Minimum Phone Error – better than MMI

Dan Povey

Jan 30th 2003

Cambridge University Engineering Department

Summary

Four sections (each followed by questions):

- Introduce MPE, give results on various corpora
- Theory of optimisation (MMI)
- Theory of optimisation (MPE)
- Practical issues for implementation

Minimum Phone Error (MPE)

- Maximise the following function:

$$\mathcal{F}_{\mathrm{MPE}}(\lambda) = \sum_{r}^{R} P_\lambda(s|\mathcal{O}_r)\mathrm{RawPhoneAccuracy}(s, s_r)$$

- i.e. an average of phone accuracy, weighted by sentence likelihood

- where $\mathrm{RawPhoneAccuracy}(s, s_r)$ is #phones in reference, minus #phone errors

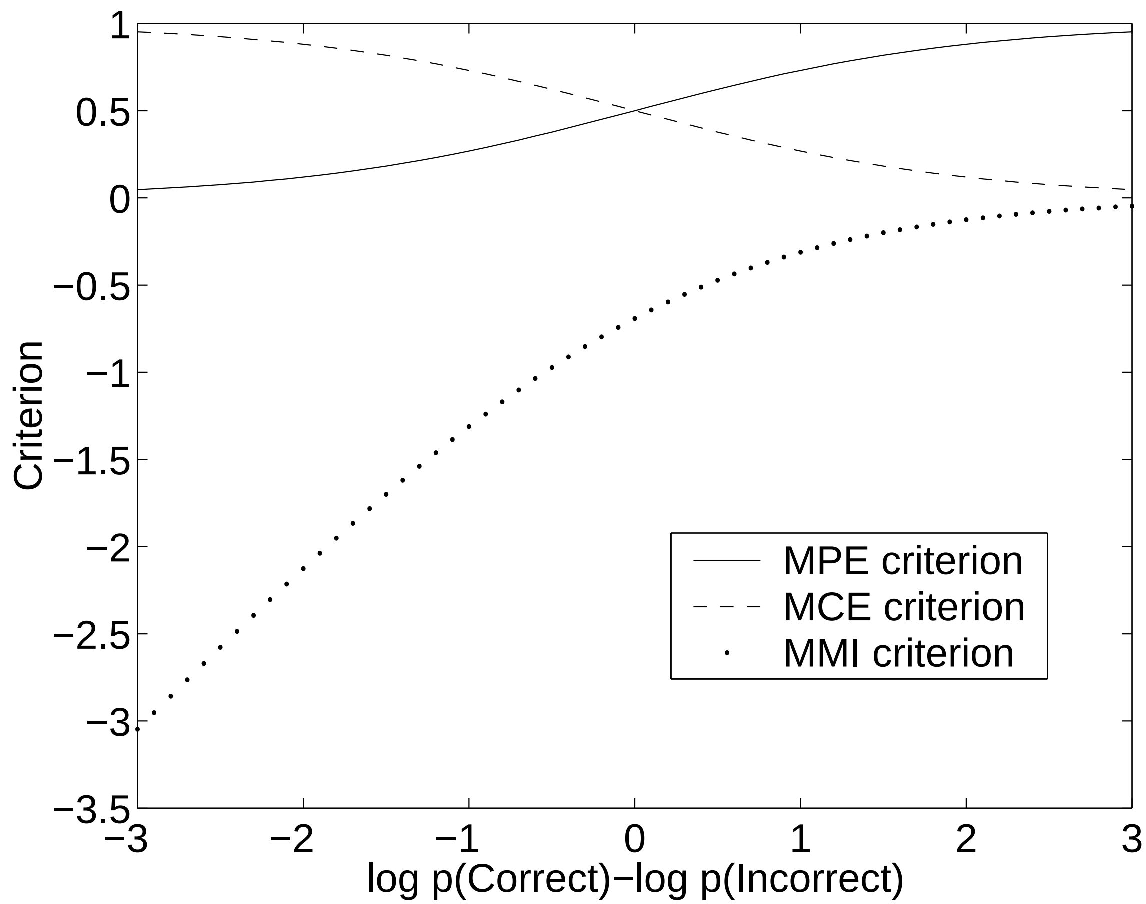
Maximum Mutual Information (MMI)

- $\mathcal{F}_{\mathrm{MMIE}}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(\mathcal{O}_r|s_r)^\kappa P(s_r)^\kappa}{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa}$
- Equals posterior probability of correct sentence given data & HMM

Comparison of objective functions

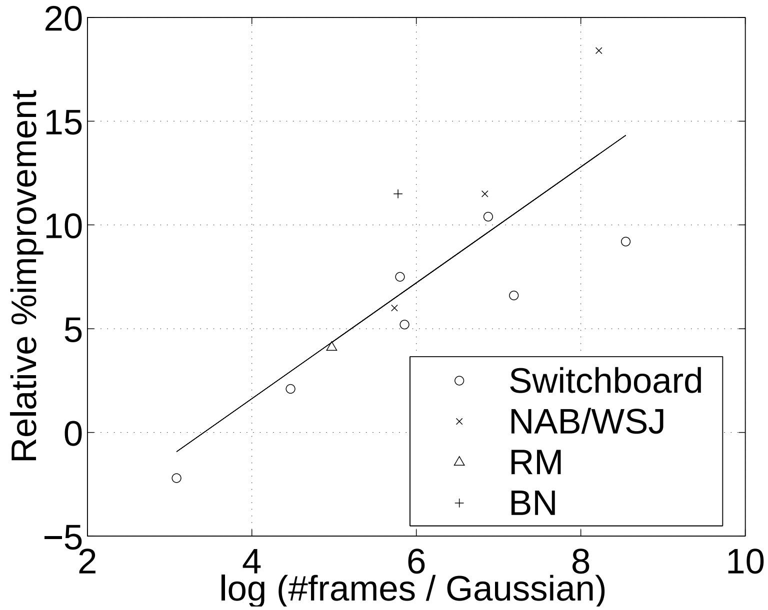
- Suppose correct sentence is "a", only alternative is "b".
- Let $a = p_\lambda(\mathcal{O}|\text{"a"})P(\text{"a"})$ (acoustic & LM likelihood), b is same for "b".
- ML objective function $= \log(a) +$ other training files.
- MMI objective function $= \log(\frac{a}{a+b}) +$ other training files.
- MPE objective function $= \frac{a \times 1 + b \times 0}{a+b} +$ other training files.
- MCE objective function $= \frac{a^\kappa \times 1 + b^\kappa \times 0}{a^\kappa + b^\kappa}$
- Difference is not so simple for more complex examples

Comparison of objective functions (cont'd)



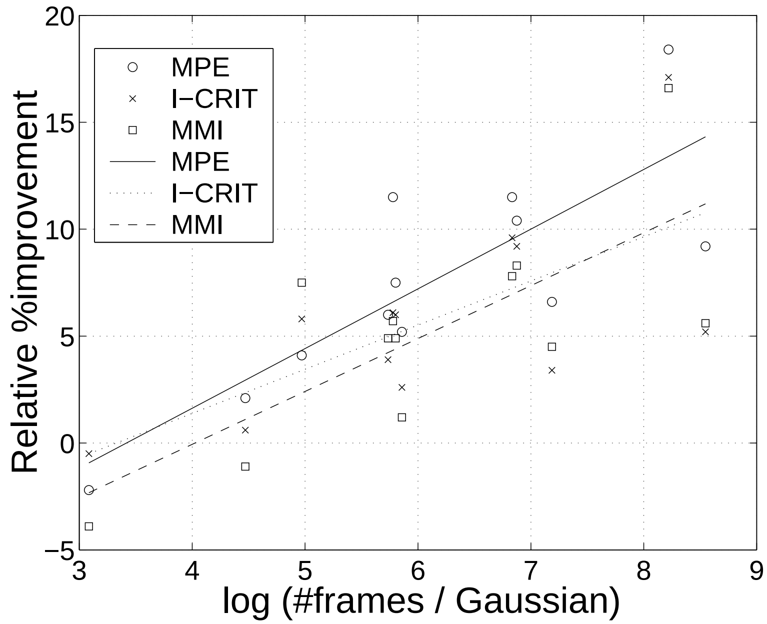
Improvement vs. ML

On baseline systems on various corpora (no MLLR), relative improvement of MPE vs ML:



Comparison of MPE with MMI, I-smoothed MMI

(I-smoothing is use of priors, will describe later)



Combination with other techniques

- On a baseline evaluation systems, typically about 10-12% relative improvement
- MLLR, HLDA, SAT, gender adaptation, improve absolute results
- ... but each tends to reduce the relative improvement due to MPE
- With all bells and whistles, perhaps 5% relative improvement from MPE
- (Depends on #Gaussians in HMM set)
- We are using MPE in evaluations
- ... cannot release HTK code before we release a recogniser which can produce phone-marked lattices

Questions

?

Auxiliary functions

- Definitions:

- $\mathcal{G}(\lambda, \lambda')$ is a *strong-sense auxiliary function* for $\mathcal{F}(\lambda)$ *around* λ', iff $\mathcal{G}(\lambda, \lambda') - \mathcal{G}(\lambda', \lambda') \leq \mathcal{F}(\lambda) - \mathcal{F}(\lambda')$,

- $\mathcal{G}(\lambda, \lambda')$ is a *weak-sense auxiliary function* for $\mathcal{F}(\lambda)$ *around* λ', iff

$$\frac{\partial}{\partial \lambda}\mathcal{G}(\lambda, \lambda')\Big|_{\lambda=\lambda'} = \frac{\partial}{\partial \lambda}\mathcal{F}(\lambda)\Big|_{\lambda=\lambda'}.$$

Auxiliary functions cont'd

(a) (b)

Use of (a) strong-sense and (b) weak-sense auxiliary functions for function optimisation

- Strong-sense auxiliary function: has the same value as real objective function at a local point $\lambda = \lambda'$, but $\leq$ objf everywhere else
- Weak-sense auxf has same differential around local point $\lambda = \lambda'$

Auxiliary functions & function maximisation

- Strong-sense auxiliary functions give a guarantee of convergence.
- A weak-sense auxiliary function does not give such a guarantee ...
- ... but if it *does* converge it will converge to a local maximum (only fixed point of update)
- Similar level of guarantee to gradient descent (which will only converge for correct speed of optimisation)
- But freer than gradient descent in functional form of update
- Useful where "natural" form of parameters is not a normal linear variable, but, say, a variance matrix or a probability, etc

Optimising Gaussian likelihoods

- Normal auxiliary function for ML is $\sum_{j=1}^{J}\sum_{m=1}^{M} -0.5\left(\gamma_{jm}\log\sigma_{jm}^2 + \frac{\theta_{jm}(\mathcal{O}^2)-2\mu_{jm}\theta_{jm}(\mathcal{O})-\gamma_{jm}\mu_{jm}^2}{\sigma_{jm}^2}\right)$ where γ_{jm}, $\theta_{jm}(\mathcal{O})$ and $\theta_{jm}(\mathcal{O}^2)$ are occupancy, sum of data & data squared for mix m of state j.

- Abbreviate this to $\sum_{j=1}^{J}\sum_{m=1}^{M} Q(\gamma_{jm}, \theta_{jm}(\mathcal{O}), \theta_{jm}(\mathcal{O}^2)|\mu_{jm}, \sigma_{jm}^2)$.

- $Q(t, X, Y|\mu, \sigma)$ is log-likelihood of t points of data with sum X and s-o-s Y, given μ, σ

- For MMI, objective function is $p_\lambda(\mathcal{O}|\mathcal{M}_{\rm num}) - p_\lambda(\mathcal{O}|\mathcal{M}_{\rm den})$

- ... and a valid *weak-sense* auxiliary function for objf is $\sum_{j=1}^{J}\sum_{m=1}^{M} Q(\gamma_{jm}^{\rm num}, \theta_{jm}^{\rm num}(\mathcal{O}), \theta_{jm}^{\rm num}(\mathcal{O}^2)|\mu_{jm}, \sigma_{jm}^2)$ $-Q(\gamma_{jm}^{\rm den}, \theta_{jm}^{\rm den}(\mathcal{O}), \theta_{jm}^{\rm den}(\mathcal{O}^2)|\mu_{jm}, \sigma_{jm}^2)$.

Optimising Gaussian likelihoods cont'd

- In order to make sure aux function is convex, add
- $\sum_{j=1}^{J} \sum_{m=1}^{M} Q(D_{jm}, D_{jm}\mu_{jm}, D_{jm}({\mu'}_{jm}^{2} + {\sigma'}_{jm}^{2}) | \mu_{jm}, \sigma_{jm}^{2})$.
- This has zero differential where $\mu_{jm} = \mu'_{jm}, \sigma_{jm}^{2} = {\sigma'}_{jm}^{2}$
- Adding this does not change the gradient where $\lambda = \lambda'$,
- ... so objective function is still weak-sense objective function for MMI objective function

Optimising Gaussian likelihoods cont'd

- Solving this leads to the Extended Baum-Welch update equations, e.g. (for the mean): $\mu_{jm} = \frac{\left\{\theta^{\mathrm{num}}_{jm}(\mathcal{O}) - \theta^{\mathrm{den}}_{jm}(\mathcal{O})\right\} + D_{jm}\mu'_{jm}}{\left\{\gamma^{\mathrm{num}}_{jm} - \gamma^{\mathrm{den}}_{jm}\right\} + D_{jm}}$

- For good convergence set D_{jm} to $E\gamma^{\mathrm{den}}_{jm}$ for e.g. $E = 1$ or 2

- Note– optimisation technique affects recognition results independently of criterion value

Questions

?

Optimising MPE objective function

- Lattice likelihood computation for MMI & MPE uses fixed start & end points for phone models
- For phone arcs q in the lattice
- … use intermeduate [weak-sense] auxiliary function which is linear expansion of MPE objective function in terms of log-likelihoods $\log p(q)$, around current parameter values
- (where $p(q)$ is a shorthand for the acoustic likelihood for arc q)
- $\mathcal{H}_{\mathrm{MPE}}(\lambda, \lambda') = \sum_{r=1}^{R} \sum_{q=1}^{Q_r} \frac{\partial \mathcal{F}_{\mathrm{MPE}}}{\partial \log p(q)} \Big|^{(\lambda=\lambda')} \log p(q)$

Optimising MPE objective function

- This is very similar to the MMI objective function, separate out +ve and -ve terms:

-

$$\begin{aligned}\mathcal{H}_{\mathrm{MPE}}(\lambda, \lambda') = & \sum_{r=1}^{R} \sum_{q=1}^{Q_r} \max(0, \frac{\partial \mathcal{F}_{\mathrm{MPE}}}{\partial \log p(q)}|^{(\lambda=\lambda')}) \log p(q) \\ - & \sum_{r=1}^{R} \sum_{q=1}^{Q_r} \max(0, -\frac{\partial \mathcal{F}_{\mathrm{MPE}}}{\partial \log p(q)}|^{(\lambda=\lambda')}) \log p(q),\end{aligned}$$

- Since $\mathcal{H}_{\mathrm{MPE}}(\lambda, \lambda')$ is basically the same form as MMI objf, a weak-sense auxiliary function for it is known

- Leads to Extended Baum-Welch equations

Optimising MPE objective function, cont'd

- Important definition $\gamma_q^{\mathrm{MPE}} = \frac{1}{\kappa}\frac{\partial \mathcal{F}_{\mathrm{MPE}}}{\partial \log p(q)}$,
- which is scaled differantial of objective function w.r.t. log likelihood of arc.
- ... Trivial to calculate sufficient statistics once this is calculated
- Can be calculated as: $\gamma_q^{\mathrm{MPE}} = \gamma_q(c(q) - c_{\mathrm{avg}})$, where
  - γ_q is the occupation probability of the arc (as in MLE),
  - $c(q)$ is average correctness of sentences including arc q, and
  - c_{avg} is average correctness of sentences in the speech file

Optimising MPE objective function, cont'd

- Calculations specific to MPE are to calculate $c(q)$ and c_{avg}.
- Two techniques used: approximate and exact
- Similar in terms of performance and time taken
- Approximate one is simpler to implement

Approximate MPE

- Function $\mathrm{RawPhoneAccuracy}(s, s_r)$ equals #phones in s_r minus #errors
- which equals # correct phones - # insertions:
- $=$ sum over phones q in sentence s, of

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

- Approximation: if q overlaps with a reference (time-marked) phone z, and extent of overlap as proportion of extent of phone z is $0 \leq e \leq 1$,
- ... use $\mathrm{PhoneAcc}(q) = \left\{ \begin{array}{l} -1 + 2e \text{ if same phone} \\ -1 + e \text{ if different phone} \end{array} \right\}.$
- Tradeoffs between insertion & correct phone, and insertion & deletion, respectively

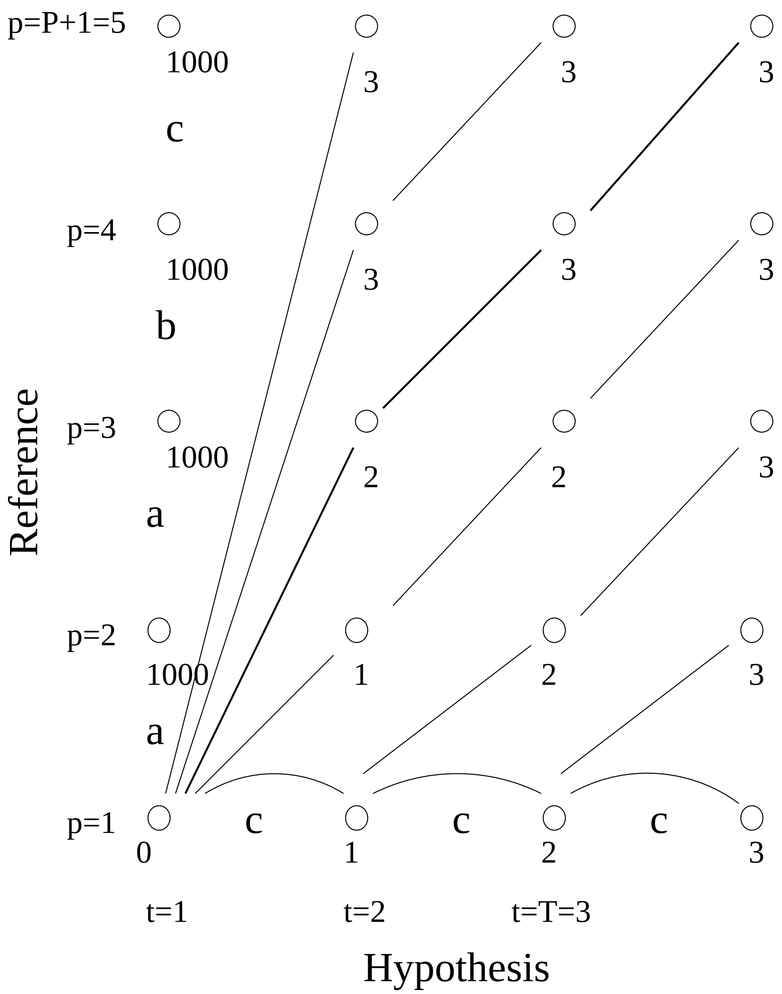
Approximate MPE cont'd

- Easy to integrate each phone's contribution into a forward-backward like algorithm to calculate $c(q)$ for each q

Exact MPE

- Doesn't rely on time alignment of reference transcription, except for optimisation
- Consider a single hypothesis sentence (i.e. single sentence in the lattice) to illustrate this
- If reference transcription has P phones,
- View each hypothesis phone as $P+1$ separate arcs, depending on position...
- Use a traceback algorithm to find alignment

Exact MPE cont'd



Exact MPE cont'd

- This traceback algorithm can be formulated in terms of transition probabilities
- ... so only the traced-back path has a nonzero probability
- This makes it possible to integrate the traceback into a forward-backward type of algorithm
- Forward-backward algorithm can be done for lattice

Optimisation schedule

- As MMI, generate lattice just once (at start)
- ... and use E=2 to control optimisation speed
- Generally more iterations than MMI to reach optimum WER (e.g. 6-8 vs. 4 for MMI)

Questions?

I-smoothing

- I-smoothing is the use of a prior over the Gaussian parameters
- Log prior distribution is $Q(\tau^I, \tau^I \mu_{\mathrm{prior}}, \tau^I(\sigma^2_{\mathrm{prior}} + \mu^2_{\mathrm{prior}}) | \mu, \sigma^2)$
- ... which is log likelihood of τ^I (e.g. 50) points from the distribution $(\mu_{\mathrm{prior}}, \sigma^2_{\mathrm{prior}})$
- ... which is set to the ML estimates of the Gaussian parameters
- Very important if MPE is to give test-set improvements!
- (Named in reference to H-criterion, turned out not to be a criterion)

Other implementation issues

- Implementation issues have mostly been tested on 3 corpora (Switchboard, BN, WSJ)
- Probability scale κ: best value is generally the inverse of the normal LM scale, i.e. generally in range $\frac{1}{20} \ldots \frac{1}{10}$
- Optimisation speed: constant E controls optimisation, $E = 2$ is generally good, 8 iterations
- (Remember– optimisation affects recognition independently of criterion value)
- I-smoothing: $\tau = 50$ generally the best value
- ...

Other implementation issues

- Language model in lattices: unigram better than bigram, zero-gram
- ... note that we use bigram to generate actual words in lattice, haven't tried unigram
- MPE better than Minimum Word Error (calculate error on a word level)
- Approximate and exact MPE give similar results
- Training lattices can be generated just once but need to be big enough, e.g. beam ¿150 or so (MPE is less sensitive to this than MMI)

Full covariances

- Optimisation technique is trivial to extend to full covariances: just use vectors not scalars in update equations
- MPE also gives improvements for full-covariance systems
- Relative improvement from MPE is less, but absolute results better than the best diagonal-covariance systems
- We don't do this because we don't have working full-cov MLLR
- To get this to work, it's necessary to
  - Use a larger value of τ^I (for smoothing) for variance, e.g. 50$\rightarrow$500
  - Smooth off-diagonal parts of the ML variance estimates which form center of prior
- Should be easy to combine this with EMLLT etc.

Transforms and MPE

- (Not published or in my PhD)
- MPE can be used to train HLDA transforms or “semitied” transforms
- Two separate cases:
  - Assume parameters are ML-trained; train transform to maximise MPE criterion (can do MPE training later)
  - Assume parameters are MPE-trained; train transform to maximise MPE criterion
- First case gave impressive gains on 1 Gauss-state WSJ system, but didn’t work well on larger systems. With MMI it gave a degradation
- Second case easier, I think it can help but I haven’t done proper comparisons

Questions

?

Grand Theory of Speech Recognition

- Is complexity (of the system) good or bad?
- I believe complexity is potentially good in terms of recognition rate
- Suppose the recognition system is described in N bits
- Let $Best(N)$ equal the WER of the best speech recognition system that can be described in N bits
- What does the function $Best(N)$ look like?

Grand Theory of Speech Recognition (2)

- Surely an increasing function!
- In simple problems like factorisation you might expect an optimum system, so $Best(N)$ would saturate at some N...
- I don't believe there is an optimum system for speech
- What if N needs to be very large?
- $\rightarrow$ complexity management!

Grand Theory of Speech Recognition (3)

- Want to increase N (have many adjustable parameters, bits that can be played with or added on, etc..)
- But don't want to lose control of the code base

Grand Theory of Speech Recognition (4)

Potential solution...

- Have parts of the system defined by some kind of code, or by numerical parameters
- So parts of the system description are not human-understandable, they are just numbers
- Issue of being able to implement them does not arise, just have to duplicate them
- Evolution, DNA, etc...
- I favour neural-net type architectures but with many different “neuron-types” with adjustable properties