Sequence discriminative training

Peter Bell

Automatic Speech Recognition – ASR Lecture 13
2 March 2020

Recall: Maximum likelihood estimation of HMMs

- Maximum likelihood estimation (MLE) sets the parameters so as to maximize an objective function F_{MLE}:

$$F_{\text{MLE}} = \sum_{u=1}^{U} \log P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W_u))$$

for training utterances $\mathbf{X}_1 \ldots \mathbf{X}_U$ where W_u is the word sequence given by the transcription of the uth utterance, $\mathcal{M}(W_u)$ is the corresponding HMM, and λ is the set of HMM parameters

MLE – Updating the mean

- Update equation for the mean vector $\boldsymbol{\mu}_{jm}$ for Gaussian component m of GMM associated with state j is:

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\sum_{u=1}^{U} \sum_{t=1}^{T} \gamma_{jm}^{u}(t) \boldsymbol{x}_t^u}{\sum_{u=1}^{U} \sum_{t=1}^{T} \gamma_{jm}^{u}(t)}$$

  where $\gamma_{jm}^{u}(t)$ is the probability of the model occupying mixture component m of state j at time t given training sentence $\mathbf{X}_u$.

- Some extra notation:

$$\Theta_{jm}^{u}(\mathcal{M}) = \sum_{t=1}^{T} \gamma_{jm}^{u}(t) \boldsymbol{x}_t^u \qquad \Gamma_{jm}^{u}(\mathcal{M}) = \sum_{t=1}^{T} \gamma_{jm}^{u}(t)$$

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\sum_{u=1}^{U} \Theta_{jm}^{u}(\mathcal{M}(W_u))}{\sum_{u=1}^{U} \Gamma_{jm}^{u}(\mathcal{M}(W_u))}$$

Problems with MLE

- Maximum likelihood is only optimal under model correctness assumptions, BUT:
- Observations are absolutely not conditionally independent, given the hidden state
- When states are phone-based, observations are not independent of past/future phone states, given the current state
- ... even when we incorporate phonetic context in the state space, or augment the feature vector

A quote

"[O]ur knowledge about speech is at such a primitive stage that if we are not to be completely devastated by the problem of having too many free parameters then any model of an informative observation sequence will have to be based on some invalid assumptions. This led us to an investigation of an alternative to MLE, MMIE, which does not derive its raison d'etre from an implicit assumption of model correctness."

Peter Brown, 1987

Maximum mutual information estimation

- Maximum mutual information estimation (MMIE) aims to directly maximise the posterior probability (sometimes called conditional maximum likelihood). Using the same notation as before, with $P(W)$ representing the language model probability of word sequence W:

$$
\begin{aligned}
F_{\text{MMIE}} &= \sum_{u=1}^{U} \log P_\lambda(\mathcal{M}(W_u) \mid \mathbf{X}_u) \\
&= \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W_u))P(W_u)}{\sum_{w'} P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W'))P(W')}
\end{aligned}
$$

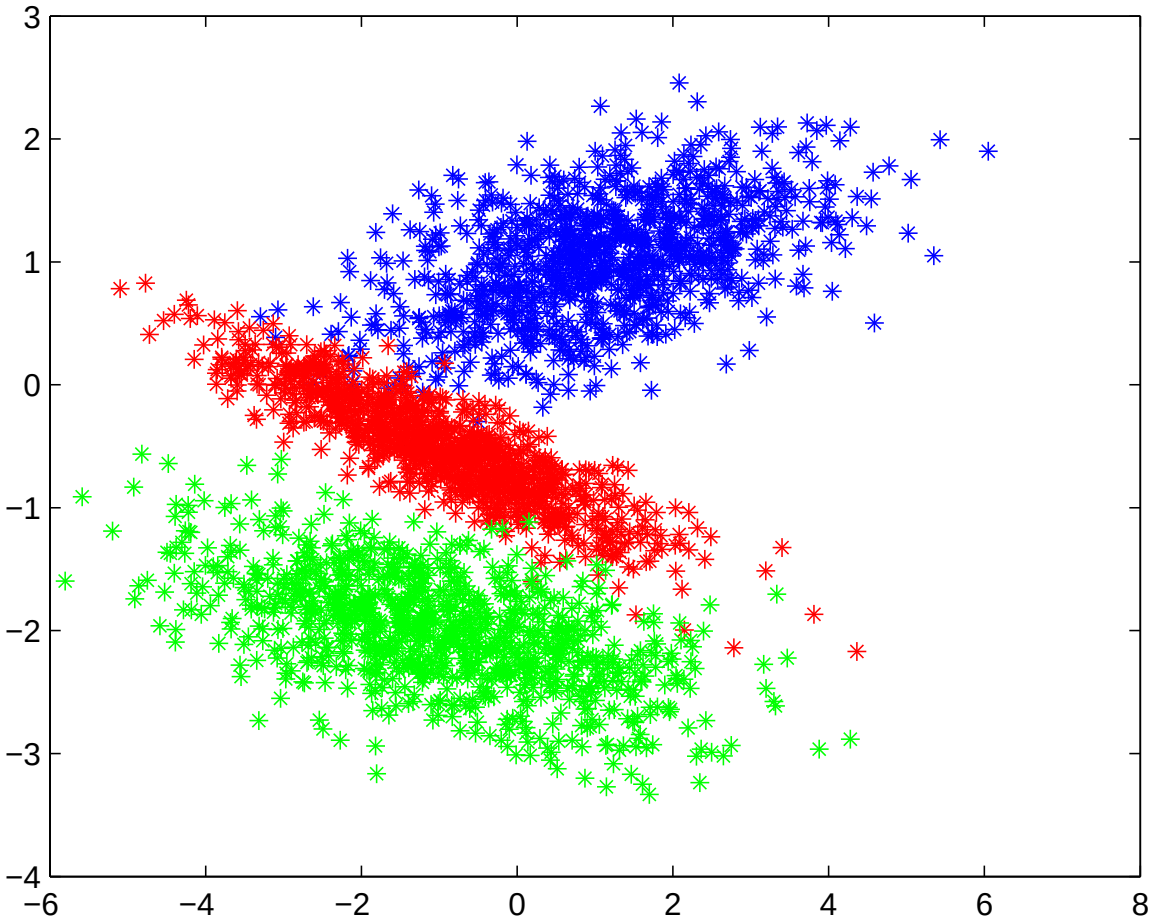
Maximum mutual information estimation

- Maximum mutual information estimation (MMIE) aims to directly maximise the posterior probability (sometimes called conditional maximum likelihood). Using the same notation as before, with $P(W)$ representing the language model probability of word sequence W:

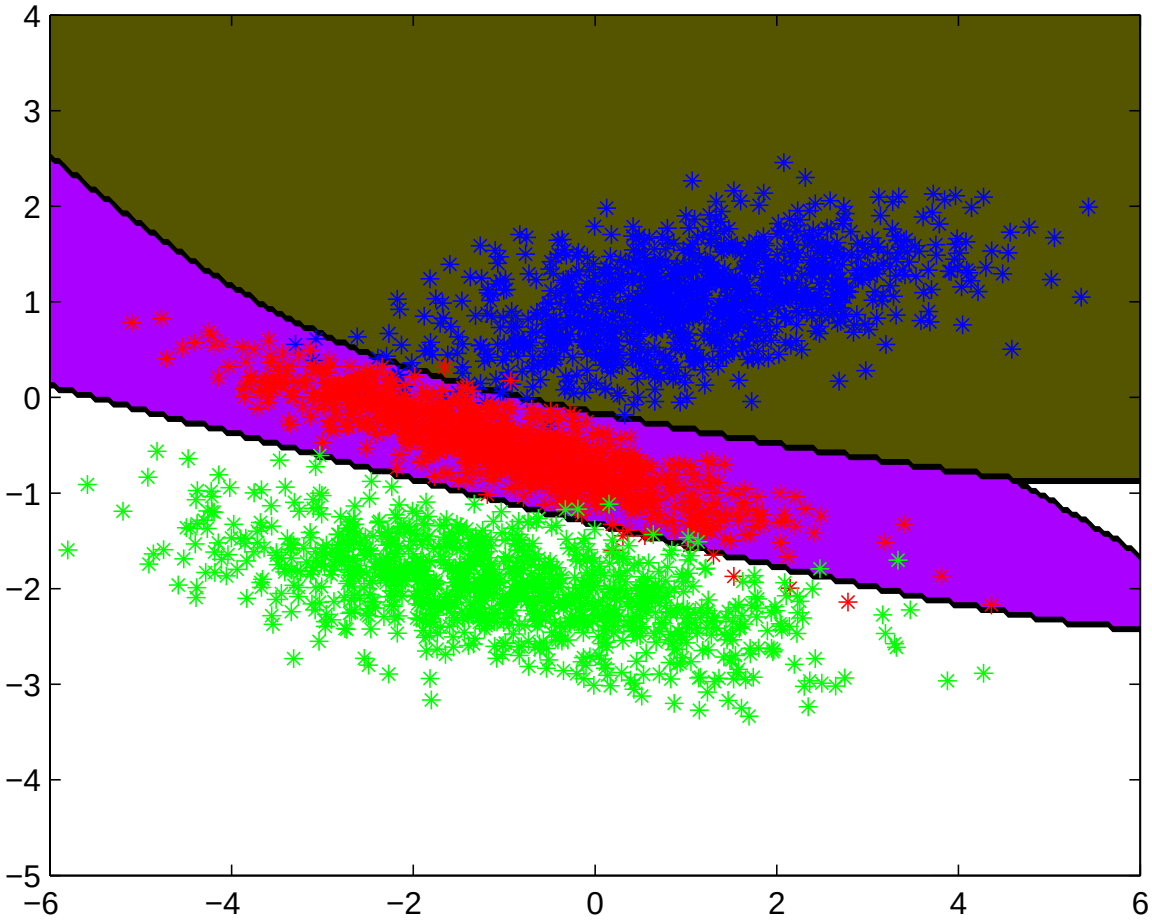
$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log P_\lambda(\mathcal{M}(W_u) \mid \mathbf{X}_u)$$

$$F_{\text{MLE}} = \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W_u))P(W_u)}{\sum_{w'} P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W'))P(W')}$$

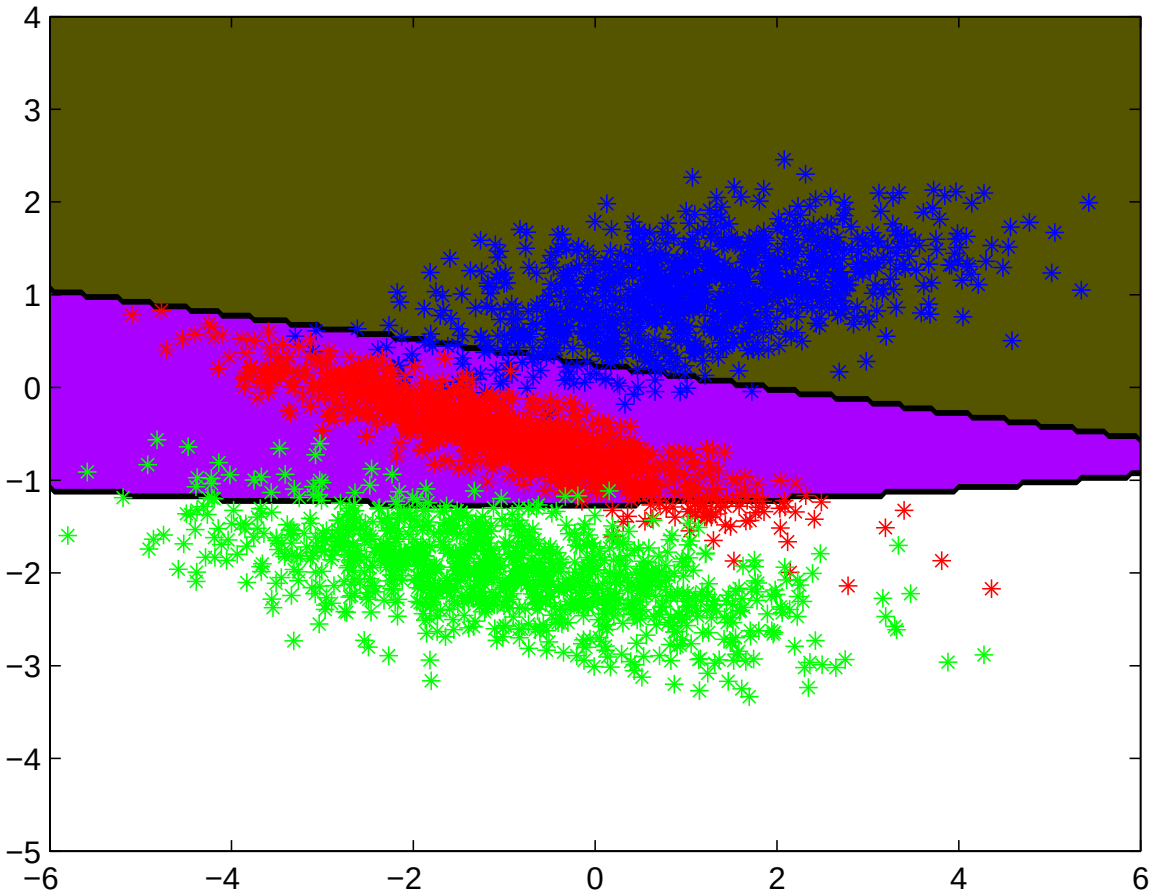
Example



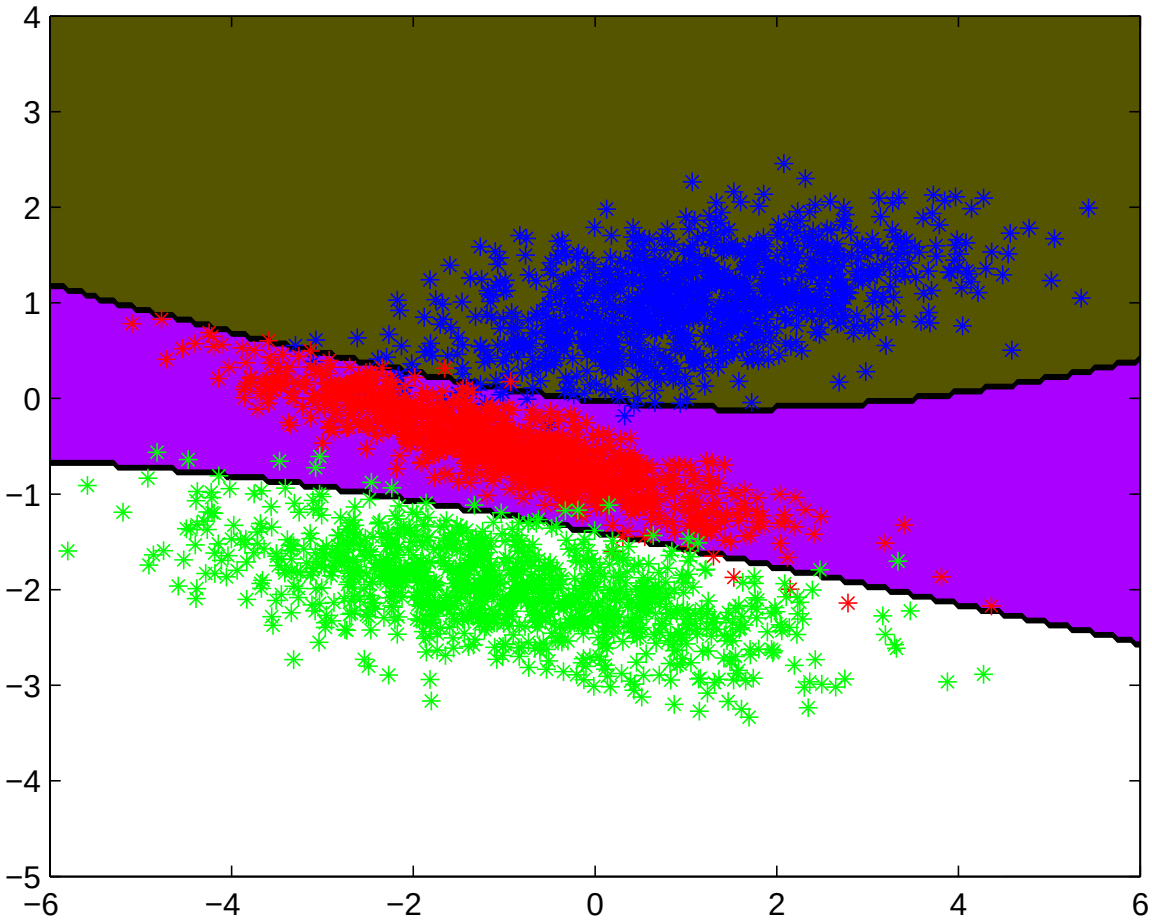
Full covariance Gaussian with MLE



Diagonal covariance Gaussian with MLE



Diagonal covariance Gaussian with MMIE



Maximum mutual information estimation

$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid M(W_u))P(W_u)}{\sum_{W'} P_\lambda(\mathbf{X}_u \mid M(W'))P(W')}$$

Maximum mutual information estimation

$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid M(W_u))P(W_u)}{\sum_{W'} P_\lambda(\mathbf{X}_u \mid M(W'))P(W')}$$

- **Numerator**: likelihood of data given correct word sequence ("clamped" to reference alignment)

Maximum mutual information estimation

$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid M(W_u))P(W_u)}{\sum_{W'} P_\lambda(\mathbf{X}_u \mid M(W'))P(W')}$$

- **Numerator**: likelihood of data given correct word sequence ("clamped" to reference alignment)
- **Denominator**: total likelihood of the data given all possible word sequences – equivalent to summing over all possible word sequences estimated by the full acoustic and language models in recognition. ("free")

Maximum mutual information estimation

$$F_{\mathrm{MMIE}} = \sum_{u=1}^{U} \log \frac{P_\lambda(\mathbf{X}_u \mid M(W_u))P(W_u)}{\sum_{W'} P_\lambda(\mathbf{X}_u \mid M(W'))P(W')}$$

- **Numerator**: likelihood of data given correct word sequence ("clamped" to reference alignment)
- **Denominator**: total likelihood of the data given all possible word sequences – equivalent to summing over all possible word sequences estimated by the full acoustic and language models in recognition. ("free")
- The objective function F_{MMIE} is optimised by making the correct word sequence likely (maximise the numerator), and all other word sequences unlikely (minimise the denominator)

Extended Baum-Welch (EBW)

- No EM-based optimization approach for F_{MMIE}
- Gradient-based approaches are straightforward but slow
- Approximation: Extended Baum-Welch (EBW) algorithm provides update formulae similar to forward-backward recursions used in MLE.
- Extended Baum-Welch – Updating the mean:

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\sum_{u=1}^{U} \left[\Theta_{jm}^{u}(\mathcal{M}_{\text{num}}) - \Theta_{jm}^{u}(\mathcal{M}_{\text{den}}) \right] + D\boldsymbol{\mu}_{jm}}{\sum_{u=1}^{U} \left[\Gamma_{jm}^{u}(\mathcal{M}_{\text{num}}) - \Gamma_{jm}^{u}(\mathcal{M}_{\text{den}}) \right] + D}$$

Extended Baum-Welch (EBW)

- No EM-based optimization approach for F_{MMIE}
- Gradient-based approaches are straightforward but slow
- Approximation: Extended Baum-Welch (EBW) algorithm provides update formulae similar to forward-backward recursions used in MLE.
- Extended Baum-Welch – Updating the mean:

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\sum_{u=1}^{U} \left[\Theta_{jm}^{u}(\mathcal{M}_{\text{num}}) - \Theta_{jm}^{u}(\mathcal{M}_{\text{den}}) \right] + D\boldsymbol{\mu}_{jm}}{\sum_{u=1}^{U} \left[\Gamma_{jm}^{u}(\mathcal{M}_{\text{num}}) - \Gamma_{jm}^{u}(\mathcal{M}_{\text{den}}) \right] + D}$$

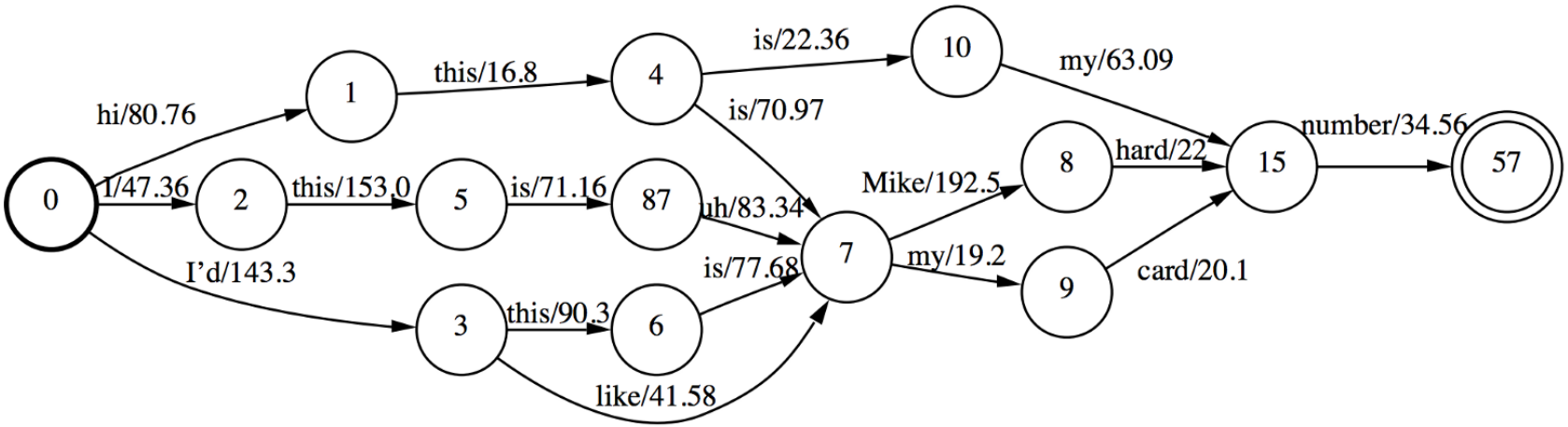
- Can interpret D as a weight between old and new estimates; in practice D estimated for each Gaussian to ensure variance updates are positive

Lattice-based sequence training

- Computing the denominator involves summing over all possible word sequences – this is hard!
- Estimate by generating word lattices, and summing over all words in the lattice
- Generate numerator and denominator lattices for every training utterance
- Denominator lattice uses recognition setup (with a weaker language model)
- Each word in the lattice is decoded to give a phone segmentation, and forward-backward is then used to compute the state occupation probabilities
- Lattices not usually re-computed during training

Forward-backward over lattices

- After decoding, states in the lattice have known start and end times
- Compute a log likelihood over each arc
- Use the forward-backward algorithm over the arcs in the lattice
- Within each arc, compute the state occupancy probabilities using forward-backward with a linear HMM



Lattice from www.cs.nyu.edu/~mohri/asr12/lecture_12.pdf

MMIE is sequence discriminative training

- **Sequence:** like forward-backward (MLE) training, the overall objective function is at the sequence level – maximise the posterior probability of the word sequence given the acoustics $P_\lambda(\mathcal{M}(W_u) \mid \mathbf{X}_u)$
- **Discriminative:** unlike forward-backward (MLE) training the overall objective function for MMIE is discriminative – to maximise MMI:
  - Maximise the numerator by increasing the likelihood of data given the correct word sequence
  - Minimise the denominator by decreasing the total likelihood of the data given all possible word sequences

  This results in "pushing up" the correct word sequence, while "pulling down" the rest

MPE: Minimum phone error

- Basic idea adjust the optimization criterion so it is directly related to word error rate
- Minimum phone error (MPE) criterion

MPE: Minimum phone error

- Basic idea adjust the optimization criterion so it is directly related to word error rate
- Minimum phone error (MPE) criterion

$$F_{\text{MPE}} = \sum_{u=1}^{U} \log \frac{\sum_W P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W))P(W)A(W, W_u)}{\sum_{W'} P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W'))P(W')}$$

- $A(W, W_u)$ is the phone transcription accuracy of the sentence W given the reference W_u

MPE: Minimum phone error

- Basic idea adjust the optimization criterion so it is directly related to word error rate
- Minimum phone error (MPE) criterion

$$F_{\text{MMIE}} = \sum_{u=1}^{U} \log \frac{\sum_W P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W_u)) P(W_u) A(W, W_u)}{\sum_{W'} P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W')) P(W')}$$

- $A(W, W_u)$ is the phone transcription accuracy of the sentence W given the reference W_u

MPE: Minimum phone error

- Basic idea adjust the optimization criterion so it is directly related to word error rate
- Minimum phone error (MPE) criterion

$$F_{\text{MPE}} = \sum_{u=1}^{U} \log \frac{\sum_W P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W))P(W)A(W, W_u)}{\sum_{W'} P_\lambda(\mathbf{X}_u \mid \mathcal{M}(W'))P(W')}$$

- $A(W, W_u)$ is the phone transcription accuracy of the sentence W given the reference W_u
- F_{MPE} is a weighted average over all possible sentences w of the raw phone accuracy
- Although MPE optimizes a phone accuracy level, it does so in the context of a word-level system: it is optimized by finding probable sentences with low phone error rates

HMM/DNN systems

- DNN-based systems are discriminative – the cross-entropy (CE) training criterion with softmax output layer "pushes up" the correct label, and "pulls down" competing labels
- CE is a frame-based criterion – we would like a sequence level training criterion for DNNs, operating at the word sequence level
- Can we train DNN systems with an MMI-type objective function?

HMM/DNN systems

- DNN-based systems are discriminative – the cross-entropy (CE) training criterion with softmax output layer "pushes up" the correct label, and "pulls down" competing labels
- CE is a frame-based criterion – we would like a sequence level training criterion for DNNs, operating at the word sequence level
- Can we train DNN systems with an MMI-type objective function? – **Yes**

Sequence training of hybrid HMM/DNN systems

- Forward- and back-propagation equations are structurally similar to forward and backward recursions in HMM training
- Initially train DNN framewise using cross-entropy (CE) error function
  - Use CE-trained model to generate alignments and lattices for sequence training
  - Use CE-trained weights to initialise weights for sequence training
- Train using back-propagation with sequence training objective function (e.g. MMI)

Sequence training of hybrid HMM/DNN systems

$$\frac{\partial F_{\text{MMIE}}}{\partial \log p(x_t|q_t = j)} = \gamma_j^{\text{num}}(t) - \gamma_j^{\text{den}}(t)$$

$$\begin{aligned}\frac{\partial F_{\text{MMIE}}}{\partial \log w_{ik}} &= \sum_j \frac{\partial F_{\text{MMIE}}}{\partial \log p(x_t|q_t = j)} \frac{\partial \log p(x_t|q_t = j)}{\partial w_{ik}} \\ &= \sum_j (\gamma_j^{\text{num}}(t) - \gamma_j^{\text{den}}(t)) \frac{\partial \log p(x_t|q_t = j)}{\partial w_{ik}}\end{aligned}$$

Sequence training results on Switchboard (Kaldi)

Results on Switchboard "Hub 5 '00" test set, trained on 300h training set, comparing maximum likelihood (ML) and discriminative (BMMI) trained GMMs with framewise cross-entropy (CE) and sequence trained (MMI) DNNs. GMM systems use speaker adaptive training (SAT).
All systems had 8859 tied triphone states.
GMMs – 200k Gaussians
DNNs – 6 hidden layers each with 2048 hidden units

	SWB	CHE	Total
GMM ML (+SAT)	21.2	36.4	28.8
GMM BMMI (+SAT)	18.6	33.0	25.8
DNN CE	14.2	25.7	20.0
DNN MMI	12.9	24.6	18.8

Veseley et al, 2013.

Lattice-Free MMI (LF-MMI

- Lattice-based sequence training of requires initially training an initial model to give a (very good) weight initialisation and to generate lattices for the denominator computation
- Lattice-free MMI (Povey et al, 2016) (sometimes called the 'Chain' model)
  - Avoids the need to pre-compute lattices for the denominator
  - Avoids the requirement to train using frame-based CE loss function, before sequence training

The core method

- Both numerator and denominator state sequences are represented as *HCLG* FSTs
- Parallelise denominator forward-backward computation on a GPU
- Replace word-level LM with a 4-gram phone LM for efficiency
- Reduce the frame rate to 30ms
- Use a simpler HMM topology motivated by CTC (see Lecture 15)

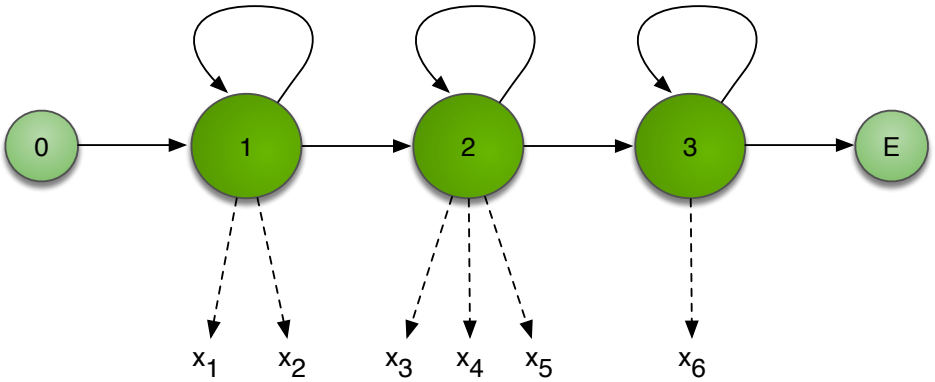
Extra tricks

- Train on small fixed-size chunks (1.5s)
  - enough to overcome the incorrectness of the conditional independence assumption
- Careful optimisation of denominator FST to minimise the size
- Various types of regularisation

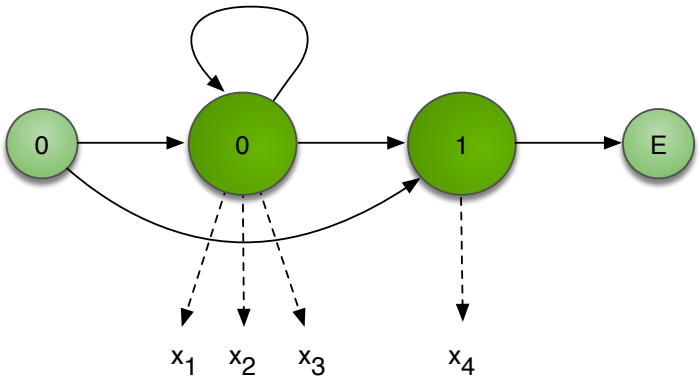
HMM topologies

Replace standard 3-state HMM with topology that can be traversed in a single frame

Standard topology



LF-MMI topology



Denominator FST

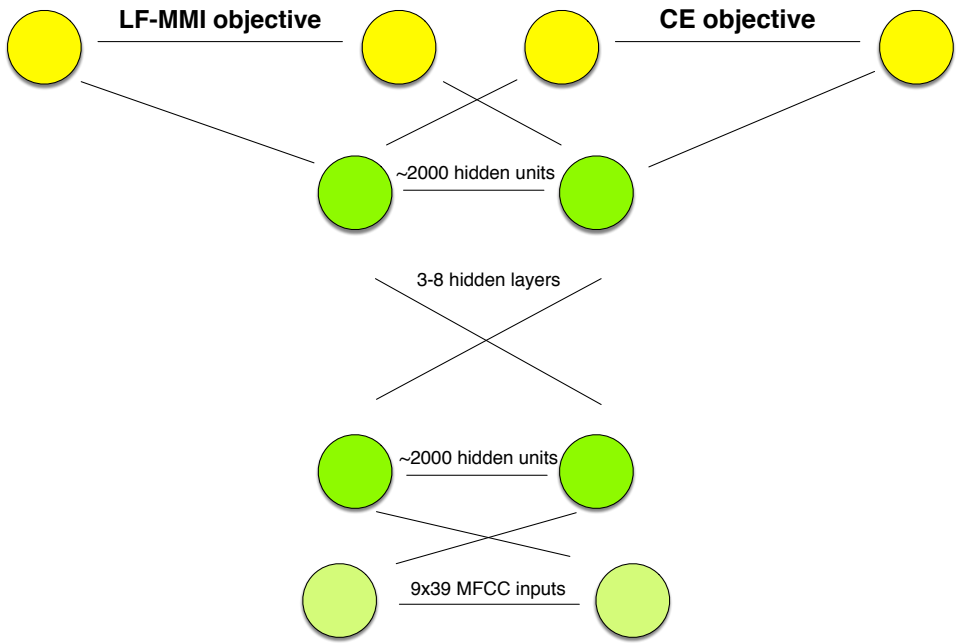
- LM is essentially a 3-gram phone LM
- No pruning and no backoff to minimise the size
  - Use of unpruned 3-grams means that there is always a 2-word history.
  - Minimises the size of the recognition graph when phonetic context is incorporated
- Addition of a fixed number of the most common 4-grams
- Conversion to *HCLG* FST in the normal way
- *HCLG* size reduced by a series of FST reversal, weight pushing and minimisation operations, followed by epsilon removal

Specialised forward-backward algorithm

- Work with probabilities rather than log-probabilities to avoid expensive log/exp operations
- Numeric overflow and underflow is a big problem
- Two specialisations:
  - re-normalise probabilities at every time step
  - the "leaky HMM" - gradual forgetting of context by assigning a small probability to transitions between any pair of states

Regularisation

- Use standard Cross-Entropy objective as a secondary task
  - all but the final hidden layer shared between tasks
  - use numerator posteriors for convenience



- L2 regularisation on the main output to prevent over-confident likelihood estimations
- Leaky HMM (mentioned earlier)

Benefits of LF-MMI

- Models are typically faster during training and decoding than standard models
- Word error rates are generally lower
- Ability to properly compute state posterior probabilities over arbitrary state sequences also opens possibilities for
  - Semi-supervised training
  - Cross-model student-teacher training

  where sequence information is critical

Benefits of LF-MMI

- Models are typically faster during training and decoding than standard models
- Word error rates are generally lower
- Ability to properly compute state posterior probabilities over arbitrary state sequences also opens possibilities for
  - Semi-supervised training
  - Cross-model student-teacher training

  where sequence information is critical

But – difficulties when training transcripts are unreliable

LF-MMI results on Switchboard

Results on SWB portion of the Hub 5 2000 test set, trained on 300h training set. Results use speed perturbation and i-vector based speaker adaptation.

Objective	Model (size)	WER (%)
CE	TDNN-A (16.6M)	12.5
CE $\rightarrow$ sMBR	TDNN-A (16.6M)	11.4
LF-MMI	TDNN-A (9.8M)	10.7
	TDNN-B (9.9M)	10.4
	TDNN-C (11.2M)	10.2
LF-MMI $\rightarrow$ sMBR	TDNN-C (11.2M)	10.0

See Povey et al (2016) for more results

Summary

- Sequence training: discriminatively optimise GMM or DNN to a sentence (sequence) level criterion rather than a frame level criterion
  - ML training of HMM/GMM – sequence-level, not discriminative
  - CE training of HMM/NN – discriminative at the frame level
  - MMI training of HMM/GMM or HMM/NN – discriminative at the sequence level
- Usually initialise sequence discriminative training
  - HMM/GMM – first train using ML, followed by MMI
  - HMM/NN – first train at frame level (CE), followed by MMI
- Sequence discriminative training is computationally costly – need to compute the "denominator lattices"
- Lattice-free MMI for HMM/DNN systems:
  - avoids the need to compute denominator lattices
  - avoids the need to first apply CE training

Reading

- HMM discriminative training: Sec 27.3.1 of: S Young (2008), "HMMs and Related Speech Recognition Technologies", in *Springer Handbook of Speech Processing*, Benesty, Sondhi and Huang (eds), chapter 27, 539–557. http://www.inf.ed.ac.uk/teaching/courses/asr/2010-11/restrict/Young.pdf
- NN sequence training: K Vesely et al (2013), "Sequence-discriminative training of deep neural networks", Interspeech-2013, http://www.fit.vutbr.cz/research/groups/speech/publi/2013/vesely_interspeech2013_IS131333.pdf
- Lattice-free MMI: D Povey et al (2016), "Purely sequence-trained neural networks for ASR based on lattice-free MMI", Interspeech-2016. http://www.danielpovey.com/files/2016_interspeech_mmi.pdf; slides – http://www.danielpovey.com/files/2016_interspeech_mmi_presentation.pptx (covered in lecture 12)