Automatic Speech Recognition Laboratory
Build Your Own Digit Recognizer

Izhak Shafran

The Center for Spoken Language Understanding
The Oregon Health & Science University
Portland, OR

June 19, 2009
Recall: The Fundamental Equation of Speech Recognition

\[
\hat{W} = \arg \max_W P(W|X; \Theta) \\
= \arg \max_W \frac{P(X|W; \Theta)P(W; \Theta)}{P(X)} \quad \text{Bayes’ Rule} \\
= \arg \max_W P(X|W; \Theta)P(W; \Theta) \quad P(X) \text{ irrelevant.}
\]

Where \( W \) is a word sequence, \( \hat{W} \) is the best word sequence, \( X \) is a sequence of acoustic features, and \( \Theta \) denotes the model parameters.
Implementing the fundamental equation

\[
\hat{W} = \arg \max_W P(X|W; \Theta) P(W; \Theta)
\]
Implementing the fundamental equation

$$\hat{W} = \arg\max_W P(X|W; \Theta) P(W; \Theta)$$
Implementing the fundamental equation

\[ \hat{W} = \arg \max_W P(X|W; \Theta)P(W; \Theta) \]
Implementing the fundamental equation

\[
\hat{W} = \arg\max_W P(X|W; \Theta)P(W; \Theta)
\]
Implementing the fundamental equation

\[
\hat{W} = \operatorname{arg\,max}_W P(X|W; \Theta)P(W; \Theta)
\]
Goals for this Lab

- Build your own digit recognizer
- Optimize performance
  - Feature extraction
  - Acoustic units
  - Grammar
- Live demo
Experimental Setup

- Corpus: OGI Numbers (subset)
- Training set: 1 hr, 2444 utterances (speakers)
- Test set: 1659 utterances (speakers)
- Toolkit: The HMM ToolKit (HTK)

```bash
scp -r login.clsp.jhu.edu:~zak/asrlab/htk ~/asrlab/
export PATH=~/asrlab/htk/bin:$PATH
open ~/asrlab/htk/HTKBook/htkbook.pdf
http://www.csee.ogi.edu/~zak/htkbook.pdf
```
Implementing the fundamental equation

\[ \hat{W} = \arg \max_{W} P(X|W; \Theta)P(W; \Theta) \]
Acoustic Model Training: Inputs

- Baseline system
  
  ```bash
  scp -r /home/zak/asrlab/expt0 ~/asrlab/
  ```

- Speech waveforms
  
  ```bash
  /home/zak/asrlab/speech
  ```

- Transcripts
  
  ```bash
  expt0/setup/{train|test}*mlf
  ```
Estimating the HMMs

- Popular Criterion: Maximize likelihood

\[ \hat{\theta} = \arg \max_{\theta} P(X|W, \theta) \]
Estimating the HMMs

- Popular Criterion: Maximize likelihood

\[ \hat{\theta} = \arg\max_{\theta} P(X|W, \theta) \]

- Approach: EM estimation (iterative)

\[ \hat{\theta}_{n+1} = \arg\max_{\theta} \log P(X|W, \theta_n) P(X|W, \theta) \]
Estimating the HMMs

- Popular Criterion: Maximize likelihood
  \[ \hat{\theta} = \operatorname{arg\ max}_\theta P(X|W, \theta) \]

- Approach: EM estimation (iterative)
  \[ \hat{\theta}_{n+1} = \operatorname{arg\ max}_\theta \log P(X|W, \theta_n) P(X|W, \theta) \]

- Problem: Local maxima
Estimating the HMMs

- Popular Criterion: Maximize likelihood

\[ \hat{\theta} = \arg \max_\theta P(X|W, \theta) \]

- Approach: EM estimation (iterative)

\[ \hat{\theta}_{n+1} = \arg \max_\theta \log P(X|W, \theta_n)P(X|W, \theta) \]

- Problem: Local maxima

- Solution: Increase complexity gradually
Training: Increasing Complexity Gradually
scripts/mlTrain.sh

- Initialization

  - Estimation of global mean and variance
    scripts/calcGlobalStats.sh, setup/hmmproto.gz, setup/mfcc.cfg
    Output: expt0/train/Global/vFloors

  - Initialization of an HMM for each acoustic unit (topology?)
    scripts/clonehmm.pl, setup/hmmtop
    Output: expt0/train/Global/MMF

Building single Gaussian models

  - EM iterations without inter-word silences (iterations?)
    scripts/EMTrain.sh, setup/train.no-iwsil.mlf
    Output: train/CI-NOSIL

  - EM iterations with inter-word silences (iterations?)
    scripts/EMTrain.sh, setup/train.ci.mlf
    Output: train/CI-IWSIL
Training: Increasing Complexity Gradually

scripts/mlTrain.sh

- **Initialization**
  - Estimate global mean and variance
    scripts/calcGlobalStats.sh, setup/hmmproto.gz, setup/mfcc.cfg
    Output: expt0/train/Global/vFloors

- Initialize an HMM for each acoustic unit (topology?)
  scripts/clonehmm.pl, setup/hmmtop
  Output: expt0/train/Global/MMF

- Building single Gaussian models
  - EM iterations without inter-word silences (iterations?)
    scripts/EMTrain.sh, setup/train.no-iwsil.mlf
    Output: train/CI-NOSIL
  - EM iterations with inter-word silences (iterations?)
    scripts/EMTrain.sh, setup/train.ci.mlf
    Output: train/CI-IWSIL
Training: Increasing Complexity Gradually

scripts/mlTrain.sh

- **Initialization**
  - Estimate global mean and variance
    - scripts/calcGlobalStats.sh, setup/hmmproto.gz, setup/mfcc.cfg
    - Output: expt0/train/Global/vFloors
  - Initialize an HMM for each acoustic unit (topology?)
    - scripts/clonehmm.pl, setup/hmmtop
    - Output: expt0/train/Global/MMF
Training: Increasing Complexity Gradually

scripts/mlTrain.sh

- **Initialization**
  - Estimate global mean and variance
    scripts/calcGlobalStats.sh, setup/hmmproto.gz, setup/mfcc.cfg
    Output: expt0/train/Global/vFloors
  - Initialize an HMM for each acoustic unit (topology?)
    scripts/clonehmm.pl, setup/hmmtop
    Output: expt0/train/Global/MMF

- **Building single Gaussian models**
Training: Increasing Complexity Gradually

scripts/mlTrain.sh

- **Initialization**
  - Estimate global mean and variance
    scripts/calcGlobalStats.sh, setup/hmmproto.gz, setup/mfcc.cfg
    Output: expt0/train/Global/vFloors
  - Initialize an HMM for each acoustic unit (topology?)
    scripts/clonehmm.pl, setup/hmmtop
    Output: expt0/train/Global/MMF

- **Building single Gaussian models**
  - EM iterations without inter-word silences (iterations?)
    scripts/EMTrain.sh, setup/train.no-iwsil.mlf
    Output: train/CI-NOSIL
Training: Increasing Complexity Gradually

scripts/mlTrain.sh

● Initialization
  ▶ Estimate global mean and variance
    scripts/calcGlobalStats.sh, setup/hmmproto.gz, setup/mfcc.cfg
    Output: expt0/train/Global/vFloors
  ▶ Initialize an HMM for each acoustic unit (topology?)
    scripts/clonehmm.pl, setup/hmmtop
    Output: expt0/train/Global/MMF

● Building single Gaussian models
  ▶ EM iterations without inter-word silences (iterations?)
    scripts/EMTrain.sh, setup/train.no-iwsil.mlf
    Output: train/CI-NOSIL
  ▶ EM iterations with inter-word silences (iterations?)
    scripts/EMTrain.sh, setup/train.ci.mlf
    Output: train/CI-IWSIL
Single Gaussian to Gaussian Mixtures

- Updating transcripts (forced alignment)
  - scripts/ViterbiAlign.sh, setup/word.mlf
  - Output: train/CI-IWSIL/hmm4/align.mlf

- Building Gaussian mixture models
  - Mixture splitting (schedule?)
    - scripts/MixUp.sh
  - EM iterations (iterations?)
    - scripts/EMTrain.sh,
      - train/CI-IWSIL/hmm4/align.mlf
Decoding Test Set

- Grammar
  - HParse
    - Output: setup/wdnet

- Decoding the test set with your models (pruning?)
  - setup/test.list, setup/wdnet
  - Output: test/hv.mlf
Measuring Performance: Word Error Rate

1. Align reference and hypothesized word strings.
2. \[ WER(\%) = 100 \frac{N_{sub} + N_{ins} + N_{del}}{N_{ref}} \]

HResult, test/result
Optimizing Performance

Now, let’s improve up on the baseline 20% WER

- Try different features: MFCC, PLP, MFCC_D_A, PLP_D_A
- Try different topology
- Try phone-based model, instead of digit based models
- Try different decoding grammars

You should easily be able to reduce WER to 5%!
Live Demo

Now, try your models on your own speech, even though it has some mismatch with the training data.

- Make sure you sample at 8KHz
- Make sure you record as single channel

Enjoy!