# Minimum Phone error and I-Smoothing for improved Discriminative Training

Dan Povey & Phil Woodland

May 8th 2001

Cambridge University Engineering Department

# Overview

- Minimum Phone Error (MPE)
  - General introduction.
  - MPE objective function.
  - Comparison with other discriminative objective functions.
- Lattice implementation of MPE.
- Optimising the MPE criterion with the EB formulae.
- Improving generalization: I-smoothing etc.
- MPE and MMI results on Switchboard (hub5), up to 265 hours training.
- Conclusions

# Minimum Phone Error

- Minimum Phone Error (MPE) is a new criterion for discriminative criterion.
- Can give better results than MMI.
- CU-HTK submission for the 2002 Switchboard (hub5) evaluation will use MPE.
- Training time and complexity of implemetation not much greater than MMIE.

# MPE Objective Function

- Maximise the following function:

$$\mathcal{F}_{\mathrm{MPE}}(\lambda) = \sum_{r}^{R} \frac{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa \mathrm{RawAccuracy}(s)}{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa}$$

  where $\lambda$ are the HMM parameters, $\mathcal{O}_r$ the speech data for file $r$, $\kappa$ a probability scale and $P(s)$ the language model probability pre-scaled by the normal scale factor.

- $\mathrm{RawAccuracy}(s)$ is a measure of the number of phones correctly transcribed in sentence $s$.
  (correct phones in s $-$ inserted phones in s).

- Weighted average of $\mathrm{RawAccuracy}(s)$ over all $s$.

- As $\kappa \to \infty$, approaches phone error on data.

# MPE & Other Discriminative Objective Functions

- MPE function is an average (weighted by sentence likelihood) of a measure of phone accuracy:

$$\mathcal{F}_{\mathrm{MPE}}(\lambda) = \sum_{r}^{R} \frac{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa \mathrm{RawAccuracy}(s)}{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa}$$

- Objective function in MMIE is the probability of the correct utterance given the speech data:

$$\mathcal{F}_{\mathrm{MMIE}}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda\left(\mathcal{O}_r|\mathcal{M}_{s_r}\right)^\kappa P(s_r)^\kappa}{\sum_s p_\lambda\left(\mathcal{O}_r|\mathcal{M}_s\right)^\kappa P(s)^\kappa}$$

- MCE (Minimum Classification Error) objective function is a differentiable approximation to the sentence error rate.

- MWE/MPE objective functions closest to what we want– the word error rate.

# Lattice implementation of MPE

- Implement in a lattice framework, for efficiency (as MMIE).

- $\mathrm{RawAccuracy}(s)$, defined on sentence level, requires expensive dynamic programming.

- Express $\mathrm{RawAccuracy}(s)$ as a sum of $\mathrm{PhoneAcc}(p)$ for all phones in the sentence:

$$\mathrm{PhoneAcc}(p) = \left\{ \begin{array}{l} 1 \text{ if correct phone} \\ 0 \text{ if substitution} \\ -1 \text{ if insertion} \end{array} \right\}.$$

- Calculating $\mathrm{PhoneAcc}(p)$ is still hard .

- Use an approximation to $\mathrm{PhoneAcc}(p)$ based on time-alignment information.

# Optimising the MPE criterion with EB

- Use Extended Baum-Welch (EB) update as in MMI.
- Use two sets of statistics (numerator and denominator) as in MMI.
- Data from each phone $q$ goes in numerator or denominator statistic depending on sign of $\frac{\partial \mathcal{F}_{\mathrm{MPE}}(\lambda)}{\partial \log p(q)}$.
- EB is viewed as a gradient descent technique and can be shown to be a valid update for MPE.
- Up to twice as many iterations of training as MMI to reach best error rates: 8 iterations of instead of 4.

# Improving generalisation using I-smoothing

- H-criterion is $h\mathcal{F}_{\mathrm{MMIE}}(\lambda) + (1-h)\mathcal{F}_{\mathrm{ML}}(\lambda)$ (Backoff between MMIE and MLE).

- I-smoothing (for MMI) is like H-criterion except proportionof MMI (i.e., h) varies depending on the amount of data for each Gaussian.

- In effect, it is like having $\tau$ points of extra MLE data for each Gaussian (do this by scaling up the normal MLE counts before updating Gaussian). Use say $\tau = 100$.

- For MMIE, I-smoothing gives an improvement on some tasks (no improvement over MMIE on others).

- For MPE, I-smoothing makes a lot of difference; without I-smoothing, MPE gives little improvement.

# Improving generalisation: other issues

- Use unigram language model in training (as for MMI).
- Set the probability scale $\kappa$ to the inverse of the normal language model scale factor (as for MMI).
- Use phones not words to calculate accuracy– so MPE not MWE.

# Experimental setup on Switchboard.

- HTK large vocabulary recognition system
- PLP cepstral features + first/second derivatives (39 dimensions in total).
- Training on h5train00 (265 hours) or h5train00sub (68 hours)
- HMM sets with tree-clustered triphone context-dependent states: 6165 HMM states, and 12 or 16 Gaussians/state.
- Testing on eval98

# Results on Switchboard.

Results trained on h5train00sub (68h train)

| | WER Train | WER Test eval98 | Abs test improvement |
|---|---|---|---|
| MLE | 26.3 | 46.6 | – |
| MMIE | 18.6 | 44.3 | 2.3% |
| MMIE+I-smoothing | 19.7 | 43.8 | 2.8% |
| MPE+I-smoothing | 20.6 | 43.1 | 3.5% |

Results trained on h5train00sub (68h train)

| | WER Train | WER Test eval98 | Abs test improvement |
|---|---|---|---|
| MLE baseline | 30.1 | 45.6 | – |
| MMIE | 23.2 | 41.8 | 3.8% |
| MMIE+I-smoothing | 22.2 | 41.4 | 4.2% |
| MPE+I-smoothing | 23.9 | 40.8 | 4.8% |

# Conclusions.

- MPE training gives good improvements, up to about 5% absolute on Switchboard.
  - MPE currently being used in Cambridge University Hub5 evaluation system (2002).
- MPE can be efficiently implemented using lattices.
  - Get around need for dynamic programming by approximating the phone accuracy.
  - Use EB formulae with same setup as MMI, for fast optimisation.