Frame Discrimination Training of HMMs for Large Vocabulary Speech Recognition

Dan Povey, Phil Woodland

Cambridge University Engineering Department

Overview

Frame Discrimination:

- Discriminative technique developed by Kapadia at CUED
- Objective function related to Maximum Mutual Information (MMI) objective function
- Results on isolated digit and letter recognition were promising

Current work:

- Implement Frame Discrimination for LVCSR
- Computational problem
- Roadmap algorithm developed to speed up computation
- Computation 3 times faster than a previous implementation of MMI using lattices
- Results as good or better than MMI

MMI and Frame Discrimination

MMI:

$$\begin{aligned}\mathcal{F}_\lambda &= \sum_{r=1}^{R} \log \frac{P_\lambda(\mathcal{O}_r|\mathcal{M}^{w_r})P(w_r)}{\sum_{\hat{w}} P_\lambda(\mathcal{O}_r|\mathcal{M}^{\hat{w}})P(\hat{w})} \\ &= \sum_{r=1}^{R} \log \frac{P_\lambda(\mathcal{O}_r|\mathcal{M}^{w_r})}{P_\lambda(\mathcal{O}_r|\mathcal{M}^{\text{gen}})}\end{aligned}$$

Discriminate against general model of speech $\mathcal{M}^{\text{gen}}$ (recognition model).

FD:

$$\mathcal{F}_\lambda = \sum_{r=1}^{R} \log \frac{P_\lambda\left(\mathcal{O}_r|\mathcal{M}^{w_r}\right)}{P_\lambda\left(\mathcal{O}_r|\mathcal{N}\right)}$$

- HMM $\mathcal{N}$ is one state HMM: weighted sum of all states in $\mathcal{M}^{\text{gen}}$
- Weights derived from Forward-Backward alignment data

Computation for Frame Discrimination training

- Extended Baum-Welch (EBW) update equations used
- Align speech data to HMMs
- Calculation dominated by computation of Gaussian probabilities

For transcription:

- Use Forward-Backward algorithm using beam pruning
- Perhaps only 100 Gaussians per time frame $\rightarrow$ fast.

For the denominator model $\mathcal{N}$:

- Tens of thousands of Gaussians per time frame: impractical

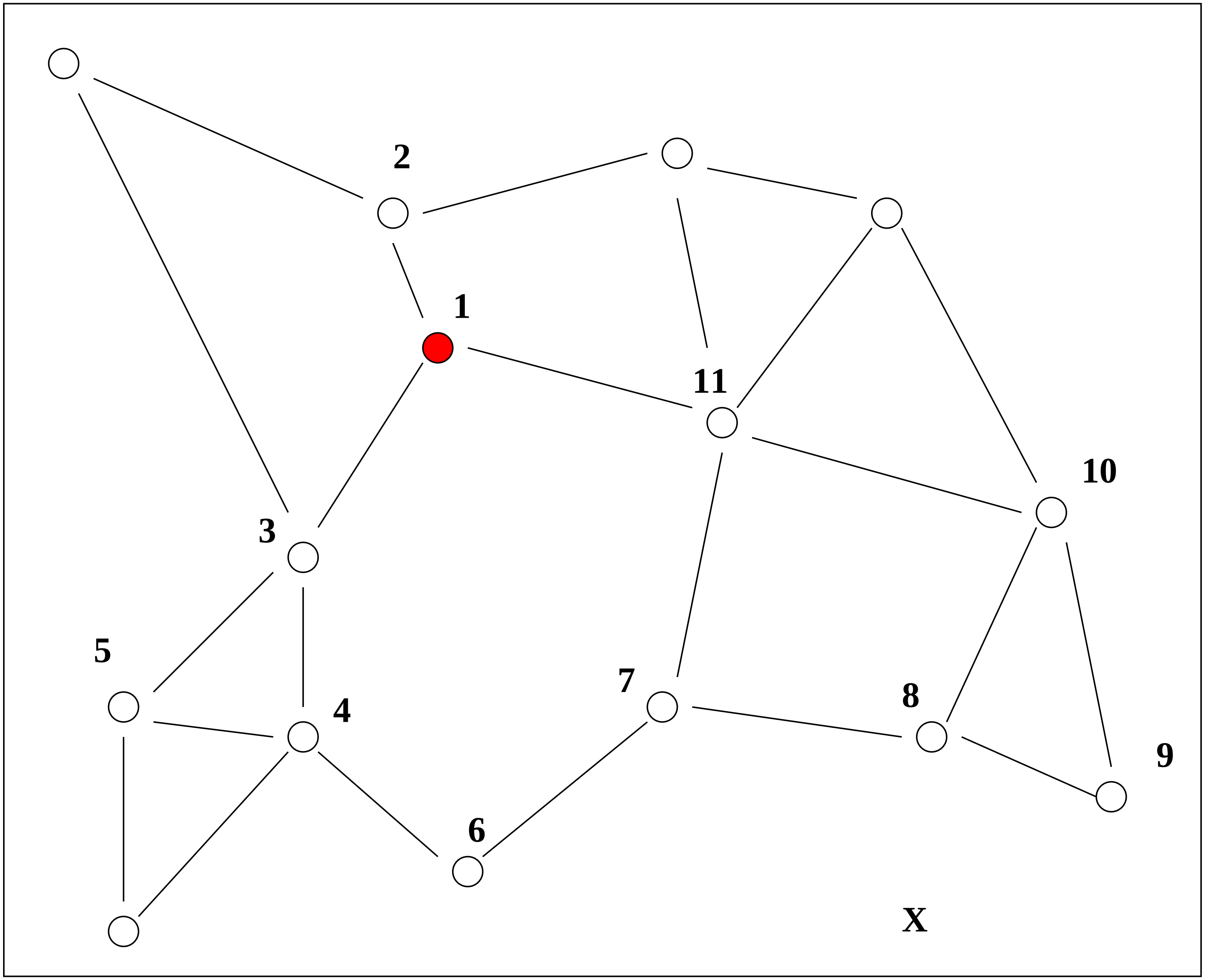
Roadmap algorithm

The problem:

- Tens of thousands of Gaussians– but frame probability may be dominated by two or three
- Algorithm needed to find the most important Gaussians per frame, without calculating them all

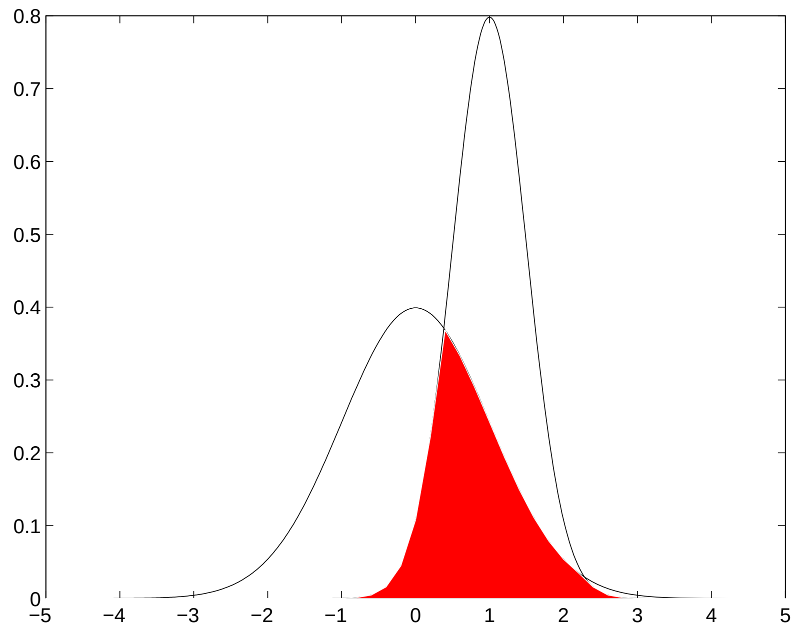
The solution:

- Make list of "near" Gaussians for each Gaussian in system $\rightarrow$ need distance measure
- Navigate between Gaussians to find best set for current input frame
- Like a roadmap...


2
1
11
10
3
5
7
8
4
9
6
X

Distance measure

Overlap for one dimension is defined as $\int_{-\infty}^{\infty}$ of the minimum of the two Gaussians:



- Suitable distance measure is $-\log(\mathrm{Overlap})$.
- Sum over all dimensions of diagonal-covariance Gaussians.

Constructing the roadmap

Use distance measure to construct a roadmap.

About 20 links to and from each Gaussian

1. - Make a list of most similar Gaussians
   - Start with random list, iteratively improve it
2. Make lists "symmetric"
3. Remove redundant links

Performance of Roadmap algorithm

Evaluating performance:

- If we miss some important Gaussians, the probability of the speech file given the model $\mathcal{N}$ will decrease
- Measure the average decrease in log probability per frame

	% Gaussians calculated	Loss in log probability
Roadmap	3.7%	0.004
VQ	4%	0.3

(For a system with 9,500 Gaussians)

FD Results on Resource Management

- 1,000 word task
- 109 speakers, 4 hours of training data
- decision-tree state-clustered cross-word triphones, 1577 states
- word-pair grammar
- 4 iterations of FD training done starting from ML-trained models

	Initial Word Error	Final Word Error	Decrease
1 Mixture	8.44	6.73	20.3 %
6 mixture	4.10	3.76	8.3 %

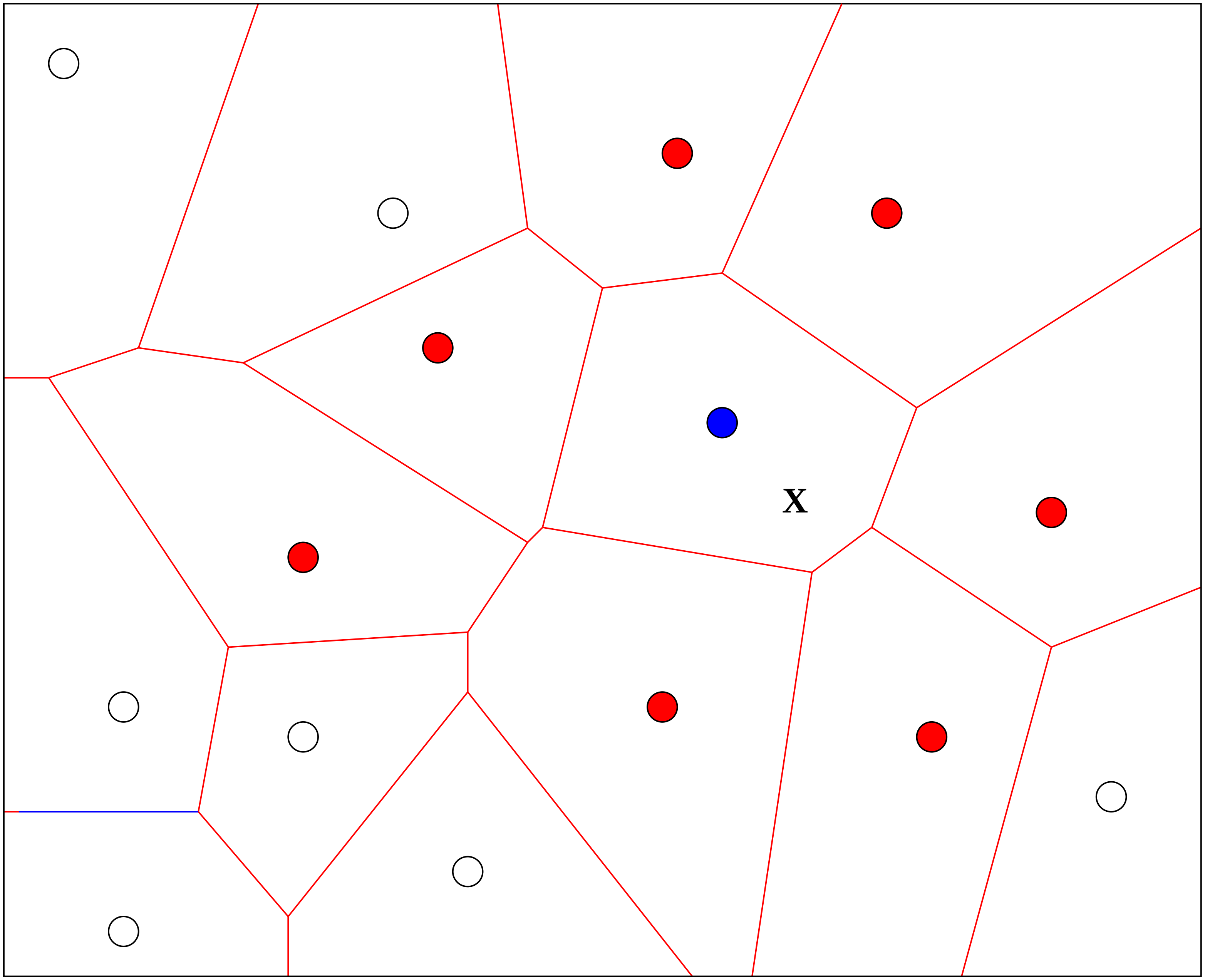
FD Results on Wall Street Journal

- 66 hours of training data (WSJ 0+1)
- decision-tree state-clustered cross-word triphones, 6399 states
- 65k word trigram LM

Num mix	% WER		% WER reduction	
Comps	MLE	FD	FD	MMIE
1	14.64	13.14	10.4	
2	12.52	11.30	9.7	8.6
4	10.96	10.30	6.0	
12	9.63	9.42	2.2	-0.9

Conclusions

- FD implemented on a large vocabulary task
- Used roadmap algorithm to greatly reduce computation
- Introduced overlap distance measure between Gaussians
- FD seems to work at least as well as MMI
- FD faster than lattice-based MMI: takes 6 times as long as ML training instead of 15 times as long

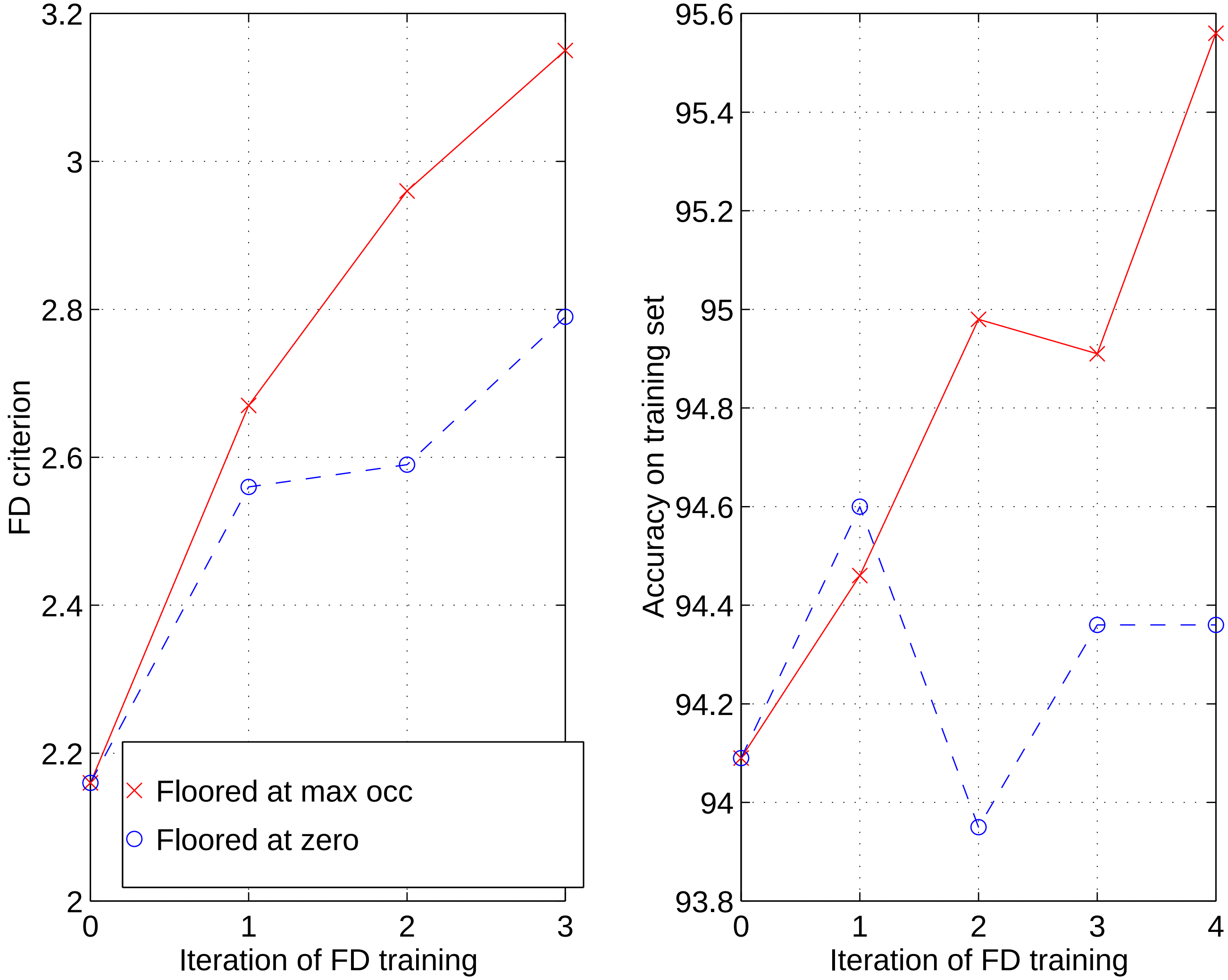

X

Smoothing in EBW equations

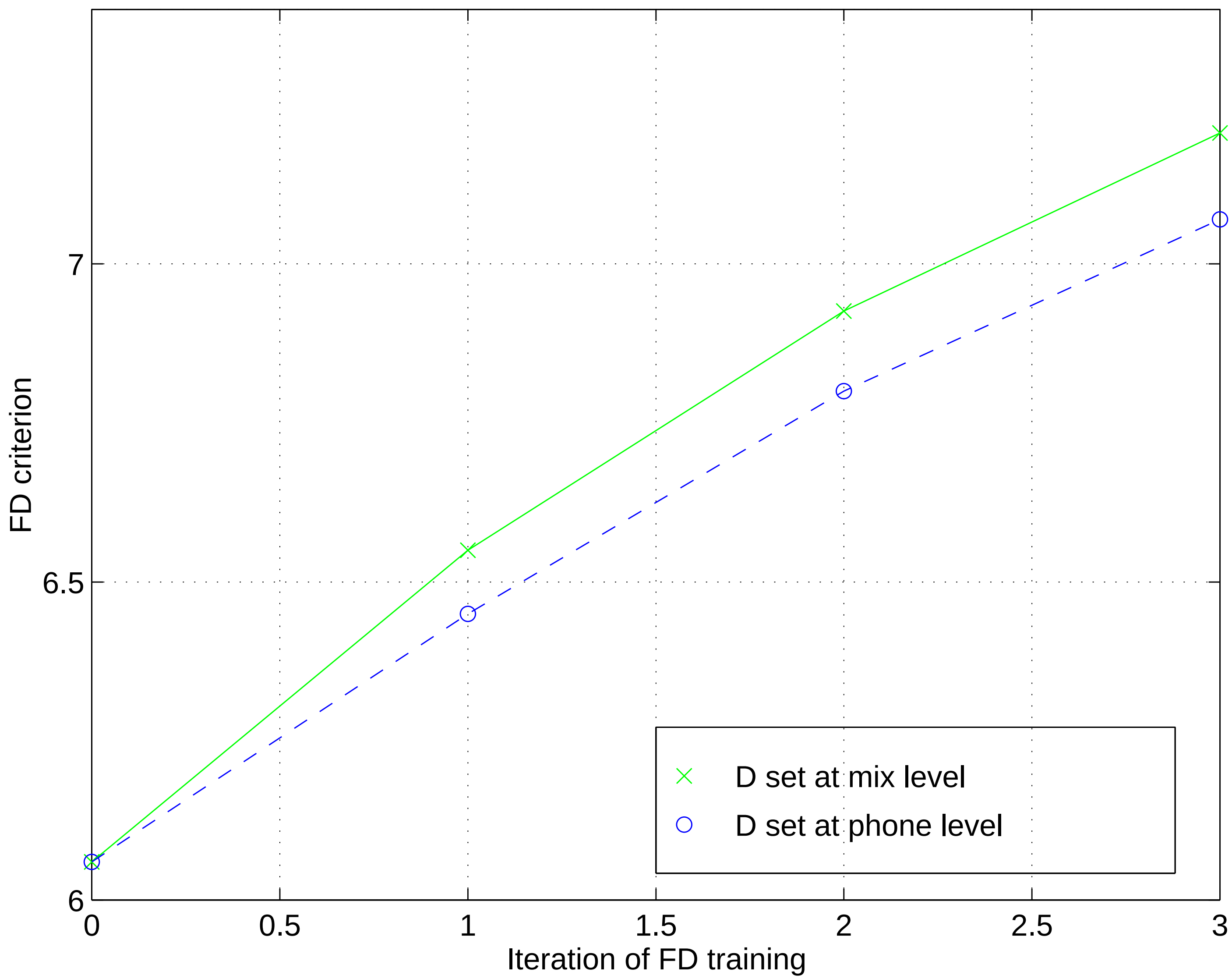
$$\hat{\mu}_{j,m} = \frac{\left\{\theta^{\text{num}}_{j,m}(\mathcal{O}) - \theta^{\text{den}}_{j,m}(\mathcal{O})\right\} + D\mu_{j,m}}{\left\{\gamma^{\text{num}}_{j,m} - \gamma^{\text{den}}_{j,m}\right\} + D},$$

$$\hat{\sigma}^2_{j,m} = \frac{\left\{\theta^{\text{num}}_{j,m}(\mathcal{O}^2) - \theta^{\text{den}}_{j,m}(\mathcal{O}^2)\right\} + D(\sigma^2_{j,m} + \mu^2_{j,m})}{\left\{\gamma^{\text{num}}_{j,m} - \gamma^{\text{den}}_{j,m}\right\} + D} - \hat{\mu}^2_{j,m},$$

- Set D at phone level
- Floor D at max in phone of any $\gamma^{\text{num}}_{j,m}$ or $\gamma^{\text{den}}_{j,m}$
- Improved both objective function and recognition results



FD criterion
3.2
3
2.8
2.6
2.4
2.2
2
0
1
2
3
Iteration of FD training
Floored at max occ
Floored at zero
Accuracy on training set
95.6
95.4
95.2
95
94.8
94.6
94.4
94.2
94
93.8
0
1
2
3
4
Iteration of FD training


FD criterion
Iteration of FD training
D set at mix level
D set at phone level
6
6.5
7
0
0.5
1
1.5
2
2.5
3