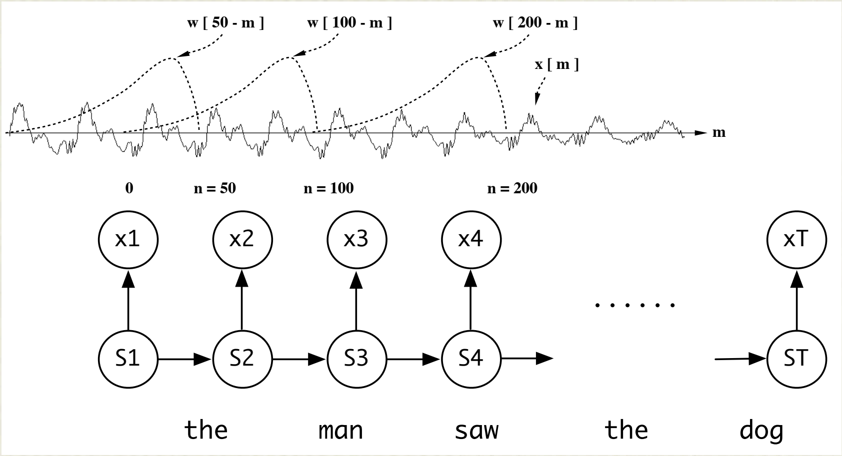
Online Learning of Large Margin HMMs for Automatic Speech Recognition

Lawrence Saul
Department of Computer Science and Engineering
UC San Diego

Joint work with **Chih-Chieh Cheng** (UCSD) and **Fei Sha** (USC)

Speech recognition since 1980s



- **Hidden Markov models (HMMs)**
  - Hidden states: phone/word classes (s_1, s_2, ..., s_T)
  - Observations: acoustic feature vectors (x_1, x_2, ..., x_T)
- **Inference and Learning**
  - Viterbi algorithm for decoding
  - Forward-backward algorithms for sufficient statistics

Types of learning

- **Maximum likelihood estimation** (ML)

  + simple updates, monotonic convergence

  - model mismatch, wrong objective

  $$\boxed{p(x|s)}$$

- **Discriminative training**

  + minimize error rates

  - more complicated, expensive

  $$\boxed{p(s|x) = \frac{p(x|s)p(s)}{\sum_{s'} p(x|s')p(s')}}$$

- **Online learning**

  + scales well to large data sets

  - potential instability

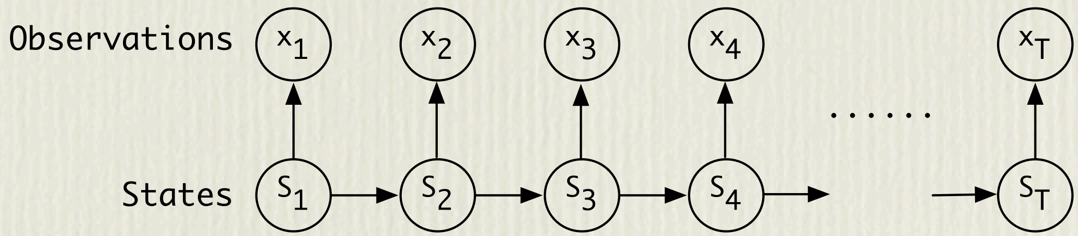
  perceptron training of **discrete** HMMs (Collins, 2002)

Outline

- Motivation and overview
- **Mistake-driven learning in CD-HMMs**
- Large margins: do they help?
- Acoustic feature adaptation
- What’s next?

Continuous-density HMMs

- **Joint distribution**



Joint Distribution $\mathcal{P}(\mathbf{s}, \mathbf{x}) = \mathcal{P}(s_1) \prod_{t=1}^{T-1} \mathcal{P}(s_{t+1}|s_t) \prod_{t=1}^{T} \mathcal{P}(x_t|s_t)$

- **Emission densities** are parameterized by Gaussian mixture models (GMMs):

$$\mathcal{P}(x|s) = \sum_c \frac{\omega_{sc}}{\sqrt{(2\pi)^d |\Sigma_{sc}|}} e^{-\frac{1}{2}(x-\mu_{sc})^\top \Sigma_{sc}^{-1}(x-\mu_{sc})}.$$

- **Maximum likelihood estimation** (MLE)

$$\Theta^{\mathrm{MLE}} = \mathrm{argmax}_\Theta \sum_{n=1}^{N} \log \mathcal{P}(\mathbf{s}_n, \mathbf{x}_n|\Theta)$$

Recognition with CD-HMMs

- **Discriminant function:**

$$\mathcal{D}(\mathbf{x},\mathbf{s}) = \log \mathcal{P}(s_1) + \sum_{t=1}^{T-1} \log \mathcal{P}(s_{t+1}|s_t) + \sum_{t=1}^{T} \log \mathcal{P}(x_t|s_t)$$

- **Correct recognition if:**

$$\forall \mathbf{s} \neq \mathbf{y}, \quad \mathcal{D}(\mathbf{x},\mathbf{y}) > \mathcal{D}(\mathbf{x},\mathbf{s})$$

$\mathbf{y}$: correct transcription of the observation $\mathbf{x}$

$\mathbf{s}$: arbitrary transcription

Online Updating

- For each $\mathbf{x}_n$ in the training set
  - compute **Viterbi** decoding sequence $\mathbf{s}_n^*$
    $$\mathbf{s}_n^* = \mathrm{argmax}_{\mathbf{s}} \mathcal{D}(\mathbf{x}_n, \mathbf{s})$$
  - compare to **ground truth** sequence $\mathbf{y}_n$
  - **update** if $\mathbf{s}_n^* \neq \mathbf{y}_n$
    $$\Theta \leftarrow \Theta + \eta \frac{\partial}{\partial \Theta} \left[\mathcal{D}(\mathbf{x}_n, \mathbf{y}_n) - \mathcal{D}(\mathbf{x}_n, \mathbf{s}_n^*) \right]$$
- Iterate until algorithm converges or no longer reduces recognition errors

Devil in the details

- How to parameterize CD-HMMs for online learning?
- How to enforce constraints on parameters?
- How to dampen fluctuations in decision boundary?

GMMs – a closer look

- Conventionally parameterized in terms of **means**, **covariance matrices**, and **mixture weights**.

- **Gradient-based learning** for component c of state s :

$$\begin{pmatrix} \nu \\ \mu \\ \Sigma \end{pmatrix}_{sc} \leftarrow \begin{pmatrix} \nu \\ \mu \\ \Sigma \end{pmatrix}_{sc} + \begin{pmatrix} \eta_\nu & 0 & 0 \\ 0 & \eta_\mu & 0 \\ 0 & 0 & \eta_\Sigma \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial \nu} \\ \frac{\partial}{\partial \mu} \\ \frac{\partial}{\partial \Sigma} \end{pmatrix}_{sc} [\mathcal{D}(\mathbf{x}_n, \mathbf{y}_n) - \mathcal{D}(\mathbf{x}_n, \tilde{\mathbf{s}}_n^*)]$$

Empirically difficult to tune multiple learning rates:
many gradient-based systems only adapt GMM means.

Reparameterization

- **Change of variables**

For each mixture component, aggregate Gaussian parameters into a single positive semidefinite matrix:

$$\Phi = \begin{array}{c} \overbrace{\hphantom{\Sigma^{-1} \quad -\Sigma^{-1}\mu}}^{d+1} \\ \left[\begin{array}{c|c} \overbrace{\quad\Sigma^{-1}\quad}^{d} & -\Sigma^{-1}\mu \\ \hline -\mu^{T}\Sigma^{-1} & \gamma \end{array}\right] \end{array} \quad \text{where} \quad \gamma = \log[(2\pi)^d|\Sigma|] + \mu^\top\Sigma^{-1}\mu$$

- **Likelihood computation**

$$\log \mathcal{P}(x|s) = -\frac{1}{2} z^T \Phi_s z \quad \text{where} \quad z = \begin{bmatrix} x \\ 1 \end{bmatrix}$$

Reparameterized Update

$$\Phi_{sc} \leftarrow \Phi_{sc} + \eta \frac{\partial}{\partial \Phi_{sc}} [\mathcal{D}(\mathbf{x}_n, \mathbf{y}_n) - \mathcal{D}(\mathbf{x}_n, \mathbf{s}_n^*)]$$

- **Problem:**

  Update can violate **positive semidefiniteness** of matrix Φ_{sc}.

- **Solution:**

  Follow each update **by projecting** Φ_{sc} back to cone of positive semidefinite matrices.

Reparameterized Update

$$\Phi_{sc} \leftarrow \Phi_{sc} + \eta \frac{\partial}{\partial \Phi_{sc}} [\mathcal{D}(\mathbf{x}_n, \mathbf{y}_n) - \mathcal{D}(\mathbf{x}_n, \mathbf{s}_n^*)]$$

- **Problem:**

  Update can violate **positive semidefiniteness** of matrix Φ_{sc}.

- **Solution:**

  Follow each update **by projecting** Φ_{sc} back to cone of positive semidefinite matrices.

- **Problem:**

  Projected gradient methods converge **much slower** than unconstrained methods.

Matrix factorization

- **Yet another reparametrization**

  Remove constraint via matrix square root:

$$\Phi_{sc} = \Lambda_{sc}\Lambda_{sc}^T$$

- **New update rule:**

$$\Lambda_{sc} \leftarrow \Lambda_{sc} + \eta \frac{\partial}{\partial \Lambda_{sc}} [\mathcal{D}(\mathbf{x}_n, \mathbf{y}_n) - \mathcal{D}(\mathbf{x}_n, \mathbf{s}_n^*)]$$

+ unconstrained update
– local minima?
– which matrix square root?

Dampening fluctuations

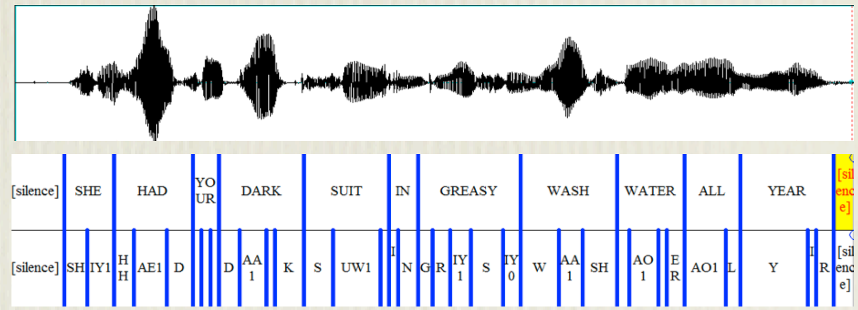
- **Cumulative averaging**

  Borrow idea from "averaged" perceptrons:

$$\tilde{\Phi}^{(i)} = \frac{1}{i} \sum_{j} \Phi^{(j)}$$

- **Smoothed parameter trajectories**
  - averaged $\tilde{\Phi}$ changes more slowly than non-averaged Φ
  - used only for testing, not training

Experiments



- **Phonetic transcription on TIMIT corpus**
  - 39 phone classes
  - Frames of speech:

    1.1M training, 120K development, 56K test

- **Evaluation**

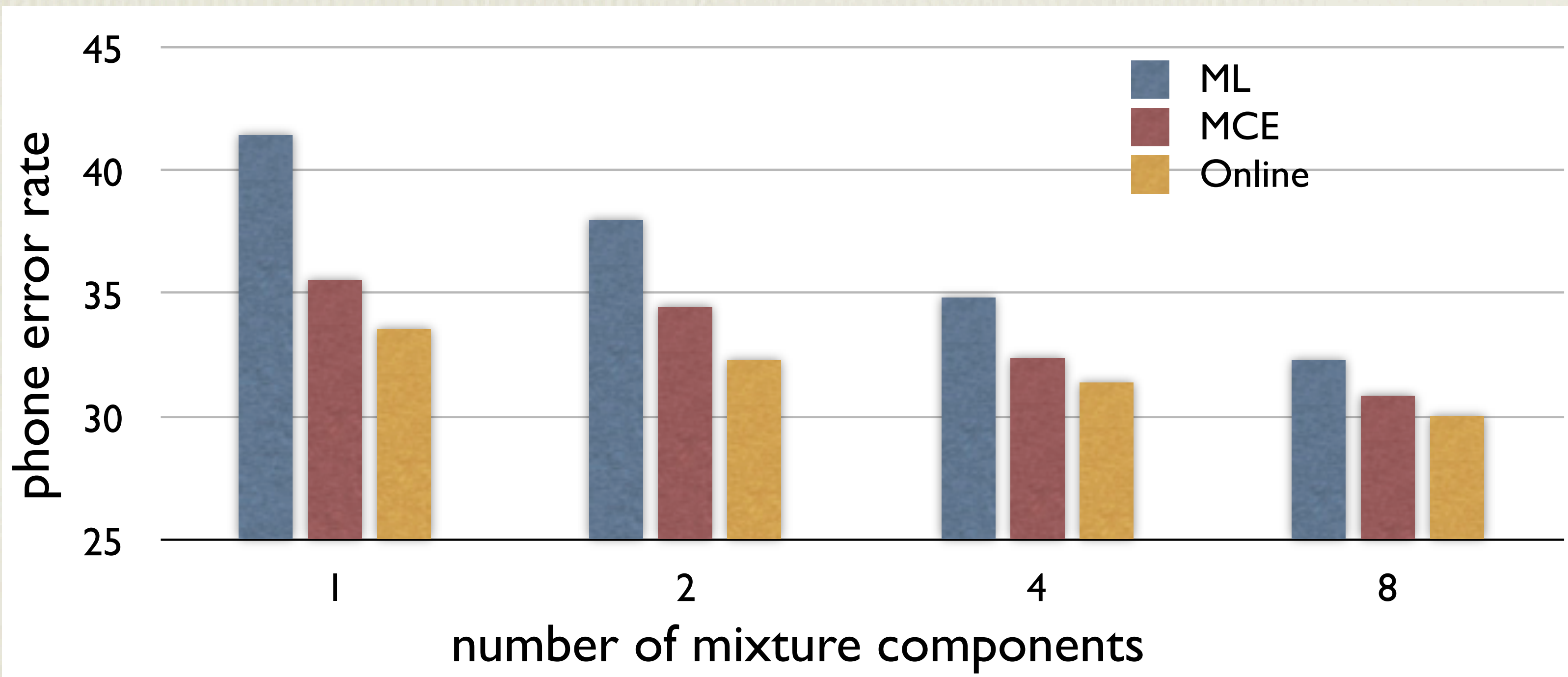
  Compare **recognized** vs **manual** transcriptions:
  - **Frame error rate (FER)**: % of misclassified frames
  - **Phone error rate (PER)**: edit distance by alignment

Batch versus Online

ML = maximum likelihood estimation (batch)

MCE = minimum classification error (batch)

Online (best configuration)



Devil in the details

Training	FER (%)
Batch ML	30.7
Online (w/o reparametrization)	33.9
Online (w/o factorization)	30.9
Online (Cholesky)	31.4
Online (w/o averaging)	35.2
Online (w/o MLE initialization)	36.2
Online (init+SVD+averaging)	**28.8**

Outline

- Motivation and overview
- Mistake-driven learning in CD-HMMs
- **Large margins: do they help?**
- Acoustic feature adaptation
- What's next?

Large Margin Training

- **Goal**

  Attempt to separate scores of correct and incorrect transcriptions by a large margin.

- **Motivation**

  Balance minimization of empirical error rate versus generalization on unseen data.

- **Large margin criterion**

$$\forall \mathbf{s} \neq \mathbf{y}, \quad \mathcal{D}(\mathbf{x}, \mathbf{y}) > \mathcal{D}(\mathbf{x}, \mathbf{s}) + \rho \mathcal{H}(\mathbf{s}, \mathbf{y})$$

$\mathcal{H}(\mathbf{s}, \mathbf{y})$ Hamming distance

$\rho > 0$ margin scaling factor

Online update rule

- For each $\mathbf{x}_n$ in the training set
  - compute the **margin-based decoding** sequence $\tilde{\mathbf{s}}_n^*$

$$\boxed{\tilde{\mathbf{s}}_n^* = \mathrm{argmax}_{\mathbf{s}}[\mathcal{D}(\mathbf{x}_n, \mathbf{s}) + \rho\mathcal{H}(\mathbf{s}, \mathbf{y})]}$$

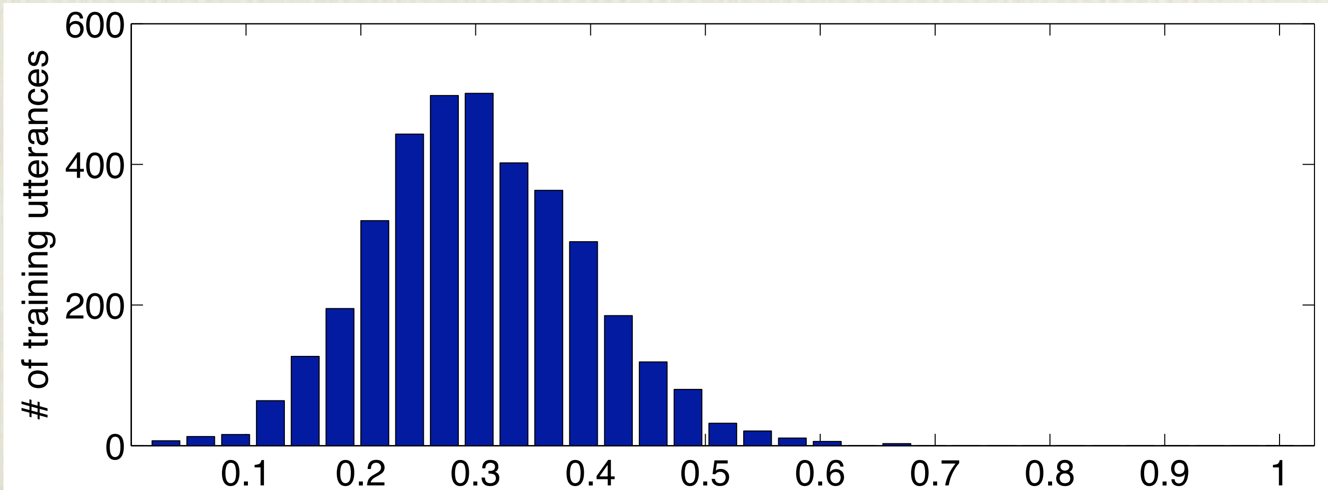
  - compare to **ground truth** sequence $\mathbf{y}_n$
  - **update** if $\tilde{\mathbf{s}}_n^* \neq \mathbf{y}_n$

$$\boxed{\Theta \leftarrow \Theta + \eta \frac{\partial}{\partial\Theta} [\mathcal{D}(\mathbf{x}_n, \mathbf{y}_n) - \mathcal{D}(\mathbf{x}_n, \tilde{\mathbf{s}}_n^*)]}$$

- iterate until the algorithm converges or no longer reduces recognition errors

Margin-based decoding

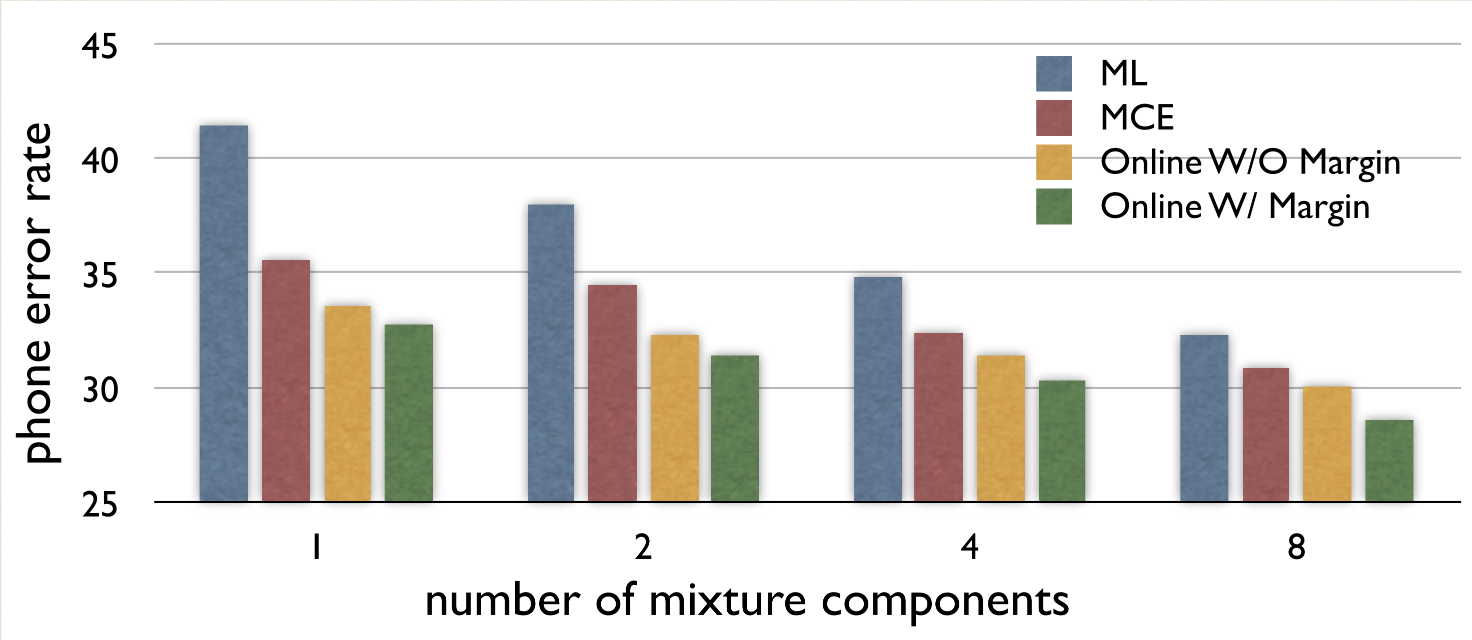
$$\mathbf{s}_n^* = \mathrm{argmax}_{\mathbf{s}} \mathcal{D}(\mathbf{x}_n, \mathbf{s})$$
$$\tilde{\mathbf{s}}_n^* = \mathrm{argmax}_{\mathbf{s}} [\mathcal{D}(\mathbf{x}_n, \mathbf{s}) + \rho \mathcal{H}(\mathbf{s}, \mathbf{y})]$$



Normalized Hamming Distance $\mathcal{H}(\mathbf{s}^*, \tilde{\mathbf{s}}^*)/\mathrm{length}(\mathbf{s}^*)$

Yields very different competing transcriptions!

Do large margins help? Yes.



- **MLE** = maximum likelihood estimation (batch)
- **MCE** = minimum classification error (batch)
- **Online w/o margin** = online algorithm for CD-HMMs
- **Online w/ margin** = online algorithm for large margin training

Outline

- Motivation and overview
- Mistake-driven learning in CD-HMMs
- Large margins: do they help?
- **Acoustic feature adaptation**
- What's next?

Acoustic features

- **Standard front end**

  Compute 13 cepstral features from each 30 ms window of speech.

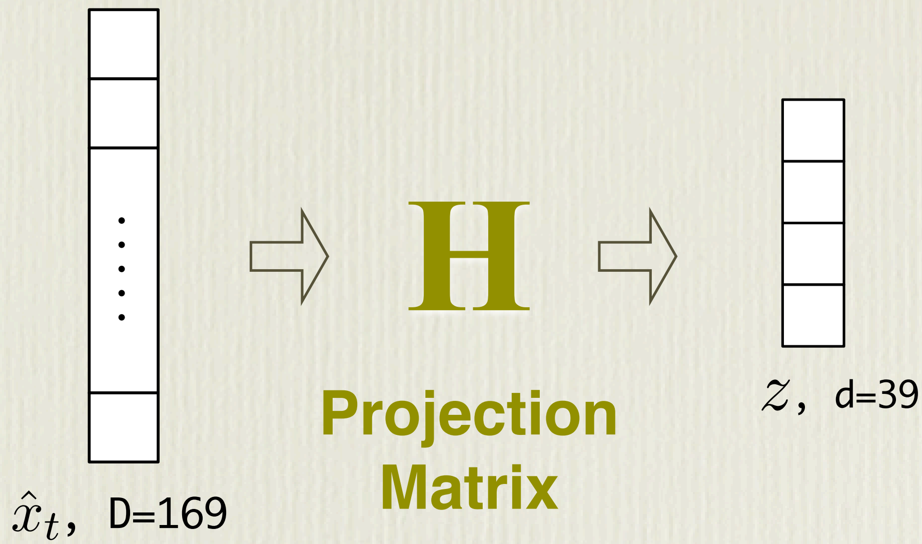
- **Context modeling**

  Incorporate features from adjacent windows into observations of CD-HMMs.

- **Scaling of model size**

  (# GMM parameters) ~ (# features)^2

Acoustic feature adaptation



- **Incorporating context**
  - Concatenate features from 13 adjacent windows.
  - Project into a lower dimensional subspace.

- **End-to-end learning**

How to adapt GMM parameters Φ_{sc} along with projection $\mathbf{H}$?

Online Optimization

- **Approach**

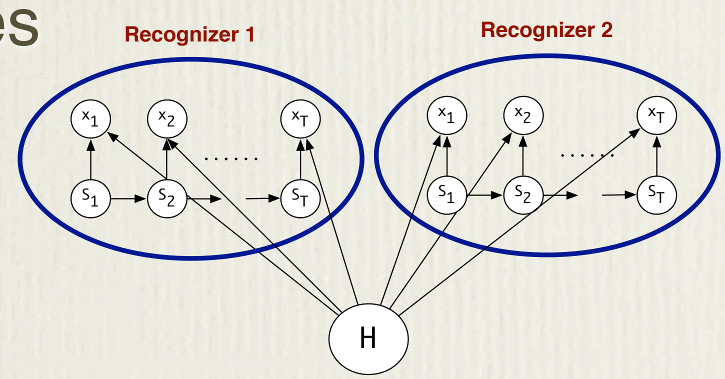
  Maximize margin by alternatively updating projection matrix H and GMM parameters Φ_{sc}.

- **Problem**

  Small changes in H (from one utterance) result in big changes to recognizer (across all phonemes).

- **Solutions**

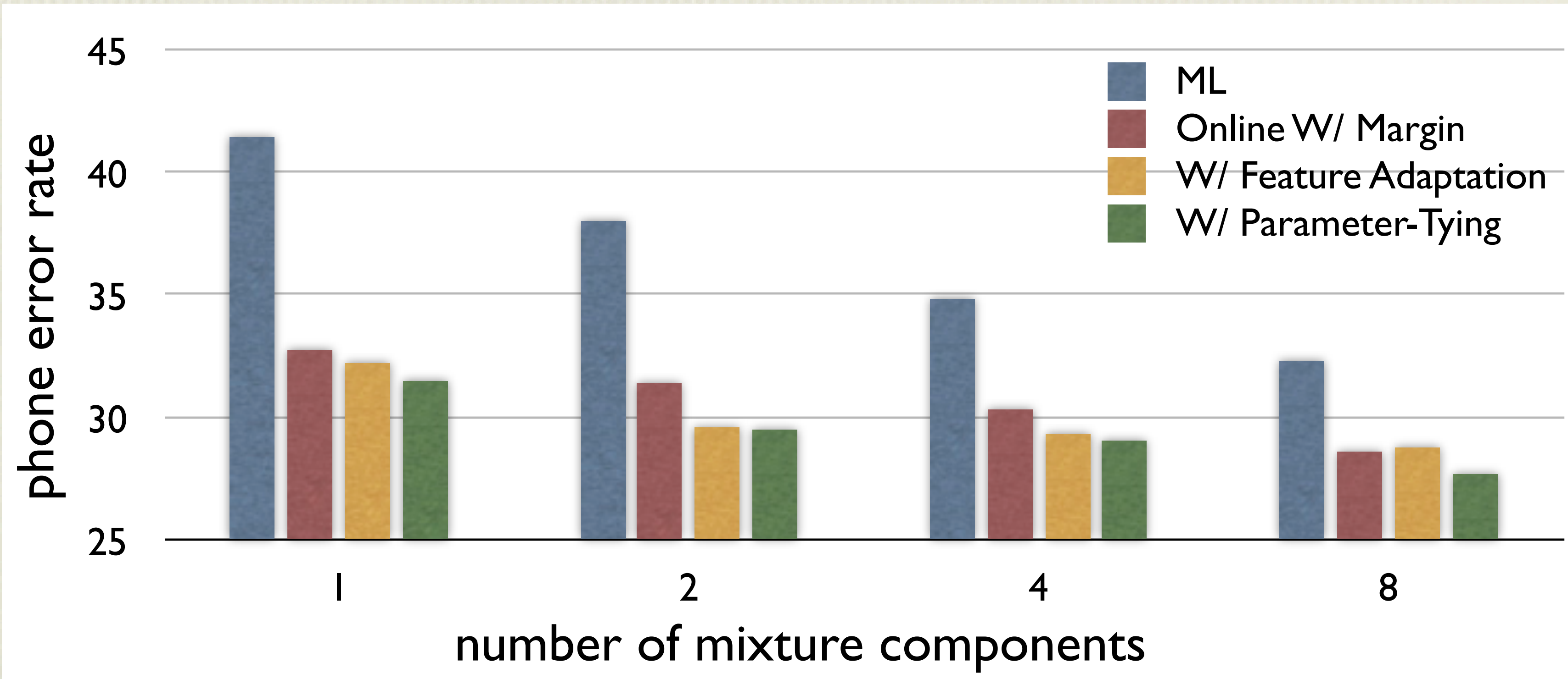
  1. Mini-batches of training utterances
  2. Parameter-tying (of H) across different **recognizers**



Experiments

- **Acoustic features**
  - $\hat{x}$ = 13 MFCCs across 13 consecutive frames (D=139)
  - z = lower-dimensional linear projection of $\hat{x}$ (d=39)
  - H = projection matrix initialized to simulate differencing operations for 13 MFCCs + 13Δ + 13$\Delta\Delta$
- **End-to-end large-margin training**
  - Initialize with maximum likelihood CD-HMMs
  - Alternately optimize H and Φ

Results



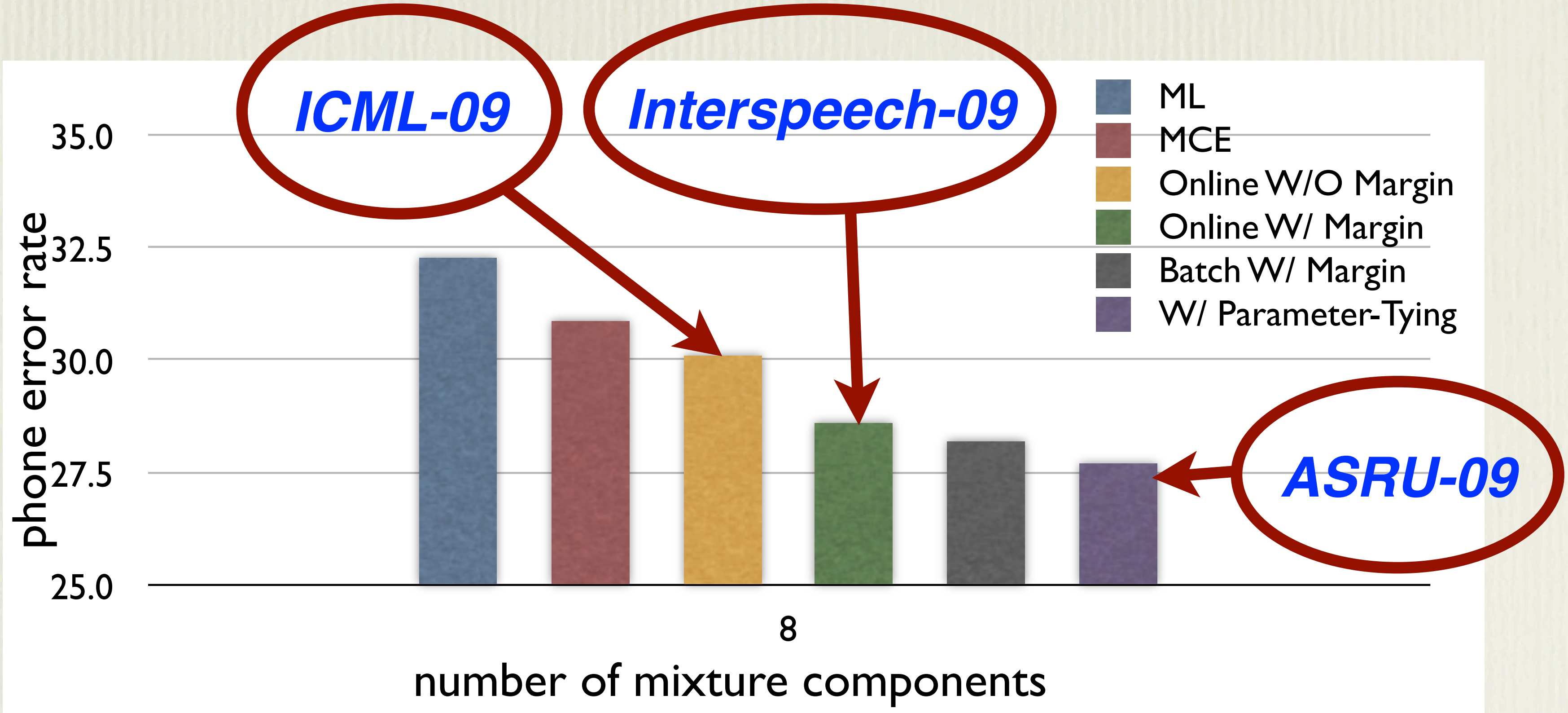
Feature adaptation works best with parameter-tying

Summary

How to improve discriminative training of CD-HMMs with online updates?

- **Best practices:**
  - **Reparameterization** $\Phi = \begin{bmatrix} \Sigma^{-1} & -\Sigma^{-1}\mu \\ -\mu^\top\Sigma^{-1} & \mu^\top\Sigma^{-1}\mu + \gamma \end{bmatrix}$
  - **Factorization** $\Phi = \Lambda\Lambda^\top$
  - **Averaging** $\tilde{\Phi}^{(i)} = \frac{1}{i}\sum_j \Phi^{(j)}$
  - **Large margin** $\tilde{\mathbf{s}}_n^* = \mathrm{argmax}_{\mathbf{s}}[\mathcal{D}(\mathbf{x}_n, \mathbf{s}) + \rho\mathcal{H}(\mathbf{s}, \mathbf{y})]$
  - **Feature adaptation with parameter-tying**
- **Did we succeed?**

Improvement over time



Online methods ultimately beat our best batch implementation.

Outline

- Motivation and overview
- Mistake-driven learning in CD-HMMs
- Large margins: do they help?
- Acoustic feature adaptation
- **What's next?**

What's next?

- **Scaling up**
  - larger corpora
  - word recognition (not phone recognition)
  - context-dependent (triphone) HMMs
  - word lattices for large-vocabulary ASR
- **Fast adaptation**
  - new speakers
  - infinite data (e.g., refreshed daily)

What's next? (con't)

- **Other models and loss functions**
  - Direct loss minimization (McAllester et al, 2010)
  - Hidden-unit conditional random field (van der Maaten et al, 2011)
  - Edit distance (versus Hamming distance)

What's next? (con't)

- **Other models and loss functions**
  - Direct loss minimization (McAllester et al, 2010)
  - Hidden-unit conditional random field (van der Maaten et al, 2011)
  - Edit distance (versus Hamming distance)

See you at the next workshop ...

Publications

C.-C. Cheng, F. Sha, and L. K. Saul (2010). **Online Learning and Acoustic Feature Adaptation in Large Margin Hidden Markov Models**. In *IEEE Journal of Selected Topics in Signal Processing* 4(6): 926-942.

C.-C. Cheng, F. Sha, and L. K. Saul (2009). **Large Margin Feature Adaptation for Automatic Speech Recognition**. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU-09)*, pages 87-92. Merano, Italy.

C.-C. Cheng, F. Sha, and L. K. Saul(2009). **A fast online algorithm for large margin training of continuous-density hidden Markov models.** In *Proceedings of the Tenth Annual Conference of the International Speech Communication Association (Interspeech-09)*, pages 668-671. Brighton, UK.

C.-C. Cheng, F. Sha, and L. K. Saul (2009). **Matrix updates for perceptron training of continuous-density hidden Markov models**. In *Proceedings of the Twenty Sixth International Conference on Machine Learning (ICML-09)*, pages 153- 160. Montreal, Canada.