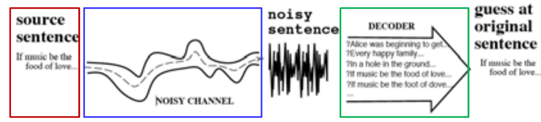
Natural Language Processing

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)

Figure: J & M

Speech Recognition Architecture

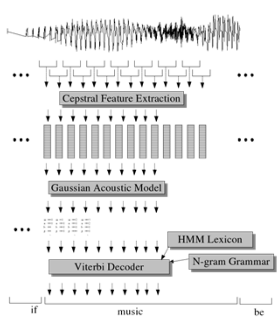


Figure: J & M

Feature Extraction

Digitizing Speech

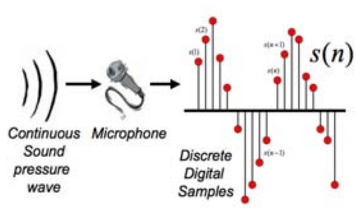


Figure: Bryan Pellom

Frame Extraction

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

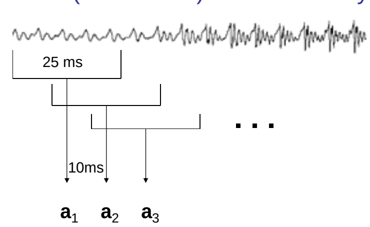
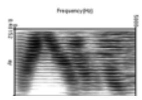
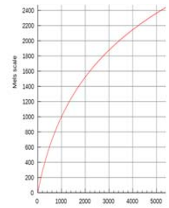


Figure: Simon Arnfield

Mel Freq. Cepstral Coefficients

- Do FFT to get spectral information
  - Like the spectrogram 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: Wikipedia]

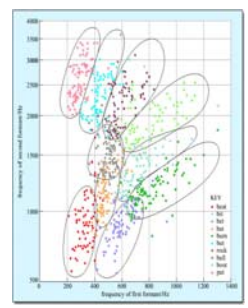
Final Feature Vector

- 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

Emission Model

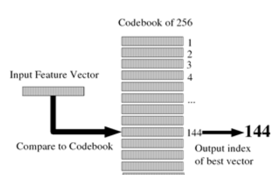
HMMs for Continuous Observations

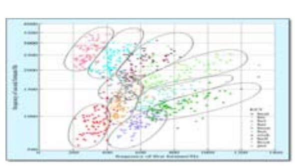
- Before: discrete set of observations
- Now: feature vectors are real-valued
- Solution 1: discretization
- Solution 2: continuous emissions
  - Gaussians
  - Multivariate Gaussians
  - Mixtures of multivariate Gaussians
- A state is progressively
  - Context independent subphone (~3 per phone)
  -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



Vector Quantization

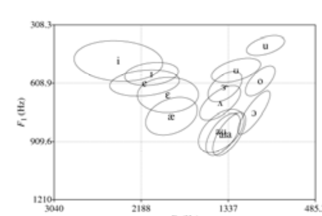
- Idea: discretization
  - Map MFCC vectors onto discrete symbols
  - Compute probabilities just by counting
- This is called vector quantization or VQ
- Not used for ASR any more
- But: useful to consider as a starting point





Gaussian Emissions

- VQ is insufficient for top-quality ASR
  - Hard to cover high-dimensional space with codebook
  - Moves ambiguity from the model to the preprocessing
- Instead: assume the possible values of the observation vectors are normally distributed.
  - Represent the observation likelihood function as a Gaussian?



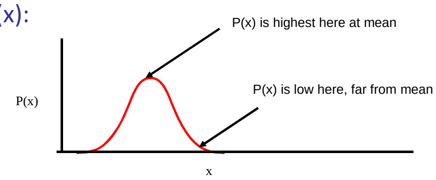
From bartus.org/akustyk

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:

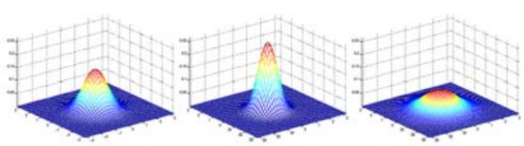
$$P(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- 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 less bad for MFCC features

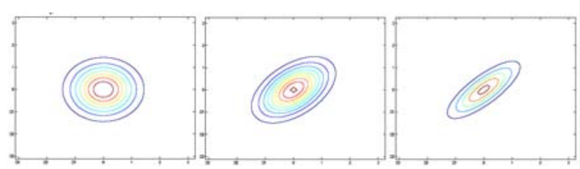
Gaussians: Size of Σ



- 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



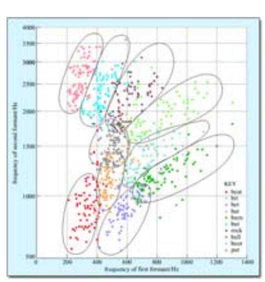
$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \quad .\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

- 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



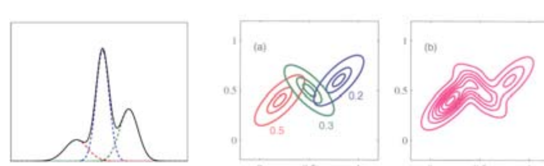
From openlearn.open.ac.uk

Mixtures of Gaussians

- Mixtures of Gaussians:

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

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

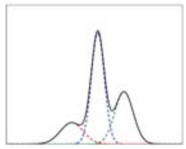


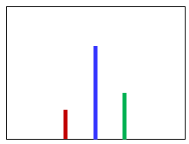
From robots.ox.ac.uk

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
- Like soft vector quantization after all
  - Think of the mixture means as being learned codebook entries
  - Think of the Gaussian densities as a learned codebook distance function
  - Think of the mixture of Gaussians like a multinomial over codes
  - (Even more true given shared Gaussian inventories, cf next week)

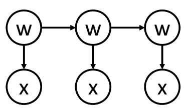




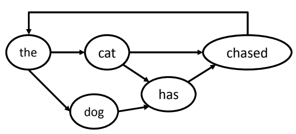
State Model

State Transition Diagrams

- Bayes Net: HMM as a Graphical Model



- State Transition Diagram: Markov Model as a Weighted FSA



ASR Lexicon

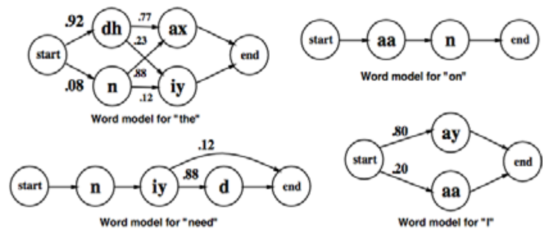


Figure: J & M

Lexical State Structure

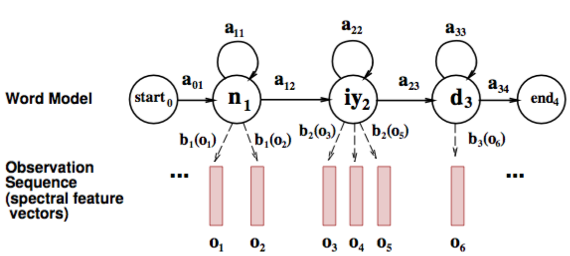


Figure: J & M

Adding an LM

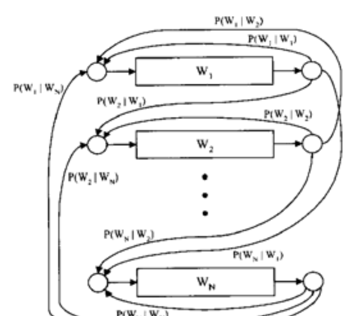


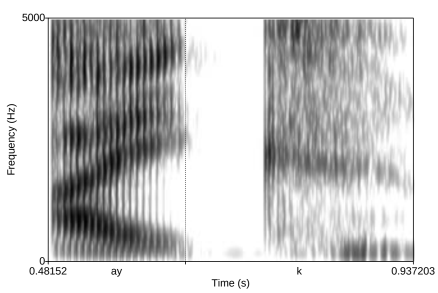
Figure from Huang et al page 618

State Space

- State space must include
  - Current word (|V| on order of 20K+)
  - Index within current word (|L| on order of 5)
- Acoustic probabilities only depend on phone type
  - E.g. P(x|lec[t]ure) = P(x|t)
- From a state sequence, can read a word sequence

State Refinement

Phones Aren't Homogeneous



Need to Use Subphones

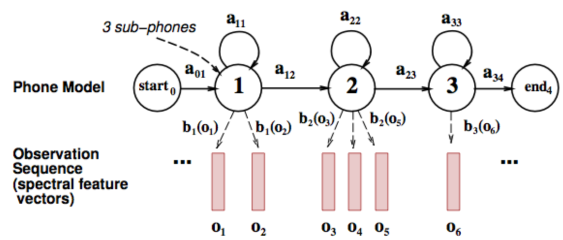


Figure: J & M

A Word with Subphones

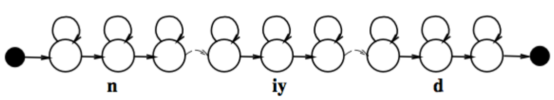
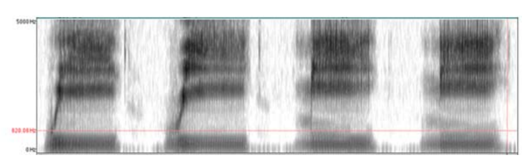


Figure: J & M

Modeling phonetic context



w iy r iy m iy n iy

“Need” with triphone models

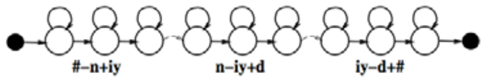


Figure: J & M

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

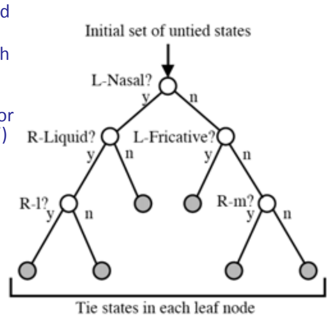


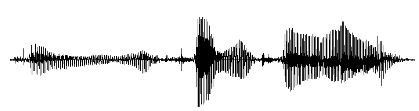
Figure: J & M

State Space

- State space now includes
  - Current word: |W| is order 20K
  - Index in current word: |L| is order 5
  - Subphone position: 3
- Acoustic model depends on clustered phone context
  - But this doesn’t grow the state space

Decoding

Inference Tasks



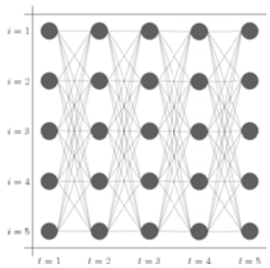
Most likely word sequence:

d - ae - d

Most likely 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 Decoding



$$\phi_t(s_t, s_{t-1}) = P(x_t|s_t)P(s_t|s_{t-1})$$

$$v_t(s_t) = \max_{s_{t-1}} \phi_t(s_t, s_{t-1})v_{t-1}(s_{t-1})$$

Figure: Enrique Benimeli

Viterbi Decoding

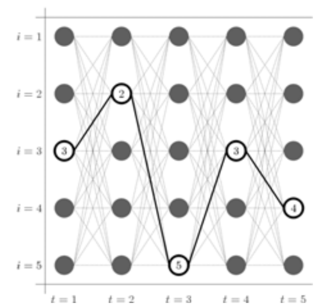
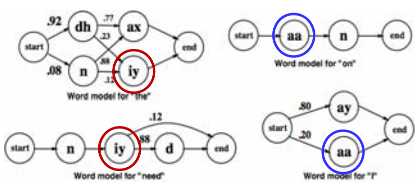


Figure: Enrique Benimeli

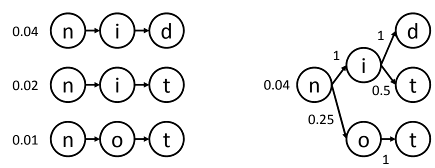
Emission Caching

- Problem: scoring all the P(x|s) values is too slow
- Idea: many states share tied emission models, so cache them



Prefix Trie Encodings

- Problem: many partial-word states are indistinguishable
- Solution: encode word production as a prefix trie (with pushed weights)

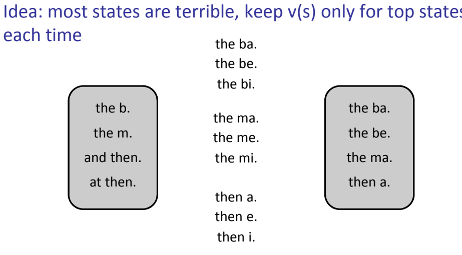


- A specific instance of minimizing weighted FSAs [Mohri, 94]

Figure: Aubert, 02

Beam Search

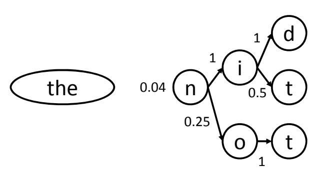
- Problem: trellis is too big to compute v(s) vectors
- Idea: most states are terrible, keep v(s) only for top states at each time



- Important: still dynamic programming; collapse equiv states

LM Factoring

- Problem: Higher-order n-grams explode the state space
- (One) Solution:
  - Factor state space into (word index, lm history)
  - Score unigram prefix costs while inside a word
  - Subtract unigram cost and add trigram cost once word is complete



LM Reweighting

- Noisy channel suggests

$$P(x|w)P(w)$$

- In practice, want to boost LM

$$P(x|w)P(w)^{\alpha}$$

- Also, good to have a "word bonus" to offset LM costs

$$P(x|w)P(w)^{\alpha}|w|^{\beta}$$

- These are both consequences of broken independence assumptions in the model