Large margin discriminative learning for acoustic modeling

Fei Sha
Dept. of Computer & Information Science
University of Pennsylvania
Acoustic Modeling

model acoustic properties of speech units

speech signal $\mathcal{A}$

Acoustic models

language models

recognition result $\mathcal{W}$

$$p(\mathcal{W}|\mathcal{A}) \propto p(\mathcal{A}|\mathcal{W})p(\mathcal{W})$$
Many research problems

- What types of units? phonemes
- What acoustic properties & how to represent? standard signal processing
- How to integrate?
- What kind of models? continuous density hidden Markov models
- How to estimate model parameters?
### Phonemes

- **Smallest distinctive sound units in a language**
  - distinguish one word from another
  - use as building blocks for other units

**Example:**

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Example</th>
<th>Phoneme</th>
<th>Example</th>
<th>Phoneme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>/i/</td>
<td>beat</td>
<td>/I/</td>
<td>bit</td>
<td>/e/</td>
<td>bait</td>
</tr>
<tr>
<td>/æ/</td>
<td>bat</td>
<td>/ɛ/</td>
<td>bet</td>
<td>/a/</td>
<td>Bob</td>
</tr>
<tr>
<td>/ɜ/</td>
<td>bird</td>
<td>/ɔ/</td>
<td>about</td>
<td>/ɔ/</td>
<td>bought</td>
</tr>
<tr>
<td>/u/</td>
<td>boot</td>
<td>/U/</td>
<td>book</td>
<td>/o/</td>
<td>boat</td>
</tr>
<tr>
<td>/ay/</td>
<td>buy</td>
<td>/ɔy/</td>
<td>boy</td>
<td>/aw/</td>
<td>down</td>
</tr>
<tr>
<td>/w/</td>
<td>wit</td>
<td>/l/</td>
<td>let</td>
<td>/r/</td>
<td>rent</td>
</tr>
<tr>
<td>/y/</td>
<td>you</td>
<td>/m/</td>
<td>met</td>
<td>/n/</td>
<td>net</td>
</tr>
<tr>
<td>/ŋ/</td>
<td>sing</td>
<td>/b/</td>
<td>bet</td>
<td>/d/</td>
<td>debt</td>
</tr>
<tr>
<td>/ɡ/</td>
<td>get</td>
<td>/p/</td>
<td>pet</td>
<td>/t/</td>
<td>ten</td>
</tr>
<tr>
<td>/k/</td>
<td>kit</td>
<td>/v/</td>
<td>vat</td>
<td>/θ/</td>
<td>thing</td>
</tr>
<tr>
<td>/z/</td>
<td>zoo</td>
<td>/ʒ/</td>
<td>azure</td>
<td>/ʃ/</td>
<td>fat</td>
</tr>
<tr>
<td>/ð/</td>
<td>that</td>
<td>/s/</td>
<td>sat</td>
<td>/ʃ/</td>
<td>shut</td>
</tr>
<tr>
<td>/h/</td>
<td>hat</td>
<td>/ʃ/</td>
<td>judge</td>
<td>/ʃʃ/</td>
<td>church</td>
</tr>
</tbody>
</table>
• **Frame-based feature representation**

- 13 mel-cepstral coefficients (MFCCs) per frame
- frame difference (delta-features) for temporal information

\[ X = \{ x_1, x_2, \ldots, x_T \} \]
Continuous density hidden Markov Models

- **Standard acoustic model**
  - HMMs state transition for modeling temporal dynamics
  - Gaussian mixture models for modeling emission distribution

\[
p(x_t | s_t = j) = \sum_{m=1}^{M} \alpha_{jm} \mathcal{N}(x_t; \mu_{jm}, \Sigma_{jm})
\]

- **Example:**
  - 3 states: left, middle, right
  - forward state transitions.
The problem of parameter estimation

A simplified framework:

• Training data:
  - acoustic features: \( x \in \mathbb{R}^D \)
  - phonetic transcriptions: \( y \in \{1, 2, \ldots, C\} \)

• Model: hidden Markov models with GMMs

\[ p(x, y; \theta) \]

• Output:
  - transitional probabilities
  - means and covariance matrices
Paradigms for parameter estimations

• Maximum likelihood estimation
  - maximize joint likelihood
  
  \[
  \arg \max \sum_{n} \log p(x_n, y_n; \theta)
  \]
  - use simple update procedures (EM algorithms)
  - link indirectly to recognition performance

• Discriminative learning
  - aim to classify and recognize correctly
  - tend to be much more computationally intensive
  - lead to better performance if done right
Ex: discriminative learning methods

- **Conditional maximum likelihood (CML/MMIE)**
  \[
  \arg \max \sum_n \log p(y_b|x_n, \theta)
  \]
  Intuition: raise likelihood for the target class, lower the likelihood for all other classes

- **Minimum classification error (MCE)**
  - raise discriminant score for the target class
  - lower discriminant score for other classes
  - transform & quantify the difference to approximate classification error
Unified framework

\[ \sum_n f \left( \log \left[ \frac{\sum_{\mathcal{W}} p^\alpha(A_n|\mathcal{W})p^\alpha(\mathcal{W})G(\mathcal{W}, \mathcal{W}_n)}{\sum_{\mathcal{W}} p^\alpha(A_n|\mathcal{W})p^\alpha(\mathcal{W})} \right]^{1/\alpha} \right) \]

- Covers many methods
  - ML, MCE, CML/MMIE
  - Minimum Phone Error / Minimum Word Error

Discriminative learning methods generally do much better than ML estimation.
Challenges of discriminative learning methods

• Need a lot of training data
  - large corpus
  - aggressive parameter tying

• Overfitting is a big problem
  - smoothing interpolation between ML and MMI
  - “confusable” set: frame discrimination, MPE/MWE

• Optimization is complicated
  - EM variants: need to set parameters for convergence
  - Gradient methods: mixed success
Focus of this thesis

examine the utility of large margin based discriminative learning algorithms for acoustic-phonetic modeling
Why large margin?

• Discriminative learning paradigm
• Very successful
  - in multiway classification, eg, SVMs
  - in structural classification, eg., M³N
• Theoretically motivated to offer better generalization
• Computationally appealing.
Outline

• Large margin GMMs
  - Label individual samples
  - Parallel to support vector machines

• Crucial extensions for modeling phonemes
  - Handle outliers
  - Classify and recognize sequences

• Summary
  - What has been done
  - What will be done
Large margin
Gaussian mixture modeling
Model each class by an ellipsoid

- centroid:
  \[ \mu_c \in \mathbb{R}^D \]

- orientation matrix:
  \[ \Psi_c \in \mathbb{R}^{D \times D} \succ 0 \]

- offset:
  \[ \theta_c \geq 0 \]
Nonlinear classification

Classify $\mathbf{x} \in \mathbb{R}^D$ by smallest score

$$y = \text{arg min}_c \left\{ (\mathbf{x} - \mu_c)^T \Psi_c (\mathbf{x} - \mu_c) + \theta_c \right\}$$

Mahalanobis distance

equivalent to Gaussian classifiers
Change of representation

- Step 1: collect parameters

\[
\begin{pmatrix}
\mu_c \\
\Psi_c \\
\theta_c
\end{pmatrix}
\rightarrow
\Phi_c =
\begin{bmatrix}
\Psi_c & -\Psi_c \mu_c \\
-\mu_c^T \Psi_c & \mu_c^T \Psi_c \mu_c + \theta_c
\end{bmatrix} \succeq 0
\]

- Step 2: augment inputs

\[
z = \begin{bmatrix} x \\ 1 \end{bmatrix} \in \mathbb{R}^{D+1}
\]
Classification rule

- **Old:**

\[ y = \arg \min_c \left\{ (x - \mu_c)^T \Psi_c (x - \mu_c) + \theta_c \right\} \]

Old representation is **nonlinear** in \( \mu_c \) and \( \Psi_c \).

- **New:**

\[ y = \arg \min_c \left\{ z^T \Phi_c z \right\} \]

New representation is **linear** in \( \Phi_c \).
Large margin learning criteria

**Input:** labeled examples $\{(x_n, y_n)\}$

**Output:** positive semidefinite matrices $\{\Phi_c\}$

**Goal:** classify by at least one unit margin

\[
\forall c \neq y_n, \quad z_n^T \Phi_c z_n - z_n^T \Phi_{y_n} z_n \geq 1
\]

margin = difference of distance between target and competing classes
Decision boundary & margin
Handling infeasible examples

• “Soft” margin constraints with slacks

\[ \forall c \neq y_n, \quad z_n^T \Phi_c z_n + \xi_{nc} - z_n^T \Phi_{y_n} z_n \geq 1 \]

• Discriminative loss function

Infeasible examples are handled through:

- Hinge loss
- 0/1 loss
- Convex but only a surrogate!
Learning via convex optimization

- **Balance total slacks and scale**

\[
\begin{align*}
\text{minimize} & \quad \sum_{n,c} \xi_{nc} + \kappa \sum_c \text{trace}(\Psi_c) \\
\text{subject to} & \quad z_n^T \Phi_c z_n + \xi_{nc} - z_n^T \Phi_{y_n} z_n \geq 1 \\
& \quad \xi_{nc} \geq 0, \quad \forall n, c \neq y_n \\
& \quad \Phi_c \succeq 0
\end{align*}
\]

- **Properties**
  - instance of semidefinite programming (SDP)
  - convex problem: no local optimum
  - efficiently solvable with gradient methods
General setting

• Model each class by $M$ ellipsoids

$$\{\Phi_{cm}\}$$

• Assume knowledge of “proxy label”

$$m_n = \arg\min_m z_n^T \Phi y_n m z_n$$
Large margin constraints

\[ \forall c \neq y_n \quad \forall m \quad z_n^T \Phi_{cm} z - z_n^T \Phi_{ynm} z \geq 1 \]

- Score for all competing centroids
- Score to the proxy centroid
Handle large number of constraints

\[ \forall c \neq y_n \quad \min_m (z_n^T \Phi_{cm} z - z_n^T \Phi_{ynmn} z) \geq 1 \]

softmax trick to squash many constraints into one

\[ \forall c \neq y_n \quad -\log_m \sum_m e^{-(z_n^T \Phi_{cm} z - z_n^T \Phi_{ynmn} z)} \geq 1 \]
Close the loop: proxy labels

- Step 1: maximum likelihood parameter estimation

\[ \Phi_{ML} = \arg \max \sum_n \log p(x_n, y_n; \Phi) \]

- Step 2: find the closest mixture

\[ m_n = \arg \min_m z_n^T \Phi_{ML} y_n m z_n \]
Optimization problem

\[
\begin{align*}
\text{minimize} & \quad \sum_{n,c} \xi_{nc} + \kappa \sum_c \text{trace}(\Psi_{cm}) \\
\text{subject to} & \quad - \log \sum_m e^{-\left( z_n^T \Phi_{cm} z - z_n^T \Phi_{ynm} z \right)} + \xi_{nc} \geq 1 \\
& \quad \xi_{nc} \geq 0, \quad \forall n, c \neq y_n \\
& \quad \Phi_{cm} \succeq 0
\end{align*}
\]

- Properties:
  - no longer an instance of SDP
  - still convex problem
Application: handwritten digit recognition

● General setup
  - train GMMs with maximum likelihood estimation
  - use GMMs to infer proxy labels
  - train large margin GMMs

● Data sets
  - MNIST handwritten digit recognition
  - 60K training & 10K testing, ten-fold cross validation
  - reduce dimensionality from 28x28 to 40 with PCA
Result: classification error

• Comparable to best SVM result (1.0% - 1.4%)
• training time: 1 - 10 minutes
Result: digit prototypes

- EM: typical images
- Large margin: supposition of images of different digits
Crucial extensions for acoustic modeling

- Handle statistical outliers
- Classify sequence
- Recognize sequence
Extension 1: outliers

dominate loss function

want to focus on these points!
Remedy

equalize contributions from outliers

estimate weighting factors from ML estimated GMMs

minimize

\[
\sum_{n,c} \alpha_n \xi_{nc} + \kappa \sum_c \text{trace}(\Phi_c)
\]

subject to

\[
\begin{align*}
z_n^T \Phi_c z_n + \xi_{nc} - z_n^T \Phi_{y_n} z_n & \geq 1 \\
\xi_{nc} & \geq 0, \quad \forall n, c \neq y_n \\
\Phi_c & \succeq 0
\end{align*}
\]
Extension 2: phonetic classification

• Input:
  - Segmented speech signals
  - Each segment is a phonetic unit

• Output:
  - Phonetic label for each segment

• Evaluation:
  - Percentage of misclassified segments

known phonetic segmentation

known phonetic segmentation

bcl l aa kcl t ey bcl
Algorithm

- Use average score to setup margin constraint

- Input: $\mathbf{x}_{n1}, \mathbf{x}_{n2}, \ldots, \mathbf{x}_{n\ell}$
- Output: $y_n$

$$\forall c \neq y_n, \quad \frac{1}{\ell} \sum_p z_{np}^T \Phi_c z_{np} - \frac{1}{\ell} \sum_p z_{np}^T \Phi_{y_n} z_{np} \geq 1$$
Application: phonetic classification

- **General setup**
  - train baseline GMMs with ML
  - use baseline GMMs to infer proxy labels
  - use baseline GMMs to detect outliers
  - train large margin GMMs

- **Data**
  - TIMIT speech database: 1.2m frames, 120K segments for training
  - Very well studied benchmark problem
Result: phonetic classification error

- Baseline EM
- Large margin

Best reported result

Number of mixture components: 1, 2, 4, 8, 16
Extension 3: phonetic recognition

• **Input:**
  - unsegmented speech
  - frames span more than one phone

• **Output:**
  - phonetic segmentation
  - phonetic labels

• **Evaluation metrics:**
  - multiple metrics
  - some relates more to word error rate
  - output phonetic segmentation; inferred indirectly from frame-level phonetic labels.
Algorithm

- **Goal:** separate target sequences from incorrect sequences by large margin

- **Model:** fully observable Markov model

  state sequence $\leftrightarrow$ phoneme label sequence

```
states

inputs
```
### Discriminant score for sequences

\[ D(X, s) = \sum_t \log a(s_{t1}, s_t) - \sum_{t=1}^T z_t^T \Phi_{st} z_t \]

- Analogous to the log probability in CD-HMMs
  - transition scores
  - emission scores
- Extendable to more complicated features
Large margin constraints

∀s ≠ y, \( D(X, y) - D(X, s) \geq h(s, y) \)

How to handle exponential number of constraints?

Hamming distance [Taskar et al 03]
-\mathcal{D}(X, y) + \max_{s \neq y} \{ h(s, y) + \mathcal{D}(X, s) \} \leq 0

squash into one constraint with softmax

\mathcal{D}(X, y) + \log \sum_{s \neq y} e^{h(s, y) + \mathcal{D}(X, s)} \leq 0

tractable with dynamic programming
Optimization

\[
\begin{align*}
\min & \quad \sum_n \xi_n + \gamma \sum_c \text{trace}(\Psi_c) \\
\text{s.t.} & \quad -D(X_n, y_n) + \log \sum_{s \neq y_n} e^{h(s, y_n)} + D(X_n, s) \leq \xi_n, \\
& \quad \xi_n \geq 0, \quad n = 1, 2, \ldots, N \\
& \quad \Phi_c \succ 0, \quad c = 1, 2, \ldots, C
\end{align*}
\]

- convex optimization
- main computation effort on computing the gradients
Results

- **General setup**
  - train baseline GMMs with ML
  - use baseline GMMs to infer proxy labels
  - use proxy label sequence as target state sequence
  - train large margin GMMs

- **Data**
  - TIMIT speech database: 1.2m frames, 3000 sentences
  - Very well studied benchmark problem
Results: phone error rate

<table>
<thead>
<tr>
<th>Number of Mixture Components</th>
<th>Baseline EM</th>
<th>Large Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>42%</td>
<td>30%</td>
</tr>
<tr>
<td>2</td>
<td>38%</td>
<td>34%</td>
</tr>
<tr>
<td>4</td>
<td>34%</td>
<td>30%</td>
</tr>
<tr>
<td>8</td>
<td>26%</td>
<td>26%</td>
</tr>
</tbody>
</table>
Result: is “sequential” really important?

<table>
<thead>
<tr>
<th>Frame</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>38%</td>
<td>26%</td>
</tr>
<tr>
<td>35%</td>
<td>29%</td>
</tr>
<tr>
<td>32%</td>
<td>29%</td>
</tr>
<tr>
<td>29%</td>
<td>26%</td>
</tr>
<tr>
<td>26%</td>
<td></td>
</tr>
</tbody>
</table>
Result: relative error reduction over baseline

![Chart showing relative error reduction over baseline for different values. The chart compares MMI and Margin for values 1, 2, 4, and 8. The error reduction percentages are 24%, 19%, and 14% respectively.]
What has been achieved?

- **Algorithms**
  - Large margin GMMs for multiway classification
  - Large margin GMMs for sequential classification

- **Advantages over traditional methods**
  - Computationally appealing thru convexity
  - Well motivated training criteria for generalization

- **Experimental results**
  - Able to train on millions of data points
  - Improve over baseline and other competitive methods
Proposed Future Work

• **Goals:**
  - is margin really helpful in getting better performance?
  - can the formulation of the optimization be improved?

• **Comparison to standard ASR algorithms:**
  - MMI/CML, MCE
  - investigate the relation with MPE/MWE

• **Comparison to structural learning methods**
  - recent advances in machine learning community
  - algorithms: extragradient (Taskar, 2006), subgradient method (Bagnell, 2006)