Hidden Markov Models, Discriminative Training, and Modern Speech Recognition

Li Deng
Microsoft Research
UW-EE516; Lecture May 21, 2009
Architecture of a speech recognition system

Voice -> Signal Processing

Application -> Decoder

Adapteration

Acoustic Models

Language Models
Fundamental Equation of Speech Recognition

\[ \hat{W} = \arg \max_W P(W \mid x) = \arg \max_W P(x \mid W)P(W) \]

- Roles of acoustic modeling and Hidden Markov Model (HMM)
Outline

Introduction to HMMs; Basic ML Training Techniques

HMM Parameter Learning
- EM (maximum likelihood)
- Discriminative learning (intro)

Engineering Intricacy in Using HMMs in Speech Recognition
- Left-to-right structure
- Use of lexicon and pronunciation modeling
- Use of context dependency (variance reduction)
- Complexity control: decision tree & automatic parameter tying

“Beyond HMM” for Speech Modeling/Recognition
- Better generative models for speech dynamics
- Discriminative models: e.g., CRF and feature engineering

 Discriminative Training in Speech Recognition --- A unifying framework
HMM Introduction:

- The 1-coin model is observable because the output sequence can be mapped to a specific sequence of state transitions.

- The remaining models are hidden because the underlying state sequence cannot be directly inferred from the output sequence.
Specification of an HMM

A - the state transition probability matrix
\[ a_{ij} = P(q_{t+1} = j|q_t = i) \]

B - observation probability distribution
\[ b_j(k) = P(o_t = k|q_t = j) \quad i \leq k \leq M \]

\( \pi \) - the initial state distribution
Central problems for HMM

Problem 1: Evaluation

- Probability of occurrence of a particular observation sequence, \( O = \{o_1, \ldots, o_k\} \), given the model: \( P(O|\lambda) \)

- Not straightforward – hidden states

- Useful in sequence classification (e.g., isolated-word recognition):
Problem 1: Evaluation

Naïve computation:

\[ P(O \mid \lambda) = \sum_{q} P(O \mid q, \lambda) P(q \mid \lambda) \]

where

\[ P(q \mid \lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T} \]

- The above sum is over all state paths
- There are \( N^T \) states paths, each ‘costing’ \( O(T) \) calculations, leading to \( O(TN^T) \) time complexity.
Problem 1

Need efficient Solution

Define auxiliary forward variable $\alpha$:

$$\alpha_t(i) = P(o_1, \ldots, o_t \mid q_t = i, \lambda)$$

$\alpha_t(i)$ is the probability of observing a partial sequence of observables $o_1, \ldots, o_t$ such that at time $t$, state $q_t = i$

Then recursion on $\alpha$ provides the most efficient solution to problem 1 (forward algorithm)
Central problems for HMM

**Problem 2: Decoding**

- Optimal state sequence to produce given observations, \( O = \{o_1, \ldots, o_k\} \), given model
- Optimality criterion
- Useful in recognition problems (e.g., continuous speech recognition)
Problem 2: Decoding

Recursion:

\[ \delta_t(j) = \max_{1 \leq i \leq N} (\delta_{t-1}(i) a_{ij}) b_j(o_t) \]

\[ \psi_t(j) = \arg \max_{1 \leq i \leq N} (\delta_{t-1}(i) a_{ij}) \]

Termination:

\[ P^* = \max_{1 \leq i \leq N} \delta_T(i) \]

\[ q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i) \]

Backtracking:

\[ q_t^* = \psi_t(q_{t+1}^*) \]

2 \leq t \leq T, 1 \leq j \leq N

\[ P^* \text{ gives the state-optimised probability} \]

\[ Q^* \text{ is the optimal state sequence} \]

\[ (Q^* = \{q_1^*, q_2^*, \ldots, q_T^*\}) \]
Central problems for HMM

Problem 3: Learning of HMM parameters:
- Determine optimum model, given a training set of observations
- Find $\lambda$, such that $P(O|\lambda)$ is maximal (ML), or class discrimination is maximal (with greater separation margins)
Outline

Introduction to HMMs; Basic ML Techniques

HMM Parameter Learning
- EM (maximum likelihood)
- Discriminative learning

Engineering Intricacy in Using HMMs (Speech)
- Left-to-right structure
- Use of lexicon and pronunciation modeling
- Use of context dependency (variance reduction)
- Complexity control: decision tree & automatic parameter tying

"Beyond HMM" for Speech Modeling/Recognition
- Better generative models for speech dynamics
- Discriminative models: e.g., CRF and feature engineering

Discriminative Training in Speech Recognition --- A unifying framework
Problem 3: Learning

Maximum Likelihood: Baum-Welch algorithm uses the forward and backward algorithms to calculate the auxiliary variables $\alpha, \beta$

B-W algorithm is a special case of the EM algorithm:

- **E-step**: calculation of posterior probs from $\alpha, \beta$

**M-step**: HMM parameter updates

Practical issues:
- Can get stuck in local maxima
- Numerical problems – log and scaling
Discriminative Learning (intro)

Maximum Mutual Information (MMI)
Minimum Classification Error (MCE)
Minimum Phone/Word Errors (MPE, MWE)
Large-Margin Discriminative Training
More Coverage later
Outline

Introduction to HMMs; Basic ML Techniques
HMM Parameter Learning (Advanced ML Techniques)
  - EM (maximum likelihood)
  - Discriminative learning

Engineering Intricacy in Using HMMs (Speech)
  - Left-to-right structure
  - Use of lexicon and pronunciation modeling
  - Use of context dependency (variance reduction)
  - Complexity control: decision tree & automatic parameter tying

“Beyond HMM” for Speech Modeling/Recognition
  - Better generative models for speech dynamics
  - Discriminative models: e.g., CRF and feature engineering

Discriminative Training in Speech Recognition --- A unifying framework
Acoustic models encode the temporal evolution of the features (spectrum).

Gaussian mixture distributions are used to account for variations in speaker, accent, and pronunciation.

Phonetic model topologies are simple left-to-right structures.

Skip states (time-warping) and multiple paths (alternate pronunciations) are also common features of models.

Sharing model parameters is a common strategy to reduce complexity.
- Phonetic units are preferred.
- Training does not require phonetic transcriptions.
- Many types of phonetic units.
- Cross-word units add complexity.

Monophone Modeling
Word internal triphone modeling
Crossword triphone modeling
Outline

Introduction to HMMs; Basic ML Techniques

HMM Parameter Learning (Advanced ML Techniques)
- EM (maximum likelihood)
- Discriminative learning
- Large-margin discriminative learning

Engineering Intricacy in Using HMMs (Speech)
- Left-to-right structure
- Use of lexicon and pronunciation modeling
- Use of context dependency (variance reduction)
- Complexity control: decision tree & automatic parameter tying

“Beyond HMM” for Speech Modeling/Recognition
- Better generative models for speech dynamics
- Discriminative models: e.g., CRF and feature engineering
HMM is a Poor Model for Speech (not just IID)

- “Did you get” is reduced such that the resulting word is pronounced “jyuge.”
- Phoneme deletion rate: ~12%
- Syllable deletion rate: ~1%
- Predicting pronunciations of words is crucial!

“have sort of like a a a manpower”

- Conversational speech defies conventional grammatical structure.
- Constrained interfaces have failed!
Direction 1: “Beyond HMM” Generative Modeling

- build more accurate statistical speech models than HMM
- Incorporate key insights from human speech processing
Hidden Dynamic Model --- Graphical Model Representation
<table>
<thead>
<tr>
<th>ASR Systems</th>
<th>Generative</th>
<th>Discriminative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early (Discrete) HMM (Lee &amp; Hon)</td>
<td>66.08</td>
<td>?</td>
</tr>
<tr>
<td>Early HMM (Cambridge U)</td>
<td>66.07</td>
<td>67.50 (MMI)</td>
</tr>
<tr>
<td>Early HMM (ATR-Japan)</td>
<td>66.80</td>
<td>68.70 (MCE)</td>
</tr>
<tr>
<td>Conditional Random Field (Ohio State U., 2005)</td>
<td>NA</td>
<td>67.87</td>
</tr>
<tr>
<td>Recurrent Neural Nets (Cambridge U.)</td>
<td>NA</td>
<td>73.90</td>
</tr>
<tr>
<td>Large-Margin HMM (U. Penn, 2006)</td>
<td>67.30</td>
<td>71.80 (NIPS 06)</td>
</tr>
<tr>
<td>Modern HMM (MSR)</td>
<td>72.54</td>
<td>72.85 (MCE)</td>
</tr>
<tr>
<td>Hidden Dynamic Model (MSR)</td>
<td>75.18</td>
<td>?</td>
</tr>
</tbody>
</table>
Major Research Directions in the Field

Direction 2: Discriminative Modeling/Learning

- Acknowledge accurate speech modeling is too hard
- View HMM not as a generative model but as discriminant function
- Large-margin discriminative parameter learning
- “Direct” models for speech-class discrimination: e.g., MEMM, CRF; feature engineering; deep learning and structure
- Discriminative training for generative models
Outline

Introduction to HMMs; Basic ML Training Techniques

HMM Parameter Learning
- EM (maximum likelihood)
- Discriminative learning (intro)

Engineering Intricacy in Using HMMs in Speech Recognition
- Left-to-right structure
- Use of lexicon and pronunciation modeling
- Use of context dependency (variance reduction)
- Complexity control: decision tree & automatic parameter tying

“Beyond HMM” for Speech Modeling/Recognition
- Better generative models for speech dynamics
- Discriminative models: e.g., CRF and feature engineering

Discriminative Training in Speech Recognition --- A unifying framework (He, Deng, Wu, 2008, IEEE SPM)
Discriminative Learning in Sequential Pattern Recognition

Discriminative learning has become a major theme in recent statistical signal processing and pattern recognition research including practically all applications of speech recognition. A key to understanding the speech process is the dynamic characterization of the speech signal.
MMIE Criterion

Maximizing the mutual information is equivalent to choosing the parameter set $\lambda$ to maximize:

$$F_{\text{MMIE}}(\lambda) = \sum_{t=1}^{R} \log \left( \frac{P_\lambda(O_t | M_{w_t})P(w_t)}{\sum_{\tilde{w}} P_\lambda(O_t | M_{\tilde{w}})P(\tilde{w})} \right)$$

Maximization implies increasing the numerator term (maximum likelihood estimation – MLE) and/or decreasing the denominator term.

The latter is accomplished by reducing the probabilities of incorrect, or competing, hypotheses.

Clever optimization methods.
MPE/MWE Criteria

\[ O_{MWE}(\Lambda) = \sum_{r=1}^{R} \frac{\sum_{s_r} p(X_r|s_r,\Lambda) P(s_r) A_1(s_r,S_r)}{\sum_{s_r} p(X_r|s_r,\Lambda) P(s_r)} \]

- Raw phone or word accuracy \( A(s, S) \) (Povey et. al. 2002)
- Requiring use of lattices to determine \( A(s, S) \)
- Very difficult to optimize wrt HMM parameters
- Engineering tricks
- Unifying MPE, MCE, MMI (He & Deng, 2007) in both criterion and optimization method
MCE Criterion

Discriminative functions:
\[ g_i(X, \Lambda), \quad i = 1, M \]

Misclassification measure:
\[ d_i(X, \Lambda) = -g_i(X, \Lambda) + \arg \max_{j \neq i, j=1,M} g_j(X, \Lambda) \]

Smoothed classification error count:
\[ d(X, \Lambda) = \sum_{i=1}^{M} d_i(X, \Lambda) \cdot 1(X \in C_i) \]

Optimization (gradient):
\[ L(\Lambda) = \mathbb{E}_{X}[l(d(X, \Lambda))] = \int_X l(d(X, \Lambda)) \cdot P(X) \, \text{d}X \]
\[ \Lambda_{opt} = \arg \min_{\Lambda} L(\Lambda) \]

where error smoothing function is sigmoidal:
\[ l(d_i(X, \Lambda)) = \frac{1}{1 + \exp(-\gamma \cdot (d_i(X, \Lambda) + \theta))}, \quad \gamma > 0 \]
Large-Margin Estimates

MMI, MPE/MWE, & common MCE do not embrace margins

“Margin” parameter $m > 0$ in MCE’s error smoothing function

$$l(d_i(X, \Lambda)) = \frac{1}{1 + \exp(-\gamma \cdot (d_i(X, \Lambda) + m(I))))}, \quad \gamma > 0$$

Careful exploitation of the “margin” parameter in LM-MCE training of HMM parameters leads to significant ASR performance improvement (Yu, Deng, He, Acero, 2006, 2007)
Large-Margin Estimates

Paper: Sha & Saul (NIPS 06): “Large Margin Training of Continuous-density Hidden Markov Models”

Re-formulate Gaussians in HMM -- turning log(Variances) into independent parameters to optimize

SVM-type formulation of objective function --- use of slack variables

Make “margin” a function of the training token according to Hamming distance

Convex optimization (vs. LM-MCE with local optima)

Good relative performance improvement on TIMIT: phone accuracy from 67.3% (EM) to 71.8% (LM)

Several other groups explore different ways of incorporating margins in HMM parameter training
MCE in more detail

Define misclassification measure:

\[ d_r(X_r, \Lambda) = \log p_{\Lambda}(X_r, s_{r,1}) - \log p_{\Lambda}(X_r, S_r) \]

the case of using correct and top one incorrect competing tokens)

Observation. seq.: \( X_r \quad x_1, x_2, x_3, x_4, \ldots, x_t, \ldots, x_T \)

**correct label:**
\( S_r \quad \text{OH THREE EIGHT} \)

**competitor:**
\( s_{r,1} \quad \text{OH SIX EIGHT} \)

\( s_{r,1} \): the top one incorrect (not equal to \( S_r \)) competing string
MCE: Loss function

Classification: 
\[ s_r^* = \arg \max_{s_r} \{ \log p_{\Lambda} (X_r, s_r) \} \]

Classifi. error: 
\[ d_r(X_r, \Lambda) > 0 \rightarrow 1 \text{ classification error} \]
\[ d_r(X_r, \Lambda) < 0 \rightarrow 0 \text{ classification error} \]

Loss function: smoothed error count function:

\[ l_r (d_r (X_r, \Lambda)) = \frac{1}{1 + e^{-d_r (X_r, \Lambda)}} \]
MCE: Objective function

MCE objective function:

\[ L_{MCE}(\Lambda) = \frac{1}{R} \sum_{r=1}^{R} l_r \left( d_r(X_r, \Lambda) \right) \]

\( L_{MCE}(\Lambda) \) is the smoothed recognition error rate on the string (token) level.

Model (acoustic model) is trained to minimize \( L_{MCE}(\Lambda) \), i.e., \( \Lambda^* = \arg\min \Lambda \{ L_{MCE}(\Lambda) \} \)
### MCE: Optimization

<table>
<thead>
<tr>
<th>Traditional Stochastic GD</th>
<th>New Growth Transform.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient descent based online optimization</td>
<td>Extend Baum-Welch based batch-mode method</td>
</tr>
<tr>
<td>Convergence is unstable</td>
<td>Stable convergence</td>
</tr>
<tr>
<td>Training process is difficult to be parallelized</td>
<td>Ready for parallelized processing</td>
</tr>
</tbody>
</table>
Introduction to EBW or GT (Part I)

Applied to rational function as the objective fctn:

\[ P(\Lambda) = \frac{G(\Lambda)}{H(\Lambda)} \]

Equivalent auxiliary objective fctn (often easier to optimize):

\[ F(\Lambda; \Lambda') = G(\Lambda) - P(\Lambda')H(\Lambda) + D \]

where \( D \) is a \( \Lambda \)-independent constant. (see proof in SPM paper)
Introduction to EBW or GT (Part II)

Applied to the objective function of the form:

\[ F(\Lambda; \Lambda') = \sum_s \sum_q \int f(\chi, q, s, \Lambda) d\chi. \]

Equivalent auxiliary objective function (easier to optimize):

\[ V(\Lambda; \Lambda') = \sum_s \sum_q \int f(\chi, q, s, \Lambda') \log f(\chi, q, s, \Lambda) d\chi, \]

(see proof in SPM paper & tech report)
MCE: Optimization

- Growth Transformation based MCE:
  If \( \Lambda = T(\Lambda') \) ensures \( P(\Lambda) > P(\Lambda') \), i.e., \( P(\Lambda) \) grows, then \( T(\cdot) \) is called a *growth transformation* of \( \Lambda \) for \( P(\Lambda) \).

Minimizing
\[
L_{MCE}(\Lambda) = \sum l(d(\cdot))
\]

Maximizing
\[
P(\Lambda) = \frac{G(\Lambda)}{H(\Lambda)}
\]

Maximizing
\[
F(\Lambda;\Lambda') = G-P'\times H+D
\]

GT formula
\[
\frac{\partial U(\cdot)}{\partial \Lambda} = 0 \Rightarrow \Lambda = T(\Lambda')
\]

Maximizing
\[
U(\Lambda;\Lambda') = \sum f'(\cdot) \log f(\cdot)
\]

Maximizing
\[
F(\Lambda;\Lambda') = \sum f(\cdot)
\]
“Rational Function” Form for MEC Objective Function

\[ d_r (X_r, \Lambda) = \log p_\Lambda (X_r, s_{r,1}) - \log p_\Lambda (X_r, S_r) \]

\[ l_r (d_r (X_r, \Lambda)) = \frac{1}{1 + e^{-d_r (X_r, \Lambda)}} \]

\[ L_{MCE} (\Lambda) = \frac{1}{R} \sum_{r=1}^{R} l_r (d_r (X_r, \Lambda)) \]

Not an obvious rational function (ratio) due to the summation
Rational Function” Form for MEC Objective Function

Re-write MCE loss function to

\[ l_r(d_r(X_r, \Lambda)) = \frac{p(X_r, s_{r,1} | \Lambda)}{p(X_r, s_{r,1} | \Lambda) + p(X_r, S_r | \Lambda)} \]

Then, min. \( L_{MCE}(\Lambda) \Leftrightarrow \max. \ Q(\Lambda) \), where

\[ Q(\Lambda) = R \left( 1 - L_{MCE}(\Lambda) \right) \]

\[ = \sum_{r=1}^{R} \frac{p(X_r, S_r | \Lambda)}{p(X_r, s_{r,1} | \Lambda) + p(X_r, S_r | \Lambda)} = \sum_{r=1}^{R} \frac{\sum_{s_r \in \{s_{r,1}, S_r\}} p(X_r, s_r | \Lambda) \delta(s_r, S_r)}{\sum_{s_r \in \{s_{r,1}, S_r\}} p(X_r, s_r | \Lambda)} \]
\[ C_{\text{MCE}}(\Lambda) = \sum_{r=1}^{\infty} \frac{\sum_{s_r} p(X_r, s_r | \Lambda) \delta(s_r, s_r)}{\sum_{s_r} p(X_r, s_r | \Lambda)} \]
\[ := O_1 \]
\[ + \frac{\sum_{s_2} p(X_2, s_2 | \Lambda) \delta(s_2, S_2)}{\sum_{s_2} p(X_2, s_2 | \Lambda)} \]
\[ := O_2 \]
\[ + \frac{\sum_{s_3} p(X_3, s_3 | \Lambda) \delta(s_3, S_3)}{\sum_{s_3} p(X_3, s_3 | \Lambda)} \]
\[ := O_3 \]
\[ + \ldots + \frac{\sum_{s_R} p(X_R, s_R | \Lambda) \delta(s_R, S_R)}{\sum_{s_R} p(X_R, s_R | \Lambda)} \]
\[ := O_R \]
\[ \sum \sum_{s_2} p(X_1, s_1 | \Lambda) p(X_2, s_2 | \Lambda) \]
\[ := \sum \sum_{s_2} p(X_1, s_1 | \Lambda) p(X_2, s_2 | \Lambda) \]
\[ + O_3 + \ldots + O_R \]
\[ \sum_{s_1 s_2} p(X_1, X_2, s_1, s_2 | \Lambda) [C_{\text{MCE}}(s_1 s_2)] \]
\[ := \sum_{s_1 s_2} p(X_1, X_2, s_1, s_2 | \Lambda) \]
\[ + O_3 + \ldots + O_R \]
\[ \sum_{s_1 s_2 s_3} p(X_1, X_2, X_3, s_1, s_2, s_3 | \Lambda) [C_{\text{MCE}}(s_1 s_2 s_3)] \]
\[ := \sum_{s_1 s_2 s_3} p(X_1, X_2, X_3, s_1, s_2, s_3 | \Lambda) \]
\[ + O_4 + \ldots + O_R \]
\[ \sum_{s_1 \ldots s_R} p(X_1 \ldots X_R, s_1 \ldots s_R | \Lambda) C_{\text{MCE}}(s_1 \ldots s_R) \]
\[ := \sum_{s_1 \ldots s_R} p(X_1 \ldots X_R, s_1 \ldots s_R | \Lambda) \]

where \( C_{\text{MCE}}(s_1 \ldots s_R) = \sum_{r=1}^{R} \delta(s_r, S_r) \). \( C_{\text{MCE}}(s_1, \ldots, s_R) \) can be interpreted as the string accuracy count for \( s_1, \ldots, s_R \).
Unified Objective Function

- Can do the same trick for MPE/MWE

- MMI objective function is naturally a rational function (due to logarithm in the MMI definition)

- Unified form:

\[
O(\Lambda) = \frac{\sum_{s_1 \ldots s_R} p(X_1 \ldots X_R, s_1 \ldots s_R | \Lambda) \cdot C_{DT}(s_1 \ldots s_R)}{\sum_{s_1 \ldots s_R} p(X_1 \ldots X_R, s_1 \ldots s_R | \Lambda)},
\]  

(25)
Unified Objective Function

**TABLE 1** \( C_{DT}(s_1 \ldots s_R) \) IN THE UNIFIED RATIONAL-FUNCTION FORM FOR MMI, MCE, AND MPE/MWE OBJECTIVE FUNCTIONS. THE SET OF "COMPETING TOKEN CANDIDATES" DISTINGUISHES N-BEST AND ONE-BEST VERSIONS OF THE MCE. NOTE THAT THE OVERALL \( C_{DT}(s_1 \ldots s_R) \) IS CONSTRUCTED FROM ITS CONSTITUENTS \( C_{DT}(s_r) \)'S IN INDIVIDUAL STRING TOKENS BY EITHER SUMMATION (FOR MCE, MPE/MWE) OR PRODUCT (FOR MMI).

<table>
<thead>
<tr>
<th>OBJECTIVE FUNCTIONS</th>
<th>( C_{DT}(s_r) )</th>
<th>( C_{DT}(s_1 \ldots s_R) )</th>
<th>LABEL SEQUENCE SET USED IN DT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>N-BEST</strong></td>
<td>( \delta(s_r, s_r) )</td>
<td>( \sum_{r=1}^{R} C_{DT}(s_r) )</td>
<td>( {s_r, s_{r,1}, \ldots, s_{r,N}} )</td>
</tr>
<tr>
<td><strong>ONE-BEST</strong></td>
<td>( \delta(s_r, s_r) )</td>
<td>( \sum_{r=1}^{R} C_{DT}(s_r) )</td>
<td>( {s_r, s_{r,1}} )</td>
</tr>
<tr>
<td><strong>MPE</strong></td>
<td>( A(s_r, s_r) )</td>
<td>( \sum_{r=1}^{R} C_{DT}(s_r) )</td>
<td>ALL POSSIBLE LABEL SEQUENCES</td>
</tr>
<tr>
<td><strong>MWE</strong></td>
<td>( A_w(s_r, s_r) )</td>
<td>( \sum_{r=1}^{R} C_{DT}(s_r) )</td>
<td>ALL POSSIBLE LABEL SEQUENCES</td>
</tr>
<tr>
<td><strong>MMI</strong></td>
<td>( \delta(s_r, s_r) )</td>
<td>( \prod_{r=1}^{R} C_{DT}(s_r) )</td>
<td>ALL POSSIBLE LABEL SEQUENCES</td>
</tr>
</tbody>
</table>
Coming back to MCE: its objective function can be reformulated as a single *fractional* function $P(\Lambda)$

$$P(\Lambda) = \frac{G(\Lambda)}{H(\Lambda)}$$

where

$$G(\Lambda) = \sum_{s_1} \ldots \sum_{s_R} \left[ p_\Lambda(X_1, \ldots, X_R, s_1, \ldots, s_R) \sum_{r=1}^{R} \delta(s_r, S_r) \right]$$

$$H(\Lambda) = \sum_{s_1} \ldots \sum_{s_R} p_\Lambda(X_1, \ldots, X_R, s_1, \ldots, s_R)$$
Increasing $P(\Lambda)$ can be achieved by maximizing

$$F(\Lambda; \Lambda') = G(\Lambda) - P(\Lambda')H(\Lambda) + D$$

as long as $D$ is a $\Lambda$-independent constant.

i.e.,

$$P(\Lambda) - P(\Lambda') = \frac{1}{H(\Lambda)}[F(\Lambda; \Lambda') - F(\Lambda'; \Lambda')]$$

($\Lambda'$ is the parameter set obtained from last iteration)

Substitute $G()$ and $H()$ into $F()$,

$$F(\Lambda; \Lambda') = \sum_q \sum_s p(X, q, s | \Lambda)[C(s) - P(\Lambda')] + D$$
MCE: Optimization (cont’d)

Reformulate $F(\Lambda;\Lambda')$ to

$$F(\Lambda;\Lambda') = \sum_s \sum_q \int f(\chi,q,s,\Lambda;\Lambda') d\chi$$

where

$$f(\chi,q,s,\Lambda;\Lambda') = [\Gamma(\Lambda') + d(s)] p(\chi,q | s, \Lambda)$$

$$\Gamma(\Lambda') = \delta(\chi,X) \sum_s p(q,s)[C(s) - P(\Lambda')]$$

$$C(s) = \sum_{r=1}^R \delta(s_r, S_r)$$

$F(\Lambda;\Lambda')$ is ready for EM-style optimization after applying GT theory (Part II)

Note: $\Gamma(\Lambda')$ is a constant, and $\log p(\chi, q | s, \Lambda)$ is easy to decompose.
MCE: Optimization (cont’d)

Increasing $F(\Lambda;\Lambda')$ can be achieved by maximizing

$$
U(\Lambda) = \sum_s \sum_q \int \log f(\chi, q, s, \Lambda'; \Lambda') f(\chi, q, s, \Lambda; \Lambda') d\chi
$$

Use EBM/GT (Part II) for E step.

$\log f(\chi, q, s, \Lambda; \Lambda')$ is decomposable w.r.t $\Lambda$, so M step is easy to compute.

So the growth transformation of $\Lambda$ for CDHMM is:

$$
\frac{\partial U(\Lambda)}{\partial \Lambda} = 0 \quad \Rightarrow \quad \Lambda = T(\Lambda')
$$
MCE: HMM estimation formulas

For Gaussian mixture CDHMM,

\[ p(x | \mu, \Sigma) \propto |\Sigma|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\} \]

GT of mean and covariance of Gaussian \( m \) is

\[
\mu_m = \frac{\sum_r \sum_t \Delta \gamma_{m,r} (t)x_{r,t} + D_m \mu_m'}{\sum_r \sum_t \Delta \gamma_{m,r} (t) + D_m}
\]

\[
\Sigma_m = \frac{\sum_r \sum_t \left[ \Delta \gamma_{m,r} (t)(x_{r,t} - \mu_m)(x_{r,t} - \mu_m)^T \right] + D_m \Sigma_m' + D_m (\mu_m - \mu_m')(\mu_m - \mu_m')^T}{\sum_r \sum_t \Delta \gamma_{m,r} (t) + D_m}
\]

where \( \Delta \gamma_{m,r} (t) = p(S_r | X_r, \Lambda')p(s_{r,1} | X_r, \Lambda') \left( \gamma_{m,r,s_r} (t) - \gamma_{m,r,s_{r,1}} (t) \right) \)
MCE: HMM estimation

Setting of $D_m$

Theoretically,

set $D_m$ so that $f(\chi, q, s, \Lambda; \Lambda') > 0$

Empirically,

$$D_m = E \cdot \sum_{r=1}^{R} p(S_r \mid X_r, \Lambda') \left[ p(S_r \mid X_r, \Lambda') \sum_t \gamma_{m,r,S_r}(t) + p(s_{r,1} \mid X_r, \Lambda') \sum_t \gamma_{m,r,s_{r,1}}(t) \right]$$
MCE: Workflow

1. Training utterances
2. Recognition
3. Last iteration
   Model \( \Lambda' \)
4. Competing strings
5. GT-MCE
6. New model \( \Lambda \)

Flow:
- Training utterances flow to Recognition.
- Recognition flow to Competing strings.
- Competing strings flow to GT-MCE.
- GT-MCE flows to New model \( \Lambda \).
- New model \( \Lambda \) flows to Last iteration Model \( \Lambda' \).

Next iteration:
- Green dashed line indicates the transition to the next iteration.
Experiment: TI-DIGITS

- **Vocabulary**: “1” to “9”, plus “oh” and “zero”
- **Training set**: 8623 utterances / 28329 words
- **Test set**: 8700 utterances / 28583 words
- **33-dimentional spectrum feature**: energy +10 MFCCs, plus $\Delta$ and $\Delta\Delta$ features.
- **Model**: Continuous Density HMMs
- **Total number of Gaussian components**: 3284
Experiment: TI-DIGITS

GT-MCE vs. ML (maximum likelihood) baseline

Obtain the lowest error rate on this task
Reduce recognition Word Error Rate (WER) by 23%
Fast and stable convergence
Summary

- HMM is an underlying model for modern speech recognizers
- Fundamental properties of HMM
- Strengths and weaknesses of HMM for speech modeling and recognition
- Discriminative training for HMM as the “generative” model
- Reasonably good theory and practice for HMM discriminative training