
<td>41.4</td>
<td>4.2%</td>
</tr>
<tr>
<td>MPE ($\tau=100$)</td>
<td>34.4</td>
<td>40.8</td>
<td>4.8%</td>
</tr>
</tbody>
</table>

HMMs trained on h5train00 (265h train). Train is lattice unigram

- I-smoothing reduces the error rate with MMI by 0.3-0.4% abs
- MPE/I-smoothing gives around 1% abs lower WER than previous MMI results
Switchboard Results (II)

<table>
<thead>
<tr>
<th></th>
<th>% WER Train</th>
<th>% WER eval98</th>
<th>% WER redn (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>41.8</td>
<td>46.6</td>
<td>-</td>
</tr>
<tr>
<td>MPE ($\tau = 0$)</td>
<td>28.5</td>
<td>50.7</td>
<td>-4.1%</td>
</tr>
<tr>
<td>MPE ($\tau = 25$)</td>
<td>27.9</td>
<td>43.1</td>
<td>3.5%</td>
</tr>
<tr>
<td>MWE ($\tau = 25$)</td>
<td>25.9</td>
<td>43.3</td>
<td>3.3%</td>
</tr>
</tbody>
</table>

HMMs trained on h5train00sub (68h train). Train use lattice unigram

- Training set WER reduces with/without I-smoothing
- I-smoothing essential for test-set gains with MPE
- Minimum Word Error (MWE) better than MPE on train
- MWE generalises less well than MPE
Summary & Conclusions

• Introduced MPE (& MWE) to give error-rate based discriminative training
  – Less affected by outliers than MMI-based training
  – Smoothed approximation to phone error in word recognition system
  – Approximate reference-hypothesis alignment
  – Use same lattice-based training framework developed for MMI
  – Compute suitable MPE statistics so still use Extended Baum-Welch update
  – Use I-smoothing to improve generalisation (essential for MPE)

• MPE/I-smoothing reduces WER over previous MMI approach by 1% abs

• MPE used for CU-HTK April 2002 Switchboard evaluation system