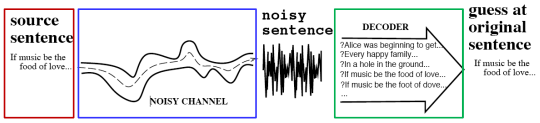
Statistical NLP
Spring 2010

Lecture 9: Acoustic Models

Dan Klein – UC Berkeley

The Noisy Channel Model

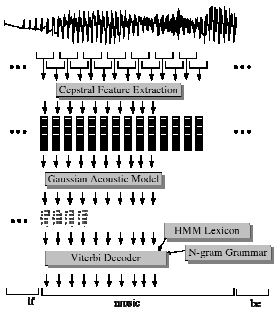


$$w^* = \arg\max_w P(w|a)$$
$$\propto \arg\max_w P(a|w)P(w)$$

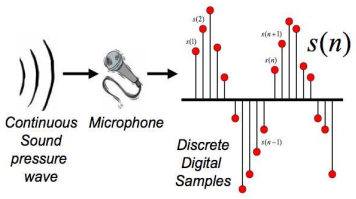
Acoustic model: HMMs over word positions with mixtures of Gaussians as emissions

Language model: Distributions over sequences of words (sentences)

Speech Recognition Architecture



Digitizing Speech



Thanks to Bryan Pellom for this slide!

Frame Extraction

- A frame (25 ms wide) extracted every 10 ms

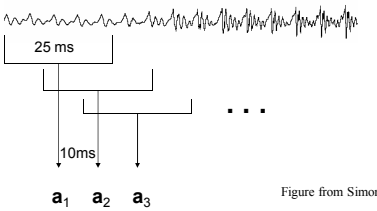
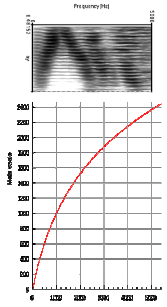


Figure from Simon Arnfield

Mel Freq. Cepstral Coefficients

- Do FFT to get spectral information
  - Like the spectrogram/spectrum we saw earlier
- Apply Mel scaling
  - Models human ear; more sensitivity in lower freqs
  - Approx linear below 1kHz, log above, equal samples above and below 1kHz
- Plus discrete cosine transform



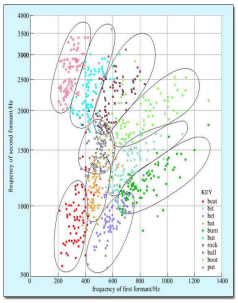
[Graph from Wikipedia]

Final Feature Vector

- 39 (real) features per 10 ms frame:
  - 12 MFCC features
  - 12 delta MFCC features
  - 12 delta-delta MFCC features
  - 1 (log) frame energy
  - 1 delta (log) frame energy
  - 1 delta-delta (log frame energy)
- So each frame is represented by a 39D vector

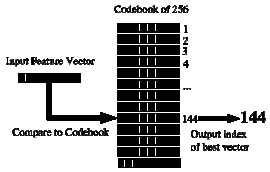
HMMs for Continuous Observations

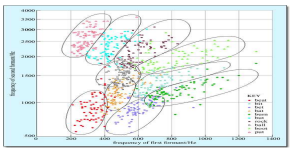
- Before: discrete set of observations
- Now: feature vectors are real-valued
- Solution 1: discretization
- Solution 2: continuous emissions
  - Gaussians
  - Multivariate Gaussians
  - Mixtures of multivariate Gaussians
- A state is progressively
  - Context independent subphone (~3 per phone)
  - Context dependent phone (triphones)
  - State tying of CD phone



Vector Quantization

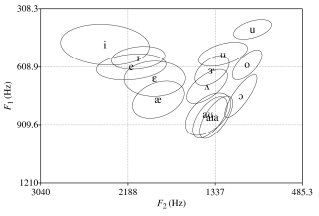
- Idea: discretization
  - Map MFCC vectors onto discrete symbols
  - Compute probabilities just by counting
- This is called vector quantization or VQ
- Not used for ASR any more; too simple
- But: useful to consider as a starting point





Gaussian Emissions

- VQ is insufficient for real ASR
  - Hard to cover high-dimensional space with codebook
  - What about our perception of low-dimensional phonemes?
- Instead: assume the possible values of the observation vectors are normally distributed.
  - Represent the observation likelihood function as a Gaussian?



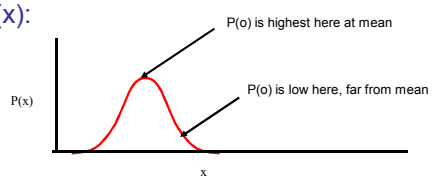
From bartus.org/akustyk

Gaussians for Acoustic Modeling

A Gaussian is parameterized by a mean and a variance:

$$P(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- P(x):



Multivariate Gaussians

- Instead of a single mean μ and variance σ^2:

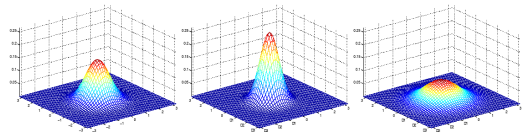
$$P(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- Vector of means μ and covariance matrix Σ

$$P(x|\mu,\Sigma) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

- Usually assume diagonal covariance (!)
  - This isn't very true for FFT features, but is often OK for MFCC features

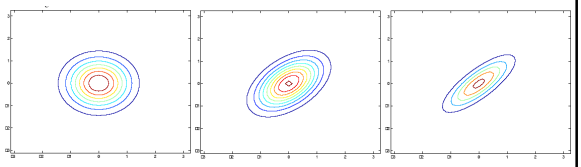
Gaussians: Size of Σ



- μ = [0 0] μ = [0 0] μ = [0 0]
- Σ = I Σ = 0.6I Σ = 2I
- As Σ becomes larger, Gaussian becomes more spread out; as Σ becomes smaller, Gaussian more compressed

Text and figures from Andrew Ng

Gaussians: Shape of Σ



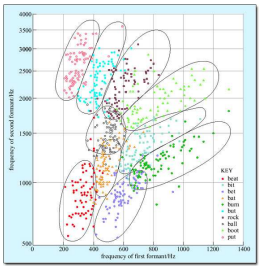
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \quad .\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

- As we increase the off diagonal entries, more correlation between value of x and value of y

Text and figures from Andrew Ng

But we're not there yet

- Single Gaussians may do a bad job of modeling a complex distribution in any dimension
- Even worse for diagonal covariances
- Solution: mixtures of Gaussians



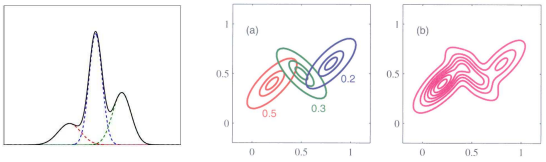
From openlearn.open.ac.uk

Mixtures of Gaussians

- M mixtures of Gaussians:

$$P(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{k/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i)\right)$$

$$P(x|\mu, \mathbf{\Sigma}, \mathbf{c}) = \sum_i c_i P(x|\mu_i, \Sigma_i)$$

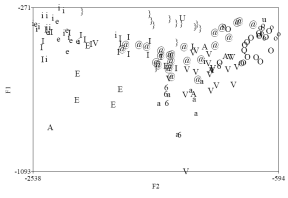


From robots.ox.ac.uk

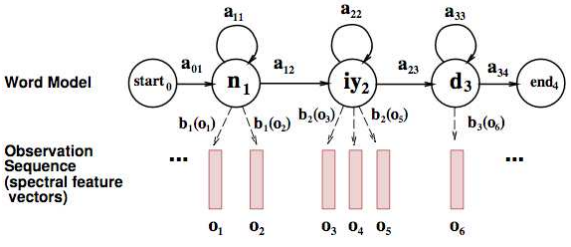
http://www.itee.uq.edu.au/~comp4702

GMMs

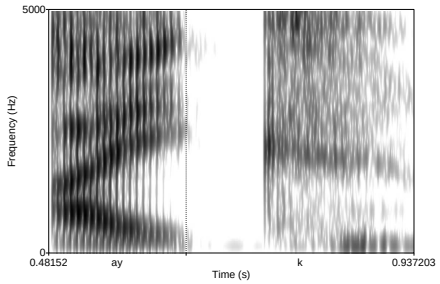
- Summary: each state has an emission distribution P(x|s) (likelihood function) parameterized by:
  - M mixture weights
  - M mean vectors of dimensionality D
  - Either M covariance matrices of DxD or M Dx1 diagonal variance vectors



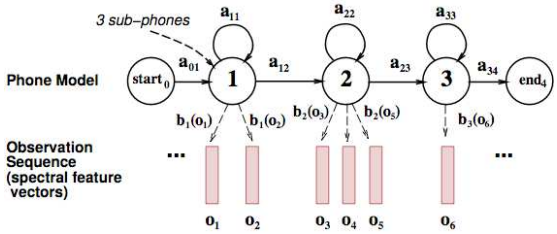
HMMs for Speech



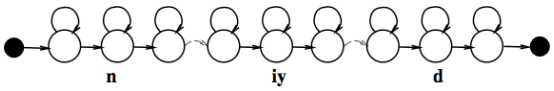
Phones Aren't Homogeneous



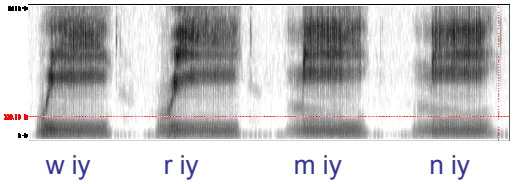
Need to Use Subphones



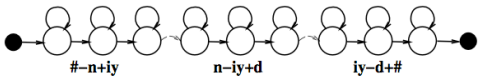
A Word with Subphones



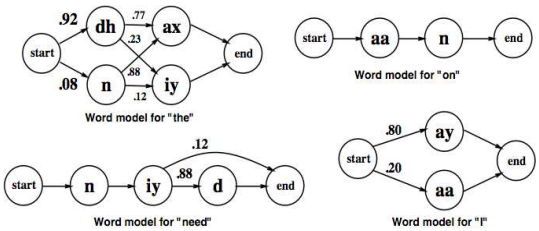
Modeling phonetic context



"Need" with triphone models



ASR Lexicon: Markov Models



Markov Process with Bigrams

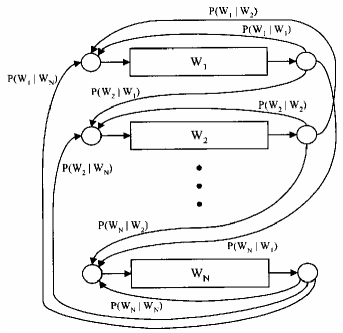


Figure from Huang et al page 618

Training Mixture Models

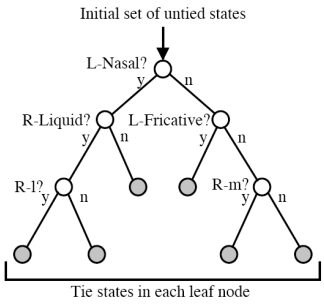
- Input: wav files with unaligned transcriptions
- Forced alignment
  - Computing the "Viterbi path" over the training data (where the transcription is known) is called "forced alignment"
  - We know which word string to assign to each observation sequence.
  - We just don't know the state sequence.
  - So we constrain the path to go through the correct words (by using a special example-specific language model)
  - And otherwise run the Viterbi algorithm
- Result: aligned state sequence

Lots of Triphones

- Possible triphones: 50x50x50=125,000
- How many triphone types actually occur?
- 20K word WSJ Task (from Bryan Pellom)
  - Word internal models: need 14,300 triphones
  - Cross word models: need 54,400 triphones
- Need to generalize models, tie triphones

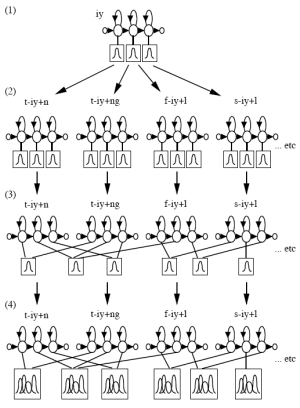
State Tying / Clustering

- [Young, Odell, Woodland 1994]
- How do we decide which triphones to cluster together?
- Use phonetic features (or 'broad phonetic classes')
  - Stop
  - Nasal
  - Fricative
  - Sibilant
  - Vowel
  - lateral

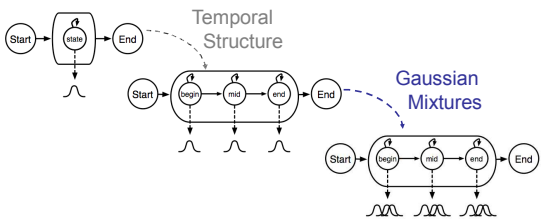


State Tying

- Creating CD phones:
  - Start with monophone, do EM training
  - Clone Gaussians into triphones
  - Build decision tree and cluster Gaussians
  - Clone and train mixtures (GMMs)
- General idea:
  - Introduce complexity gradually
  - Interleave constraint with flexibility

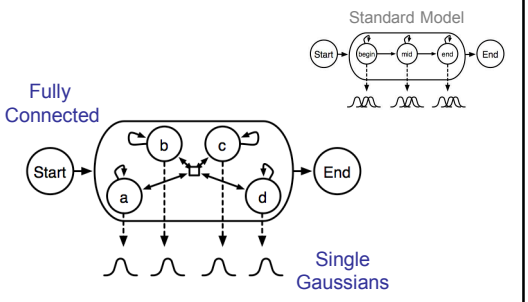


Standard subphone/mixture HMM



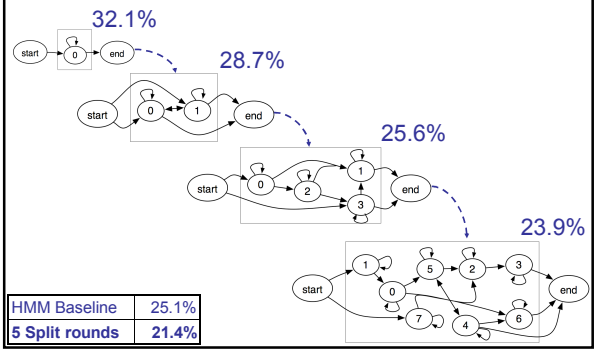
Model	Error rate
HMM Baseline	**25.1%**

An Induced Model



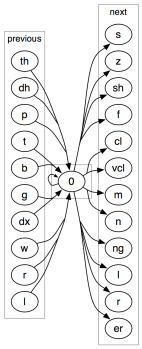
[Petrov, Pauls, and Klein, 07]

Hierarchical Split Training with EM

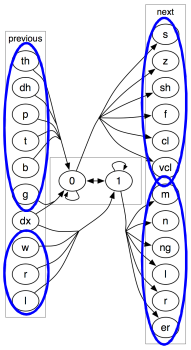


HMM Baseline	25.1%
5 Split rounds	**21.4%**

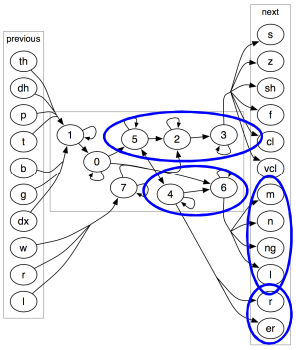
Refinement of the /ih/-phone



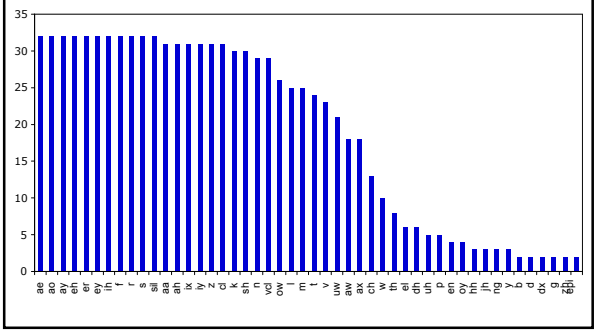
Refinement of the /ih/-phone



Refinement of the /ih/-phone



HMM states per phone



Inference

- State sequence:
  d_1-d_6-d_6-d_4-ae_5-ae_2-ae_3-ae_0-d_2-d_2-d_3-d_7-d_5 **Viterbi**
- Phone sequence:
  d - d - d -d -ae - ae - ae - ae - d - d -d - d - d **Variational**
- Transcription
  d - ae - d **???**