Large Margin HMMs for Speech Recognition

Prof. Hui Jiang
Department of Computer Science and Engineering
York University, Toronto, Ont. M3J 1P3, CANADA
Email: hj@cs.yorku.ca

(This is a joint work with Xinwei Li, Chao-Jun Liu)
Outline

• Background
  – Discriminative Training for ASR
  – Large Margin Classifiers: concept & theory

• Large Margin HMMs for ASR
  – A new estimation criterion for HMM
  – Analysis of Margin in CDHMMs
  – Large Margin Estimation (LME): Constrained minimax optimization

• Optimization Methods
  – Gradient Descent (GD) search
  – Semi-definite Programming (SDP)

• Experiments
  – The ISOLET recognition task
  – The TIDIGITS connected digit string recognition

• Summary
**Automatic Speech Recognition**

- **Statistical Speech Model**

  Word Sequence $W$ \[\xrightarrow{\text{Noisy Channel}} \] Speech Signal $X$

  Speech Signal $X$ \[\xrightarrow{\text{Channel Decoding}} \] Word Sequence $W$

- **Bayesian Decision Rule:**

  $\hat{W} = \arg\max_{W \in \Omega} p(W \mid X) = \arg\max_{W \in \Omega} P(W) \cdot p(X \mid W) = \arg\max_{W \in \Omega} F(X \mid \Lambda_W)$

  **Discriminant Function**

  **Language Model**

  **Acoustic Model**
HMM Estimation Methods

- Maximum Likelihood Estimation (MLE)
  - The Baum-Welch algorithm: the EM algorithm for HMM

- Discriminative Training (DT)
  - Maximum Mutual Information Estimation (MMIE):
    - MPE, MWE, etc.
  - Minimum Classification Error (MCE):

- Discriminative training can improve over the standard ML training.

- Can we do better than DT?
Large-Margin Classifier:
Support Vector Machine (SVM)
Large-Margin Classifiers

- Why large margin classifiers yield better generalization performance?
  - Conceptually, large margin
    - Robustness \( w.r.t. \) data patterns
    - Robustness \( w.r.t. \) classifier parameters
  - Theoretically ...
In pattern classification, the generalization upper bound holds with probability $1-\delta$ (Vapnik et. al.):

$$R(\theta) \leq R_{\text{emp}}(\theta) + \sqrt{\frac{1}{N} \left( V \left( \log \frac{2N}{V} + 1 \right) - \log \left( \frac{\delta}{4} \right) \right)}$$

- **Test Error Rate**
- **Training Error Rate**
- **VC Confidence**

$N$: size of training set  
$V$: VC dimension
Statistical Learning Theory

- In pattern classification, the generalization upper bound holds with probability $1-\delta$ (Vapnik et. al.):

$$R(\theta) \leq R_d(\theta) + \sqrt{\frac{C}{N}} \left( \frac{V \log^2(N/V)}{d^2} + \log\left( \frac{1}{\delta} \right) \right)$$

- Test Error
- Margin Error
- VC Confidence

- $V$: VC dimension
- $d$: margin
- $N$: size of training set
- $C$: universal constant
How about using SVM for Speech Recognition?

- Done in some simple ASR tasks:
  - phoneme recognition; speaker recognition
  - small vocabulary isolated speech recognition

- Hard to extend to large-scale continuous speech recognition.

- No significant improvement is reported.
  - still not a main-stream method

- Why?
  - SVM: binary, static classifier.
  - Lack of a proper kernel function to map speech samples from one dynamic high-dimension space to another high-dimension space, which is suitable for linear classifiers.
Large-Margin HMM-based Classifier

separation boundary \( F(X|\Lambda_1) - F(X|\Lambda_2) = 0 \)
Large-Margin HMM-based Classifier

original separation boundary $F(X|\Lambda_1) - F(X|\Lambda_2) = 0$

new separation boundary $F(X|\Lambda'_1) - F(X|\Lambda'_2) = 0$
How to define separation *margin*? (1)

- In 2-class separable problem:
  - For a data token, $x_1$, of class $\Lambda_1$
    \[
    d(x_1) = F(x_1|\Lambda_1) - F(x_1|\Lambda_2) > 0
    \]
  - For a data token, $x_2$, of class $\Lambda_2$
    \[
    d(x_2) = F(x_2|\Lambda_2) - F(x_2|\Lambda_1) > 0
    \]
How to define separation *margin*? (2)

- Extend to multiple-class problem:
  - N classes $\Lambda_1, \Lambda_2, \ldots, \Lambda_N$,
  - For a data token, $x_i$, of class $\Lambda_i$

$$d(x_i) = F(x_i | \Lambda_i) - \max_{j \neq i} F(x_i | \Lambda_j)$$

$$= \min_{j \neq i} \left[ F(x_i | \Lambda_i) - F(x_i | \Lambda_j) \right]$$
Large-Margin Estimation of HMMs

- An $N$-class problem: each class is represented by an HMM

$$\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_N\}$$

- Given a training set $D$, define a subset, called *support token set* $S$, as:

$$S = \{X_i \mid X_i \in D \text{ and } 0 \leq d(X_i) \leq \varepsilon\}$$

- Large-Margin Estimation (*LME*) of HMMs:

$$\hat{\Lambda} = \arg\max_{\Lambda} \min_{X_i \in S} d(X_i) \quad \text{(subject to all } d(X_i) > 0)$$
Large-Margin Estimation of HMMs

- Convert to a minimax optimization problem.

- Assume $X_i$ belongs to class $\Lambda_i$:

\[
\hat{\Lambda} = \arg \min_{\Lambda} \max_{X_i \in S, j \neq i} \left[ F(X_i|\Lambda_j) - F(X_i|\Lambda_i) \right]
\]

subject to constraints:

\[
F(X_i|\Lambda_j) - F(X_i|\Lambda_i) < 0
\]

for all $X_i \in S$ and $j \neq i$. 
Analysis of Margins in CDHMM

- The margin in CDHMM is **unbounded** without additional constraints.
- Adjust CDHMM parameters in certain way to increase the margin unlimitedly.

- Adopt Viterbi approximation:

\[ d_{ij}(X_i) = \mathcal{F}(X_i | \lambda_{W_i}) - \mathcal{F}(X_i | \lambda_j) \]

\[ \approx c_{ij} - \frac{1}{2} \sum_{t=1}^{R} \sum_{d=1}^{D} \left[ \frac{(x_{itd} - \mu_{s_t'} l_t'd)^2}{\sigma^2_{s_t' l_t'd}} - \frac{(x_{itd} - \mu_{s_t'' l_t''d})^2}{\sigma^2_{s_t'' l_t''d}} \right] \]
Analysis of Margins in CDHMM

\[ d_{ij}(X_i) \approx c_{ij} + \sum_{t=1}^{R} \left\{ \sum_{d \in D_{t1}} K_{itd} \cdot (x_{itd} - \mathcal{L}_{itd}) \right\} + \sum_{d \in D_{t2}} \left[ A_{itd} \cdot (x_{itd} - B_{itd})^2 - C_{itd} \right] \]

- Each dimension: independent
- Linear: same variance
- Quadratic: different variances
Margin: Linear Dimensions

\[ d = \frac{\mu_2 - \mu_1}{\sigma^2} \left( x \frac{\mu_2 + \mu_1}{2} \right) \]
Margin: Quadratic Dimensions
Constraints in LME of CDHMM

- Impose constraints to make LME solvable:
  - Linear part: fix the norm of the slope to a constant
    \[ R_1(\Lambda_{W_i}, \Lambda_j \mid X_i) = \sum_{t=1}^{R} \sum_{d \in D_{t1}} K_{itd}^2 = g_{ij}^2 \]
  - Quadratic part: constrain the vertex to a range
    \[ R_2(\Lambda_{W_i}, \Lambda_j \mid X_i) = \sum_{t=1}^{R} \sum_{d \in D_{t2}} (B_{itd} - B_{itd}^{(0)})^2 \leq G_{ij}^2 \]
LME: constrained minimax optimization

- Large Margin Estimation (LME) of CDHMM → a constrained minimax optimization problem

\[ \hat{\Lambda} = \arg \min_{\Lambda} \max_{X_i \in S, j \neq i} \left[ F(X_i | \Lambda_j) - F(X_i | \Lambda_i) \right] \]

subject to constraints:

\[ R_1(\Lambda_{W_i}, \Lambda_j | X_i) = g_{ij}^2 \]
\[ R_2(\Lambda_{W_i}, \Lambda_j | X_i) \leq G_{ij}^2 \]

for all \( X_i \in S \) and \( j \neq i \).
Optimization Methods

- Gradient Descent (GD) Search
  - approx the objective function with a differentiable one
  - cast constraints as penalty terms

- Semi-definite Programming (SDP)
  - math manipulation
  - relaxation
LME Optimization: Gradient Descent

- Approximate $Q(\Lambda)$ with summation of exponentials

\[
Q(\Lambda) = \min_i d(X_i)
\]

\[
Q(\Lambda) \approx Q_\eta(\Lambda) = \frac{1}{\eta} \log \left[ \sum_{X_i \in S, j \neq i} \exp[\eta \cdot d(X_i)] \right]
\]

\[
Q(\Lambda) > Q_\eta(\Lambda) \quad (\eta < 0)
\]

\[
\lim_{\eta \to -\infty} Q_\eta(\Lambda) = Q(\Lambda)
\]

- Constraints $\Rightarrow$ Penalty terms:

\[
O(\Lambda) = Q_\eta(\Lambda) + \tau_1 \cdot P_1(\Lambda) + \tau_2 \cdot P_2(\Lambda)
\]

\[
P_1(\Lambda) = \sum_{i,j,j \neq W_i} \left( R_1(\Lambda_j, \Lambda_{W_i} | X_i) - g_{ij}^2 \right)^2
\]

\[
P_2(\Lambda) = \sum_{i,j,j \neq W_i} \max \left(0, R_2(\Lambda_j, \Lambda_{W_i} | X_i) - G_{ij}^2 \right)^2
\]
LME Optimization: Gradient Descent

- The gradient descent optimization:
  \[
  \hat{\Lambda}'(n+1) = \hat{\Lambda}'(n) + \varepsilon \cdot \frac{\partial O(\Lambda)}{\partial \Lambda} \bigg|_{\Lambda=\Lambda'(n)}
  \]

- Gradient descent optimization:
  - Many parameters to be tuned experimentally:
    - step size, penalty coefficients, \( \eta \), etc.
  - Slow convergence speed.
  - Local optimum.
How to calculate the gradient for continuous density HMM? (1)

\[
\frac{\partial Q_\eta(\Lambda)}{\partial \Lambda} = \sum_{X_i \in S, j \neq i} \exp[\eta \cdot d(X_i)] \cdot \frac{\partial d(X_i)}{\partial \Lambda}
\]

\[
\frac{\partial d(X_i)}{\partial \Lambda_i} = \frac{\partial F(X_i \mid \Lambda_i)}{\partial \Lambda_i}
\]

\[
\frac{\partial d(X_i)}{\partial \Lambda_j} = -\frac{\partial F(X_i \mid \Lambda_j)}{\partial \Lambda_j}
\]
How to calculate the gradient for continuous density HMM? (2)

- Assumption 1: adjust CDHMM mean vectors only
- Assumption 2: diagonal precision matrices
- Assumption 3: use the Viterbi approximation

\[
\mathbf{F}(X_i \mid \Lambda_i) \approx C' - \frac{1}{2} \sum_{t=1}^{T} \sum_{d=1}^{D} r_{s_{t}l_{t}d}^{(i)} (X_{itd} - m_{s_{t}l_{t}d}^{(i)})^2
\]

\[
\mathbf{F}(X_i \mid \Lambda_j) \approx C'' - \frac{1}{2} \sum_{t=1}^{T} \sum_{d=1}^{D} r_{s'_{t}l'_{t}d}^{(j)} (X_{itd} - m_{s'_{t}l'_{t}d}^{(j)})^2
\]
Semi-definite Programming (SDP)

- An active area in optimization community nowadays.
- The standard SDP form

$$\min_{X_1, X_2, \ldots, X_N} \sum_{j=1}^{N} C_j \cdot X_j$$

subject to

$$\sum_{j=1}^{N} A_{ij} \cdot X_j = b_i \quad i = 1, \ldots, M, \ X_j \in \Psi$$

- Linear function of symmetric matrices in semi-definite matrix cone
- SDP can solve nonlinear optimization if configured properly.
- Convex conic optimization $\Rightarrow$ Global optimal solution
- New efficient algorithms are developed.
LME Optimization: SDP

- LME: convert the constrained minimax optimization →
  semi-definite programming (SDP)
- Introduce a new constraint to the minimax optimization problem:

\[
\tilde{\Lambda} = \arg \min_{\Lambda} \max_{X_i \in \mathcal{S}, j \in \Omega, W_i \neq j} \left[ \mathcal{F}(X_i|\lambda_j) - \mathcal{F}(X_i|\lambda_{W_i}) \right]
\]

subject to

\[
R(\Lambda) = \sum_{k=1}^{L} \sum_{d=1}^{D} \frac{(\mu_{kd} - \mu_{kd}^{(0)})^2}{\sigma_{kd}^2} \leq r^2
\]

\[
\mathcal{F}(X_i|\lambda_j) - \mathcal{F}(X_i|\lambda_{W_i}) < 0
\]
LME-SDP: Minimization

- Minimax optimization $\Rightarrow$ minimization
  - Replace $\text{max}$ with a common upper-bound $-\rho$:

  $$\tilde{\Lambda} = \arg \min_{\Lambda, \rho} -\rho$$

  subject to

  $$\mathcal{F}(X_i|\lambda_j) - \mathcal{F}(X_i|\lambda_{W_i}) \leq -\rho$$

  $$R(\Lambda) = \sum_{k=1}^{L} \sum_{d=1}^{D} \left( \frac{\mu_{kd} - \mu_{kd}^{(0)}}{\sigma_{kd}^2} \right)^2 \leq r^2$$

  $$\rho \geq 0.$$
LME-SDP: Matrix Form

- **Transform into matrix form**

  \[
  \min_{Z, \rho} -\rho
  \]

  subject to

  \[
  A_{ij} \cdot Z + \rho \leq c_{ij}
  \]

  \[
  Q \cdot Z \leq r^2
  \]

  \[
  Z = \begin{pmatrix}
  I_D & U \\
  U^T & Y
  \end{pmatrix}
  \]

  \[
  Y = U^T U \quad \rho \geq 0.
  \]

  \[
  A_{i} = \frac{1}{2} \sum_{t=1}^{R} (\tilde{x}_{it} ; e_{it})(\tilde{x}_{it} ; e_{it})^T
  \]

  \[
  Q = \sum_{k=1}^{L} (\tilde{\mu}_{k}^{(0)} ; e_{k})(\tilde{\mu}_{k}^{(0)} ; e_{k})^T
  \]
LME-SDP: Relaxation

- Matrix Relaxation: equality $\Rightarrow$ Inequality

$$Y = U^T U \quad \Leftrightarrow \quad Y - U^T U \succeq 0 \quad \Leftrightarrow \quad Z = \begin{pmatrix} I_D & U \\ U^T & Y \end{pmatrix} \preceq 0$$

- Relaxation to an SDP problem

$$\min_{Z, \rho} -\rho$$

subject to

$$A_{ij} \cdot Z + \rho \leq c_{ij} \quad Z_{1:D,1:D} = I_D$$

$$Q \cdot Z \leq r^2 \quad Z \succeq 0 \quad \rho \geq 0$$
LME-SDP: Relaxation Analysis

- Relaxation – geometry explanation
  - Augment $x$ and $u$ to a higher dimension
  - Solve the problem in the augmented space
LME-SDP: training procedure

Diagram:
- Training Data
- Viterbi Decoding
- Support Tokens
- HMMs
- SDP Solution
- SDP Algorithm

Arrows:
- Input from Training Data to Viterbi Decoding
- Output from Viterbi Decoding to Support Tokens
- Input to HMMs
- Update to SDP Solution
- Input to SDP Algorithm
- Output from SDP Algorithm to SDP Solution
- Output from HMMs to SDP Algorithm
Experiments: overview

- Implemented under the HTK framework.

- Added more training tools:
  - MCE training tool: *HMce.c*
  - LME-GD training tool: *HCLme.c*
  - LME-SDP training tool: C programs + Matlab program + *dsdp* package

- ASR Tasks
  - OGI ISOLET E-set recognition
  - TIDIGITS
Experimental Result: ISOLET

- Feature vector is of 39 dimensions:
  \[(12 \text{ MFCC } + E) + \Delta + \Delta\Delta\]
- MLE models (14 states per HMM) are trained by HTK.
- MLE \(\Rightarrow\) MCE ; MCE \(\Rightarrow\) LME.
- Alphabet (26-letter) recognition (training – 3120 utterances; test – 1560 utterances):
  - OGI (96%), Cambridge (96.73%).
  - Ours: MLE (95.4%), MCE (96.1%), LME (96.92%).
Experimental Result:
ISOLET E-set

- ISOLET E-set: \{B, C, D, E, G, P, T, V, Z\}
- Training: 1080 utterances; Test: 540 utterances
- word accuracy (in %) on ISOLET E-set test data

<table>
<thead>
<tr>
<th></th>
<th>mix-1</th>
<th>mix-2</th>
<th>mix-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>85.56</td>
<td>90.56</td>
<td>91.48</td>
</tr>
<tr>
<td>MCE</td>
<td>91.48</td>
<td>94.07</td>
<td>93.89</td>
</tr>
<tr>
<td>LME-GD</td>
<td>92.96</td>
<td>95.00</td>
<td>94.44</td>
</tr>
<tr>
<td>LME-SDP</td>
<td>92.96</td>
<td>95.19</td>
<td>95.00</td>
</tr>
</tbody>
</table>
Experimental Result: TIDIGITS

- Connected digit strings: ‘1’ to ‘9’ plus ‘oh’ and ‘zero’
- Training with 8623 sentences; test 8700 sentences.
- Feature vector is of 39 dimensions:
  \[(12 \text{ MFCC} + E) + \Delta + \Delta\Delta.\]
- Unknown length digit string recognition.
- Context-independent whole-word HMM models.
- MLE models (12 states per HMM) are trained by HTK.
- MLE \(\Rightarrow\) MCE \(\Rightarrow\) LME.
- MCE/LME training: N-best (N=5) based string-level.
String-Level LME Training

1. Identify support tokens
2. LME optimization
3. Converge or not
Experimental Result - TIDIGITS

- **string accuracy (in %) in test data**

<table>
<thead>
<tr>
<th></th>
<th>ML</th>
<th>MCE</th>
<th>LME- GD</th>
<th>LME-SDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-mix</td>
<td>87.39</td>
<td>93.28</td>
<td>96.23</td>
<td>97.25</td>
</tr>
<tr>
<td>2-mix</td>
<td>94.74</td>
<td>96.06</td>
<td>98.30</td>
<td>98.76</td>
</tr>
<tr>
<td>4-mix</td>
<td>96.52</td>
<td>97.77</td>
<td>98.76</td>
<td>99.11</td>
</tr>
<tr>
<td>8-mix</td>
<td>99.06</td>
<td>98.59</td>
<td>99.13</td>
<td>99.32</td>
</tr>
<tr>
<td>16-mix</td>
<td>98.28</td>
<td>98.89</td>
<td>99.18</td>
<td>99.37</td>
</tr>
<tr>
<td>32-mix</td>
<td>98.66</td>
<td>99.10</td>
<td>99.34</td>
<td>99.47</td>
</tr>
</tbody>
</table>
Experimental Result - TIDIGITS

- WER (in %) on TIDIGITS Test set

<table>
<thead>
<tr>
<th></th>
<th>MLE</th>
<th>MCE</th>
<th>LME -GD</th>
<th>LME-SDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-mix</td>
<td>4.55</td>
<td>2.40</td>
<td>1.28 (47%)</td>
<td>0.93 (61%)</td>
</tr>
<tr>
<td>2-mix</td>
<td>1.86</td>
<td>1.41</td>
<td>0.57 (60%)</td>
<td>0.42 (70%)</td>
</tr>
<tr>
<td>4-mix</td>
<td>1.18</td>
<td>0.76</td>
<td>0.43 (43%)</td>
<td>0.29 (62%)</td>
</tr>
<tr>
<td>8-mix</td>
<td>0.66</td>
<td>0.49</td>
<td>0.30 (39%)</td>
<td>0.23 (53%)</td>
</tr>
<tr>
<td>16-mix</td>
<td>0.57</td>
<td>0.38</td>
<td>0.29 (24%)</td>
<td>0.21 (45%)</td>
</tr>
<tr>
<td>32-mix</td>
<td>0.45</td>
<td>0.30</td>
<td>0.22 (27%)</td>
<td><strong>0.18 (40%)</strong></td>
</tr>
</tbody>
</table>
TIDIGITS Results
LME-GD vs. LME-SDP

Gradient Descent

SDP
### TIDIGITS results
#### LME vs. others

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Model</th>
<th>string error rate (%)</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MLE</strong> (with HTK)</td>
<td>context-indep whole word model</td>
<td>1.34</td>
<td>0.45</td>
</tr>
<tr>
<td><strong>MMIE</strong> (Normandin’94)</td>
<td>context-indep two models per word</td>
<td>0.89</td>
<td>0.29</td>
</tr>
<tr>
<td><strong>MCE</strong> (Juang et. al. ’97)</td>
<td>context-indep whole word model</td>
<td>0.95</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>MCE</strong> (Juang et. al. ’97)</td>
<td>context-dep head-body-tail model</td>
<td>0.72</td>
<td>0.24</td>
</tr>
<tr>
<td><strong>LME-GD</strong></td>
<td>context-indep whole word model</td>
<td>0.66</td>
<td>0.22</td>
</tr>
<tr>
<td><strong>LME-SDP</strong></td>
<td>context-indep whole word model</td>
<td>0.53</td>
<td>0.18</td>
</tr>
</tbody>
</table>
## Summary

- **HMM Estimation methods for ASR**

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Optimization methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Likelihood Estimation (MLE)</td>
<td>EM or Baum-Welch (BW)</td>
</tr>
<tr>
<td>Maximum Mutual Information Estimation (MMIE)</td>
<td>growth-transformation or extended BW (EBW)</td>
</tr>
<tr>
<td>(Maximum Conditional Likelihood)</td>
<td></td>
</tr>
<tr>
<td>Minimum Classification Error (MCE)</td>
<td>gradient descent, GPD Quickprop, etc.</td>
</tr>
</tbody>
</table>

| **Large Margin Estimation (LME)**             | gradient descent                                          |
| **Maximum Relative Margin Estimation (MRME)** | semi-definite programming (SDP)                           |
|                                               | gradient descent                                          |
Large Margin Estimation (LME) vs. Discriminative Training (DT)

- MCE or MMIE is only asymptotic bound of the Bayes error.

\[
R(\theta) \leq - \lim_{N \to \infty} Q_{MMI}(\theta, N)
\]

\[
R(\theta) \leq \lim_{N \to \infty} Q_{MCE}(\theta, N)
\]

- But Vapnik’s generalization bound holds for a finite body of training data.

\[
R(\theta) \leq R_{emp}(\theta) + \sqrt{\frac{1}{N} \left( V \left( \log \frac{2N}{V} + 1 \right) - \log \left( \frac{\delta}{4} \right) \right)}
\]
Ongoing Works

- How to handle training errors?
  - combined objective function: margin + training errors
    \cite{ICASSP06}

- Extend to large-scale subword-based speech recognition:
  - WSJ-5K
  - SPINE