Algorithms for NLP

Acoustic Models, HMM

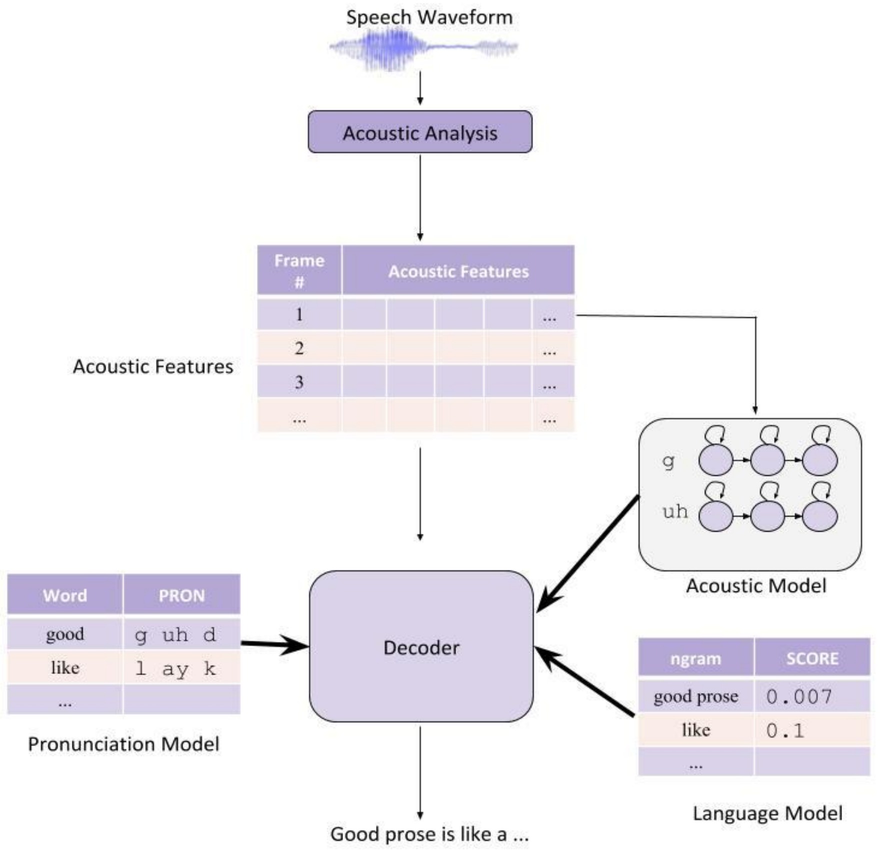
Yulia Tsvetkov – CMU

Slides: Taylor Berg-Kirkpatrick – CMU/UCSD
Dan Klein – UC Berkeley

HW grading

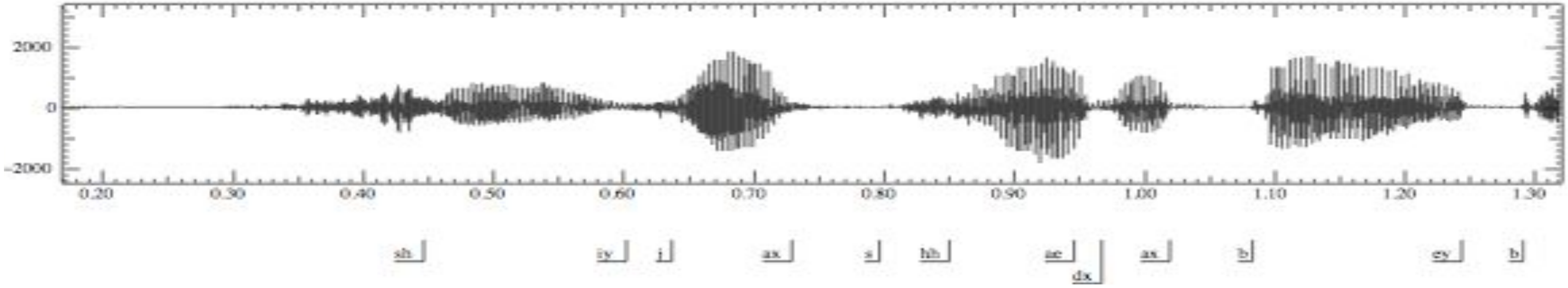
- 9 points is sufficient to get an A and additional points are for A+

Acoustic Modeling



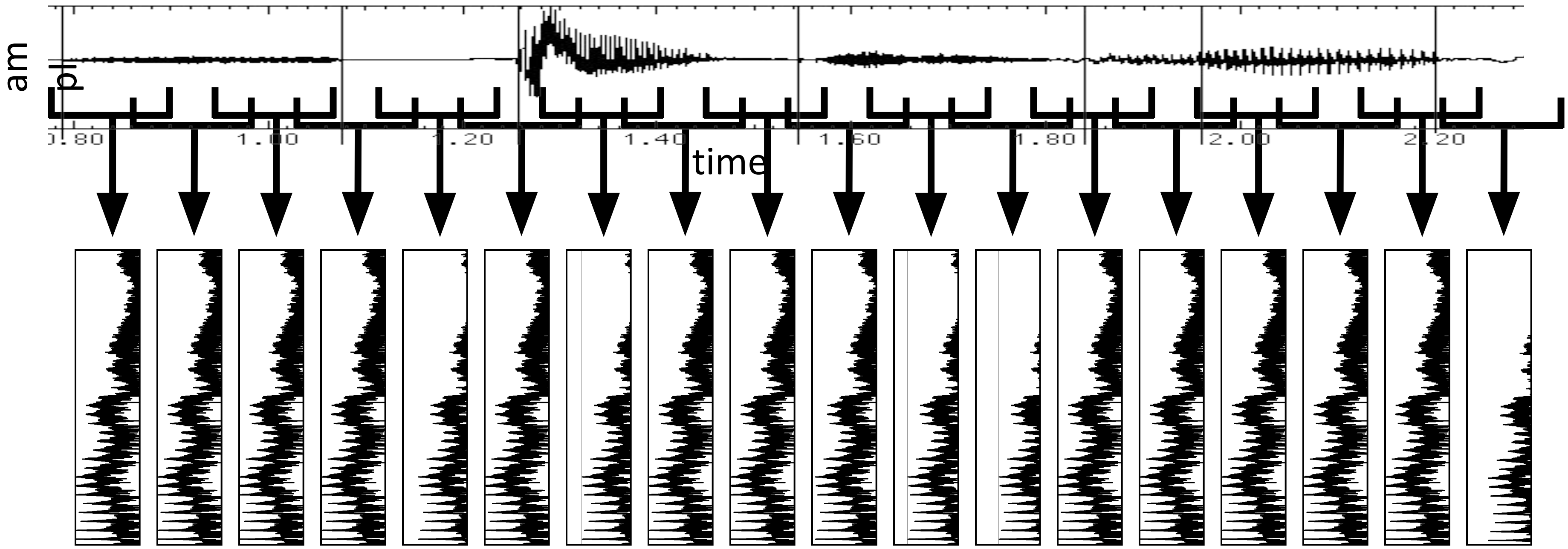
Slide by Preethi Jyothi

“She just had a baby”

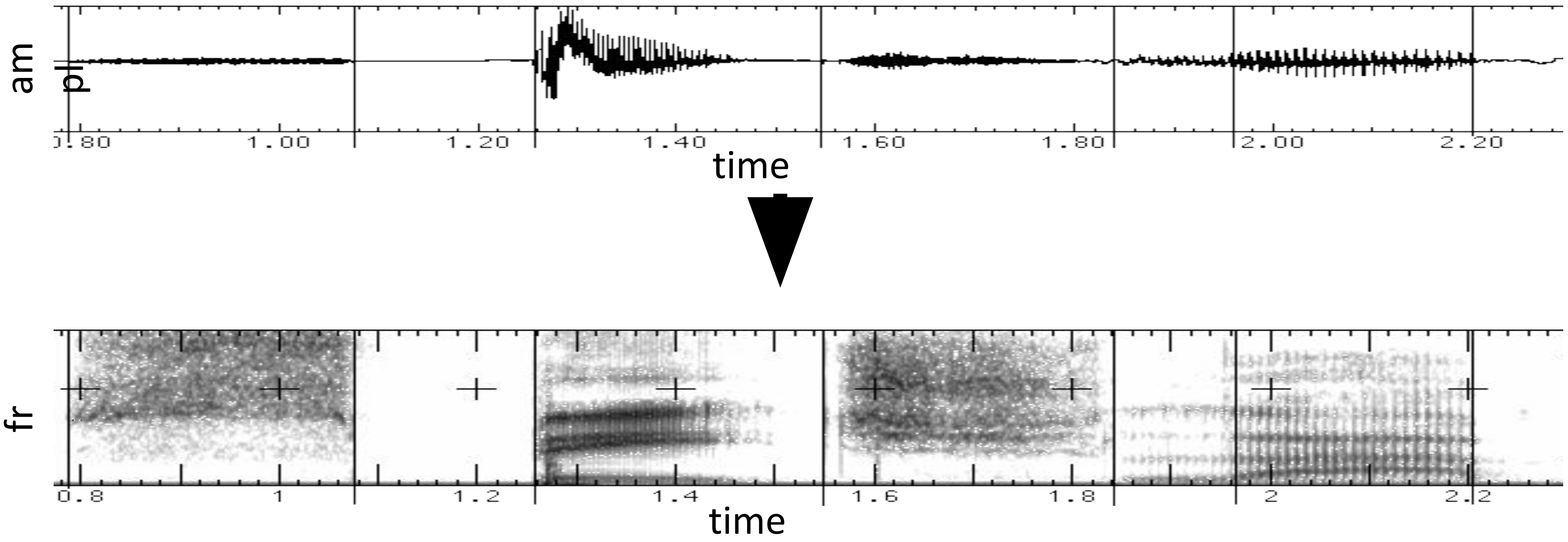


- What can we learn from a wavefile?
  - No gaps between words (!)
  - Vowels are voiced, long, loud
  - Voicing: regular peaks in amplitude
  - When stops closed: no peaks, silence
  - Peaks = voicing: .46 to .58 (vowel [iy], from second .65 to .74 (vowel [ax]) and so on
  - Silence of stop closure (1.06 to 1.08 for first [b], or 1.26 to 1.28 for second [b])
  - Fricatives like [sh]: intense irregular pattern; see .33 to .46

Spectrograms



Spectrograms



Places of Articulation

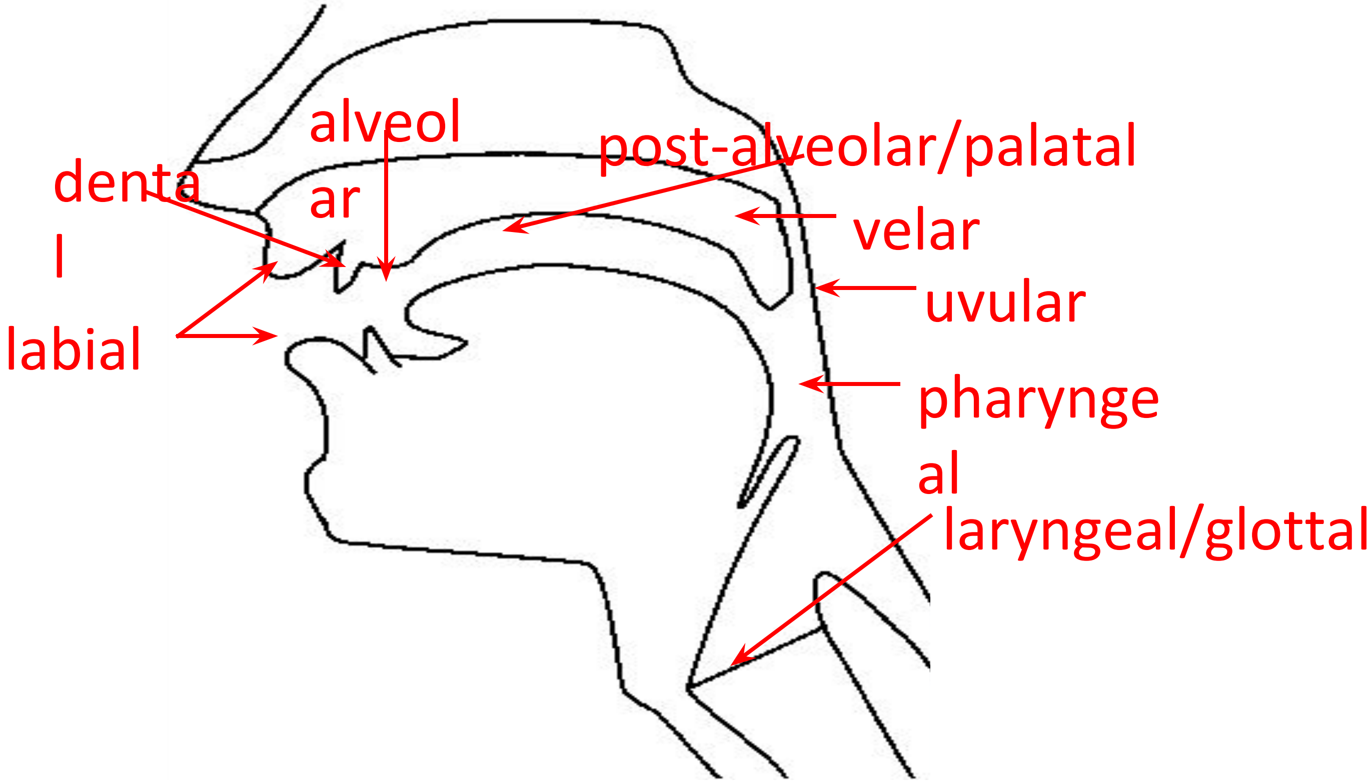


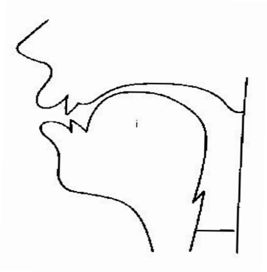
Figure thanks to Jennifer Venditti

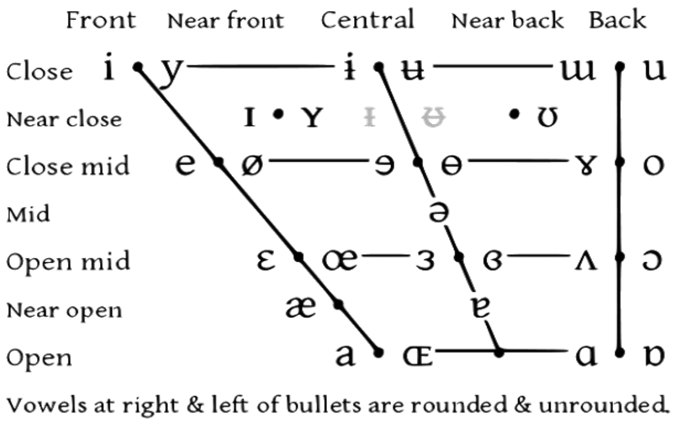
Space of Phonemes

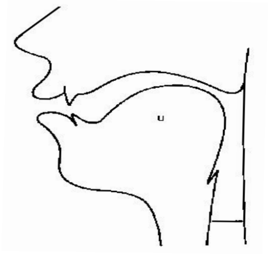
	LABIAL		CORONAL				DORSAL			RADICAL		LARYNGEAL
	Bilabial	Labio-dental	Dental	Alveolar	Palato-alveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Epi-glottal	Glottal
Nasal	m	ɱ	n			ɳ	ɲ	ŋ	ɴ			
Plosive	p b	ȹ ȸ	t d			ʈ ɖ	c ɟ	k ɡ	q ɢ		ʡ	ʔ
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	ʜ ʢ	h ɦ
Approximant		ʋ	ɹ			ɻ	j	ɰ				
Trill	ʙ		r						ʀ		я	
Tap, Flap		ⱱ	ɾ			ɽ						
Lateral fricative			ɬ ɮ			ꞎ	𝼆	𝼄				
Lateral approximant			l			ɭ	ʎ	ʟ				
Lateral flap			ɺ			𝼈						

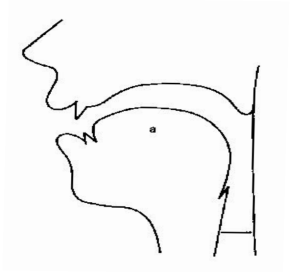
- Standard international phonetic alphabet (IPA) chart of consonants

Vowel Space

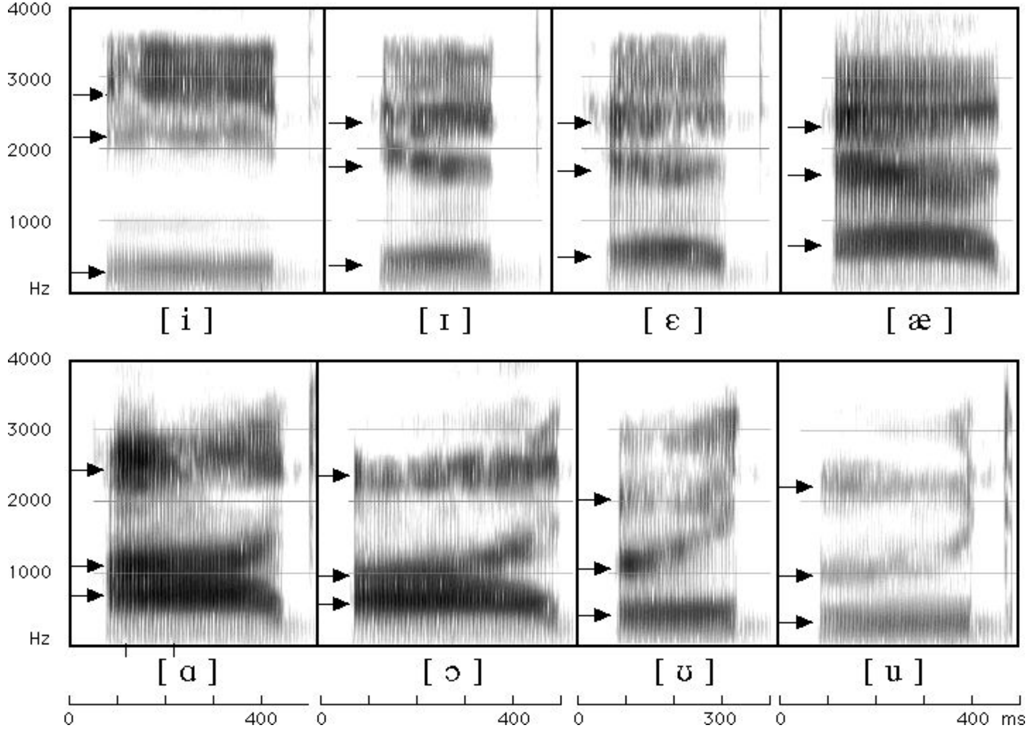



Front Near front Central Near back Back
Close i • y —— ɨ • ʉ —— ɯ • u
Near close ɪ • ʏ ᵻ ᵿ • ʊ
Close mid e • ø —— ɘ • ɵ —— ɤ • o
Mid ə
Open mid ɛ • œ —— ɜ • ɞ —— ʌ • ɔ
Near open æ ɐ
Open a • ɶ —— ɑ • ɒ
Vowels at right & left of bullets are rounded & unrounded.

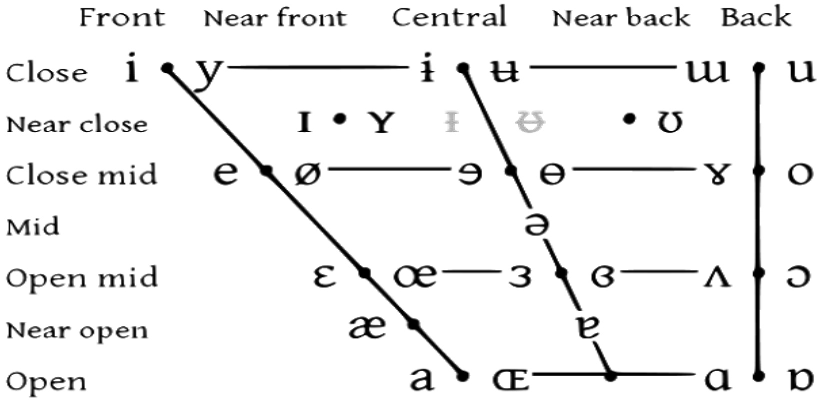




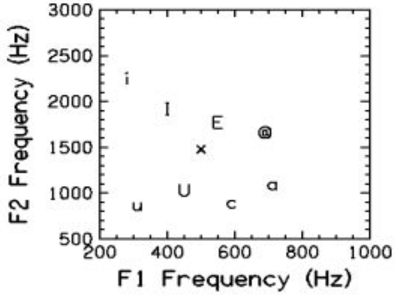
Seeing Formants: the Spectrogram

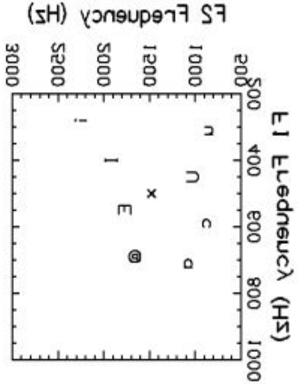


Vowel Space

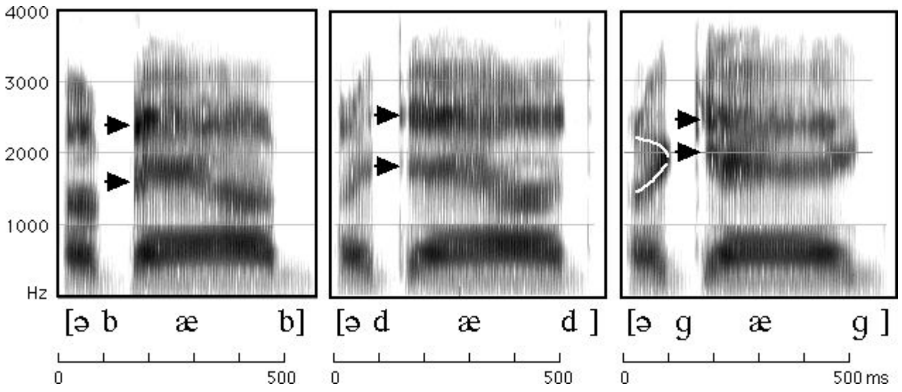


Vowels at right & left of bullets are rounded & unrounded.





Pronunciation is Context Dependent



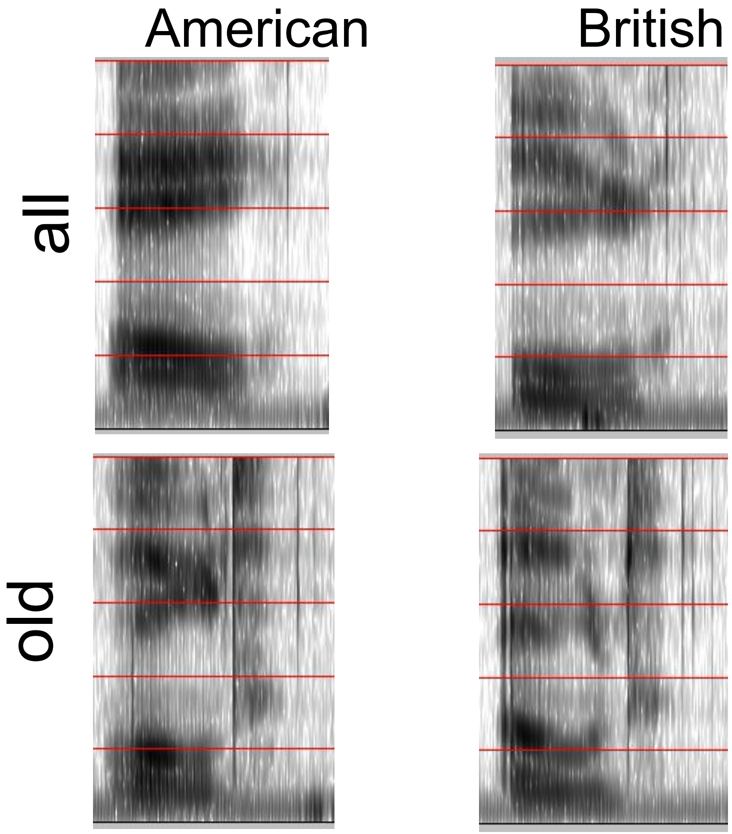
- [bab]: closure of lips lowers all formants: so rapid increase in all formants at beginning of "bab"
- [dad]: first formant increases, but F2 and F3 slight fall
- [gag]: F2 and F3 come together: this is a characteristic of velars. Formant transitions take longer in velars than in alveolars or labials

From Ladefoged “A Course in Phonetics”

Dialect Issues

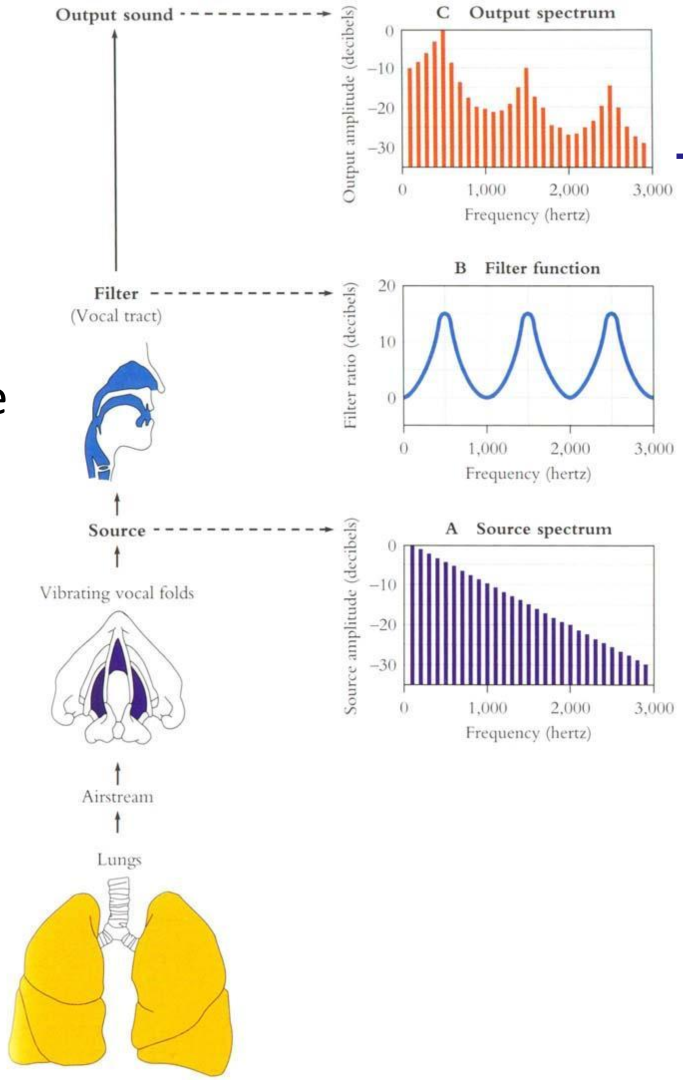
- Speech varies from dialect to dialect (examples are American vs. British English)
  - Syntactic ("I could" vs. "I could do")
  - Lexical ("elevator" vs. "lift")
  - Phonological
  - Phonetic

- Mismatch between training and testing dialects can cause a large increase in error rate



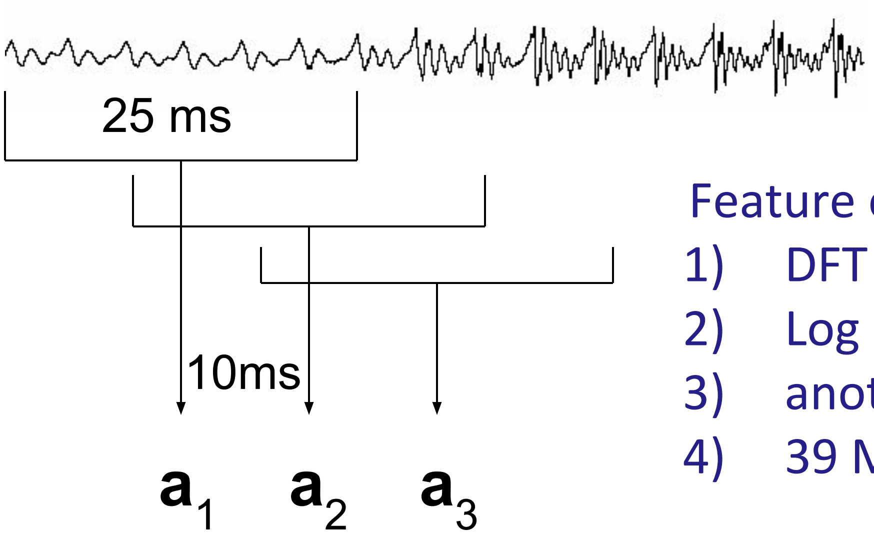
Why these Peaks?

- Articulation process:
  - The vocal cord vibrations create harmonics
  - The mouth is an amplifier
  - Depending on shape of mouth, some harmonics are amplified more than others



Frame Extraction

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



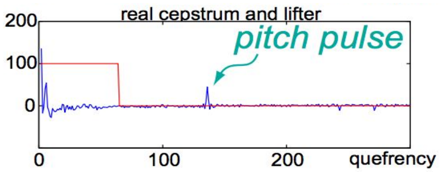
Feature extraction for each frame:

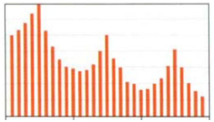
1) DFT (Spectrum)
2) Log (Calibrate)
3) another DFT (Cepstrum)
4) 39 MFCC features

Figure: Simon Arnfield

Final Feature Vector

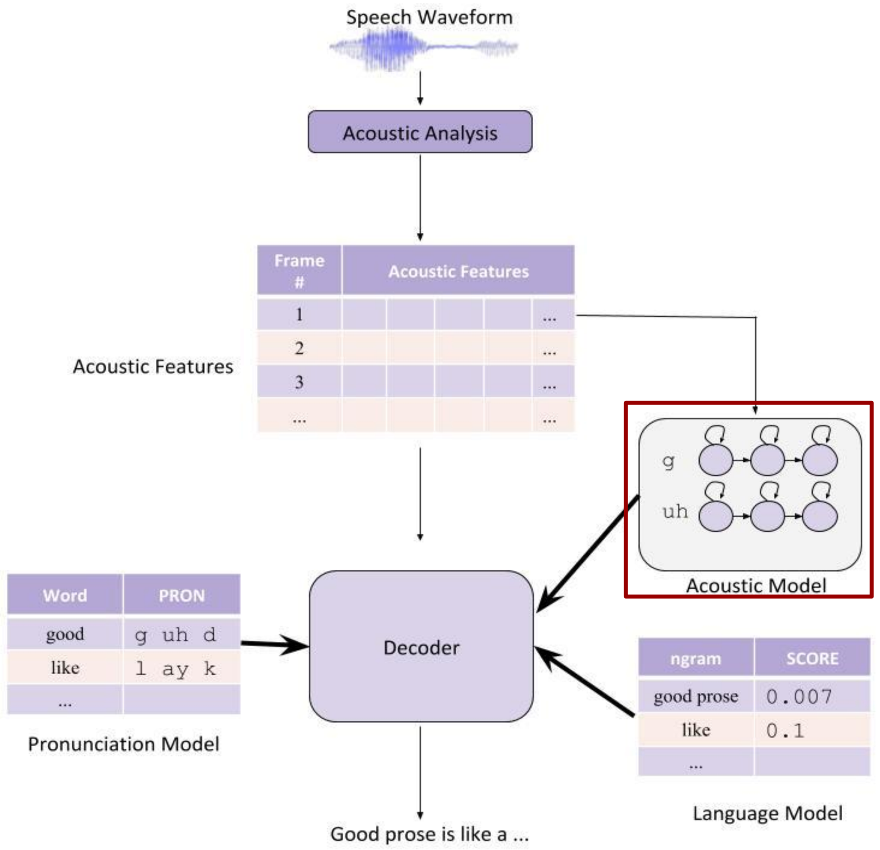
- 39 (real) features per 25 ms frame:
  - 12 MFCC features
  - 12 delta MFCC features
  - 12 delta-delta MFCC features
  - 1 (log) frame energy
  - 1 delta (log) frame energy
  - 1 delta-delta (log frame energy)





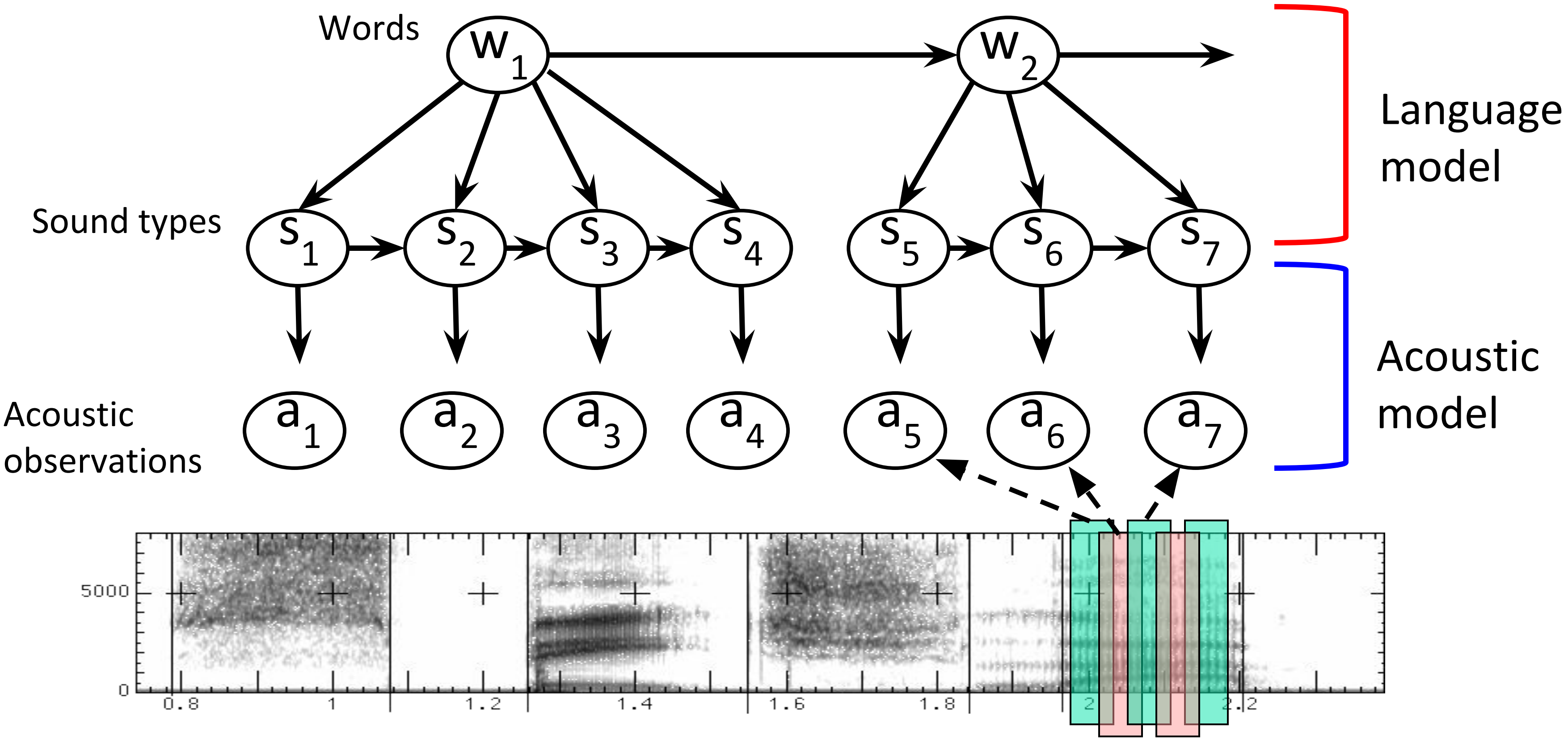
- So each frame is represented by a 39D vector

Acoustic Modeling

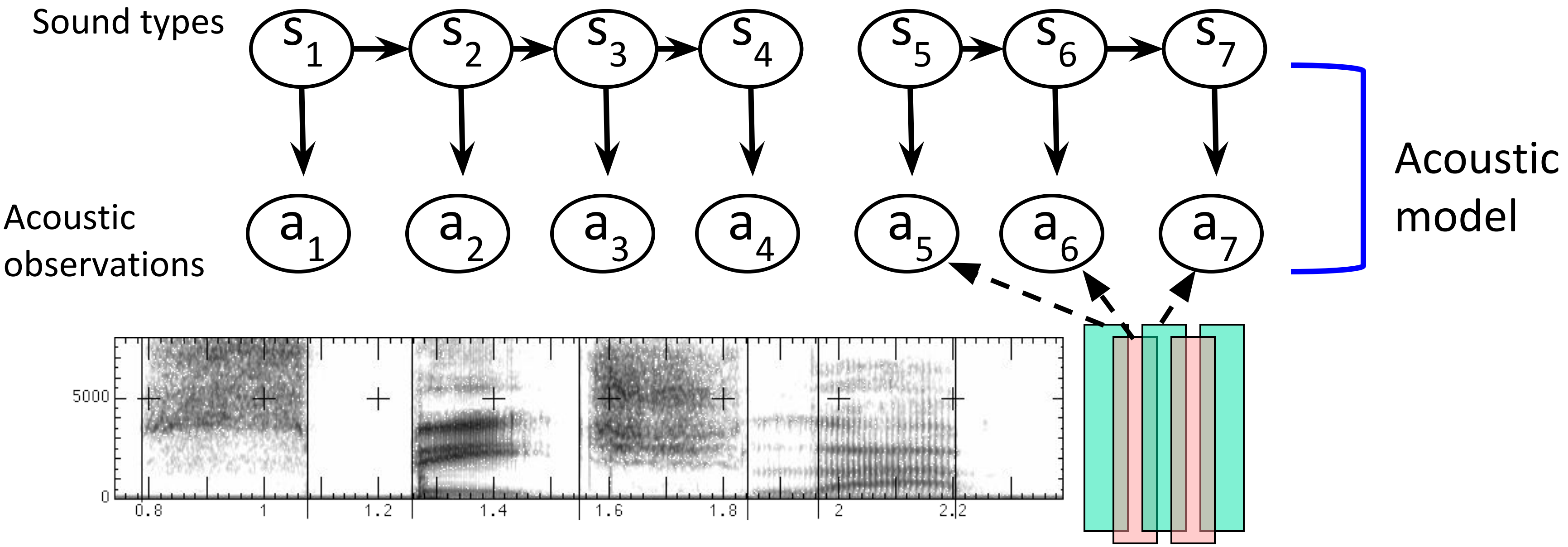


Slide by Preethi Jyothi

Speech Model



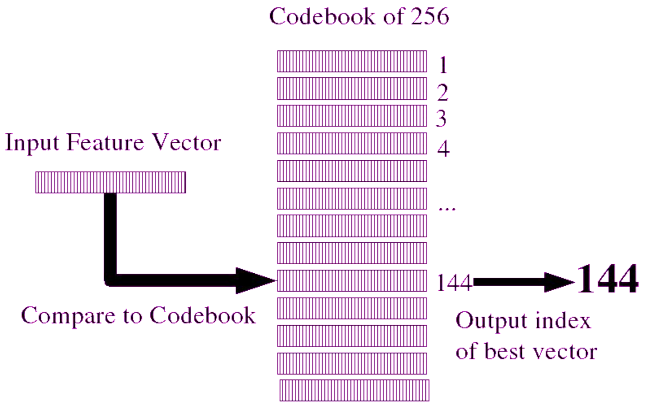
Acoustic Model



Naive Solution:
Vector Quantisation

Vector Quantization

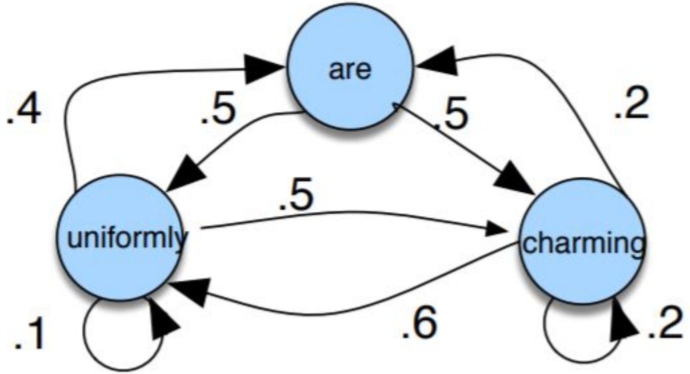
- Idea: discretization
  - Map MFCC vectors onto discrete symbols
  - Compute probabilities just by counting



- This is called vector quantization or VQ
- Not used for ASR any more
- But: useful to consider as a starting point

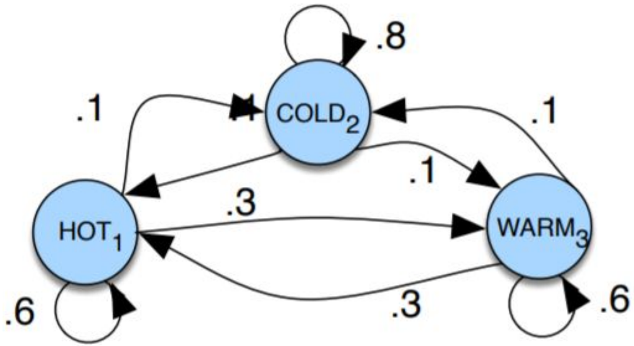
Hidden Markov Models

Markov Chain: words



the future is independent of the past given the present

Markov Chain: weather

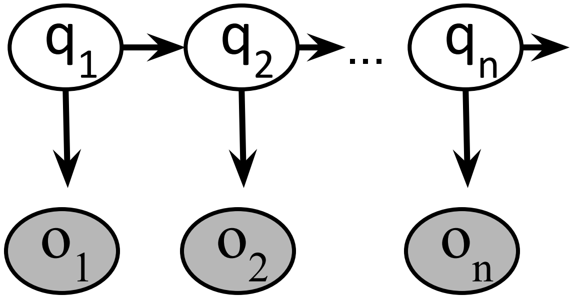


$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

From J&M

HMM

- In real world many events are not observable
  - Speech recognition: we observe acoustic features but not the phones
  - POS tagging: we observe words but not the POS tags



Markov Assumption: $P(q_i|q_1...q_{i-1}) = P(q_i|q_{i-1})$

Output Independence: $P(o_i|q_1 \ldots q_i, \ldots, q_T, o_1, \ldots, o_i, \ldots, o_T) = P(o_i|q_i)$

Generative vs. Discriminative models

- Generative models specify a joint distribution over the labels and the data. With this you could generate new data

$$P(x,y) = P(y)\,P(x \mid y)$$

- Discriminative models specify the conditional distribution of the label y given the data x. These models focus on how to discriminate between the classes

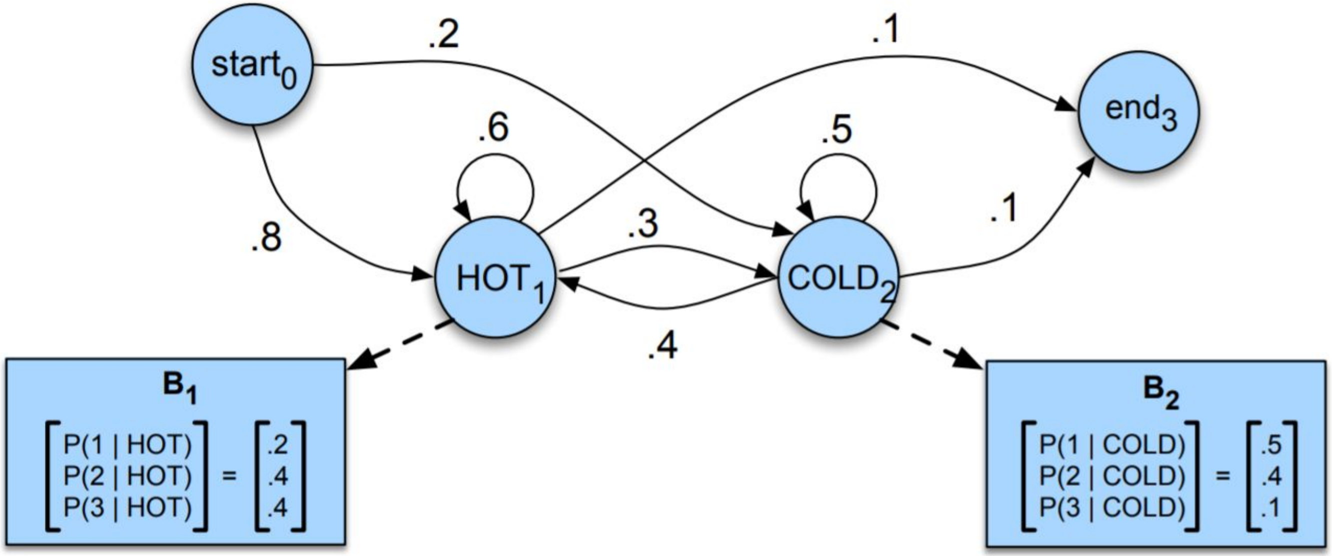
$$P(y \mid x)$$

From Bamman

HMM in Language Technologies

- Part-of-speech tagging (Church, 1988; Brants, 2000)
- Named entity recognition (Bikel et al., 1999) and other information extraction tasks
- Text chunking and shallow parsing (Ramshaw and Marcus, 1995)
- Word alignment of parallel text (Vogel et al., 1996)
- Acoustic models in speech recognition (emissions are continuous)
- Discourse segmentation (labeling parts of a document)

HMM example



Markov Assumption: $P(q_i|q_1 ... q_{i-1}) = P(q_i|q_{i-1})$

Output Independence: $P(o_i|q_1 \dots q_i, \dots, q_T, o_1, \dots, o_i, \dots, o_T) = P(o_i|q_i)$

From J&M

HMM

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
q_0, q_F	a special **start state** and **end (final) state** that are not associated with observations, together with transition probabilities $a_{01} a_{02} \ldots a_{0n}$ out of the start state and $a_{1F} a_{2F} \ldots a_{nF}$ into the end state

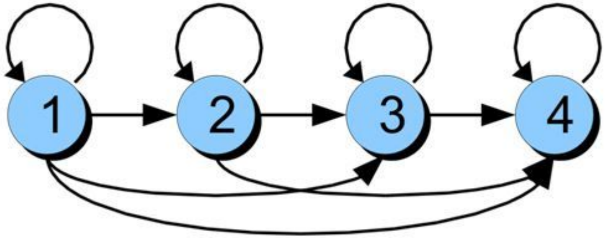
From J&M

HMM Parameters

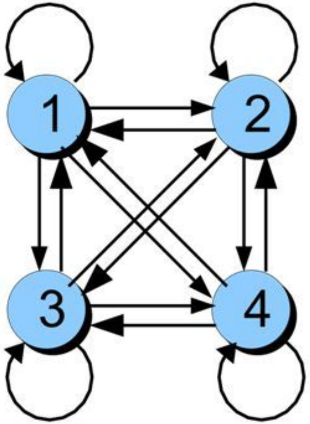
$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
q_0, q_F	a special **start state** and **end (final) state** that are not associated with observations, together with transition probabilities $a_{01} a_{02} \ldots a_{0n}$ out of the start state and $a_{1F} a_{2F} \ldots a_{nF}$ into the end state

From J&M

Types of HMMs



Bakis = left-to-right



Ergodic = fully-connected

- + many more

From J&M

HMMs:Questions

An influential tutorial by Rabiner (1989), based on tutorials by Jack Ferguson in the 1960s, introduced the idea that hidden Markov models should be characterized by **three fundamental problems**:

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

From J&M

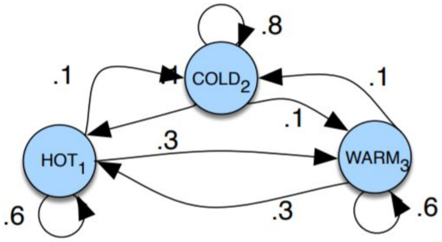
HMMs:Algorithms

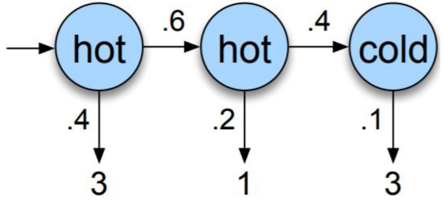
Forward	**Problem 1 (Likelihood):**	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Viterbi	**Problem 2 (Decoding):**	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Forward–Backward; Baum–Welch	**Problem 3 (Learning):**	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

From J&M

Likelihood Computation

Problem 1 (Likelihood): Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.





$$P(3\ 1\ 3, \text{hot hot cold}) = P(\text{hot}|\text{start}) \times P(\text{hot}|\text{hot}) \times P(\text{cold}|\text{hot}) \times P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold})$$

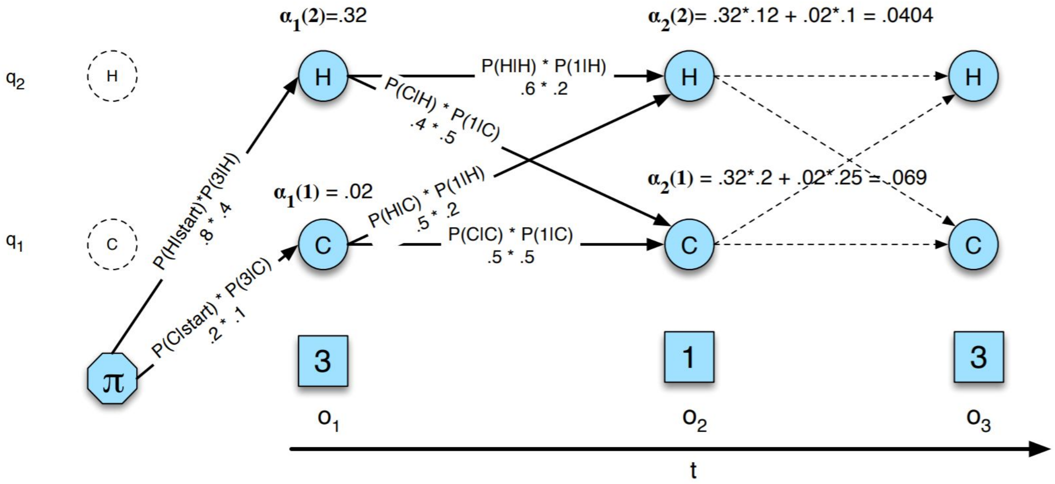
$$P(3\ 1\ 3) = P(3\ 1\ 3, \text{cold cold cold}) + P(3\ 1\ 3, \text{cold cold hot}) + P(3\ 1\ 3, \text{hot hot cold}) + \ldots$$

P(HHC)=? P(313)=?

Complexity?

From J&M

Forward Trellis



$$\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$$

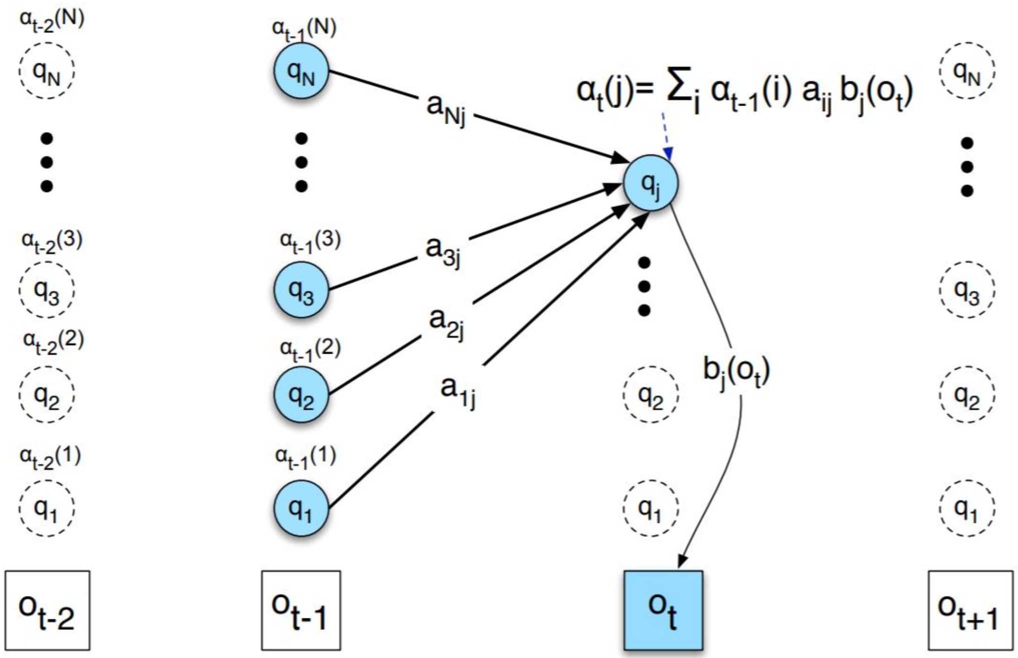
$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

$\alpha_{t-1}(i)$	the **previous forward path probability** from the previous time step
a_{ij}	the **transition probability** from previous state q_i to current state q_j
$b_j(o_t)$	the **state observation likelihood** of the observation symbol o_t given the current state j

P(313)=?

From J&M

Forward Algorithm



Complexity?

From J&M

Forward Algorithm

1. Initialization:

$$\alpha_1(j) \;=\; a_{0j}b_j(o_1) \;\; 1 \le j \le N$$

2. Recursion (since states 0 and F are non-emitting):

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \le j \le N, 1 < t \le T$$

3. Termination:

$$P(O|\lambda) = \alpha_T(q_F) = \sum_{i=1}^{N} \alpha_T(i)\, a_{iF}$$

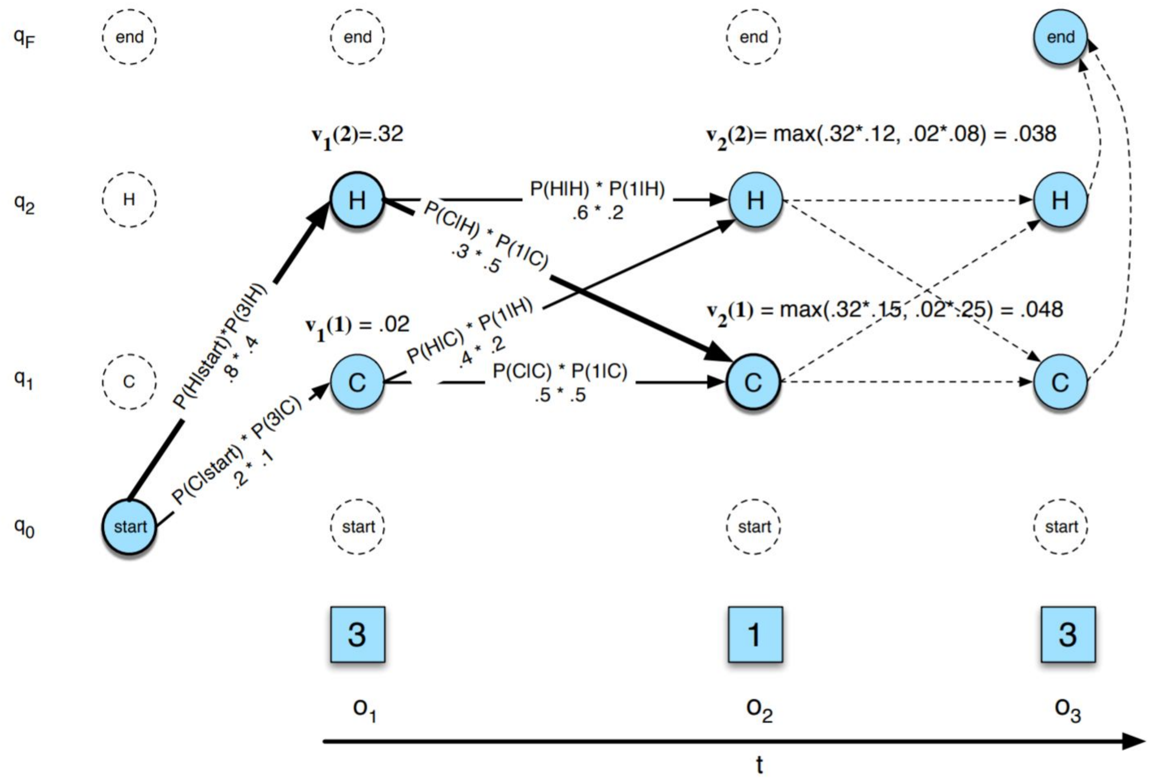
HMMs:Questions

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

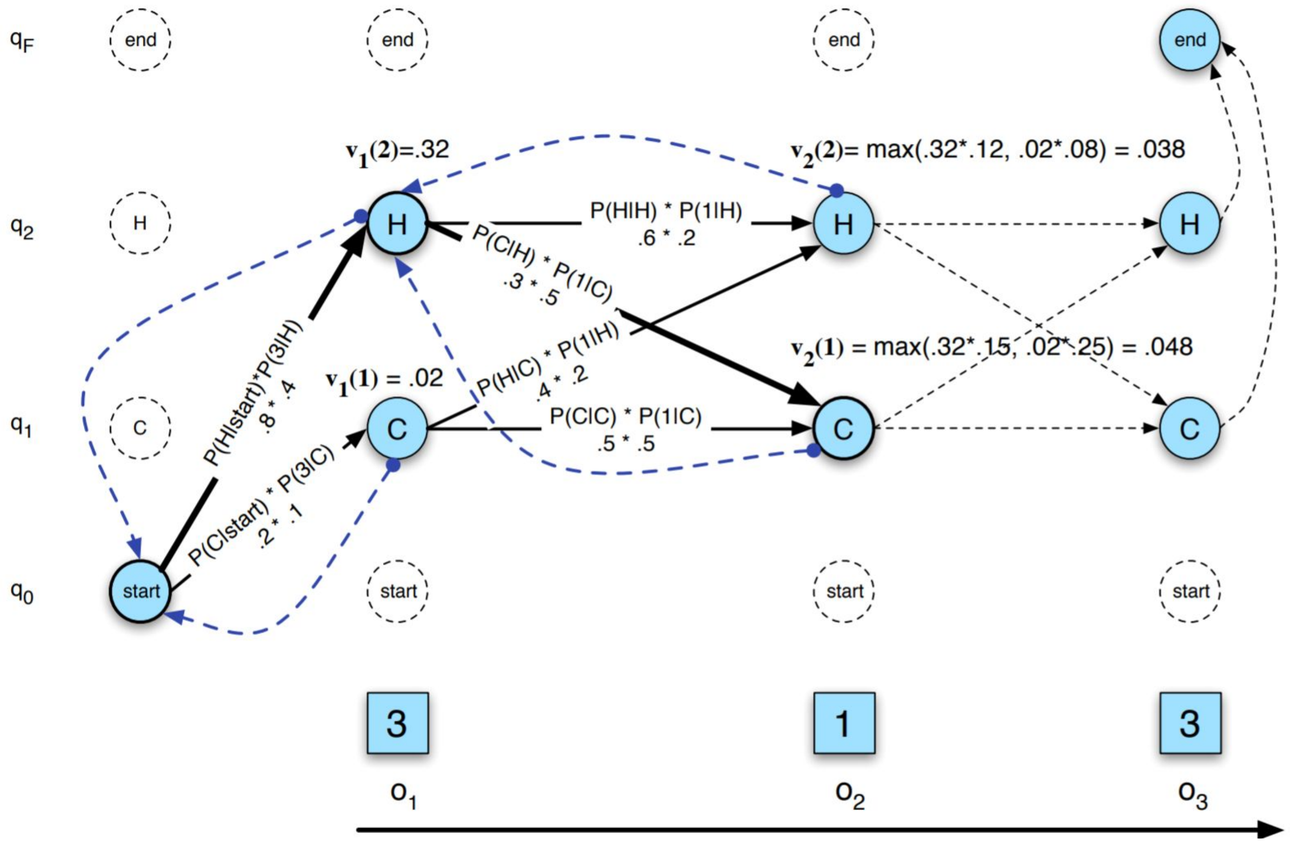
From J&M

Viterbi Trellis

$$v_t(j) = \max_{q_0,q_1,\ldots,q_{t-1}} P(q_0,q_1 \ldots q_{t-1},o_1,o_2 \ldots o_t,q_t = j|\lambda) \qquad v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$$



Viterbi Backtrace



Viterbi Algorithm

1. **Initialization:**

$$v_1(j) = a_{0j} b_j(o_1) \quad 1 \leq j \leq N$$
$$bt_1(j) = 0$$

2. **Recursion** (recall that states 0 and q_F are non-emitting):

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$
$$bt_t(j) = \operatorname{argmax}_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$

3. **Termination:**

The best score: $P* = v_T(q_F) = \max_{i=1}^{N} v_T(i) * a_{iF}$

The start of backtrace: $q_T* = bt_T(q_F) = \operatorname{argmax}_{i=1}^{N} v_T(i) * a_{iF}$

Viterbi

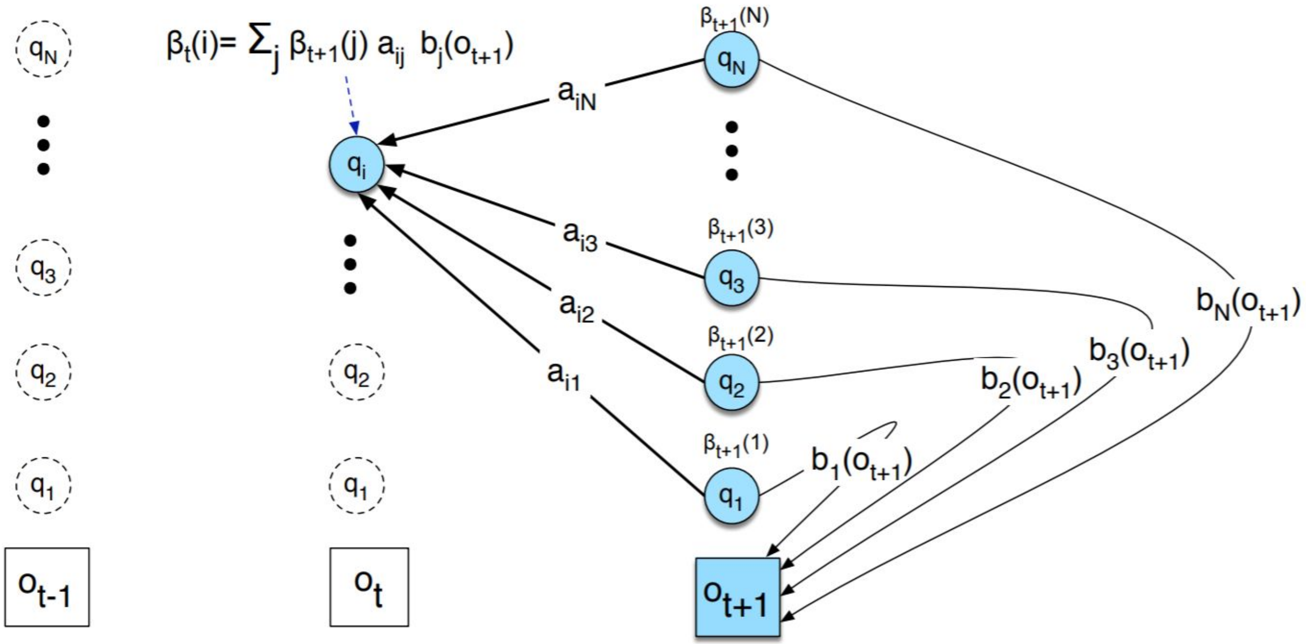
- *n*-best decoding
- relationship to sequence alignment
-

Citation	Field
Viterbi (1967)	information theory
Vintsyuk (1968)	speech processing
Needleman and Wunsch (1970)	molecular biology
Sakoe and Chiba (1971)	speech processing
Sankoff (1972)	molecular biology
Reichert et al. (1973)	molecular biology
Wagner and Fischer (1974)	computer science

HMMs:Questions

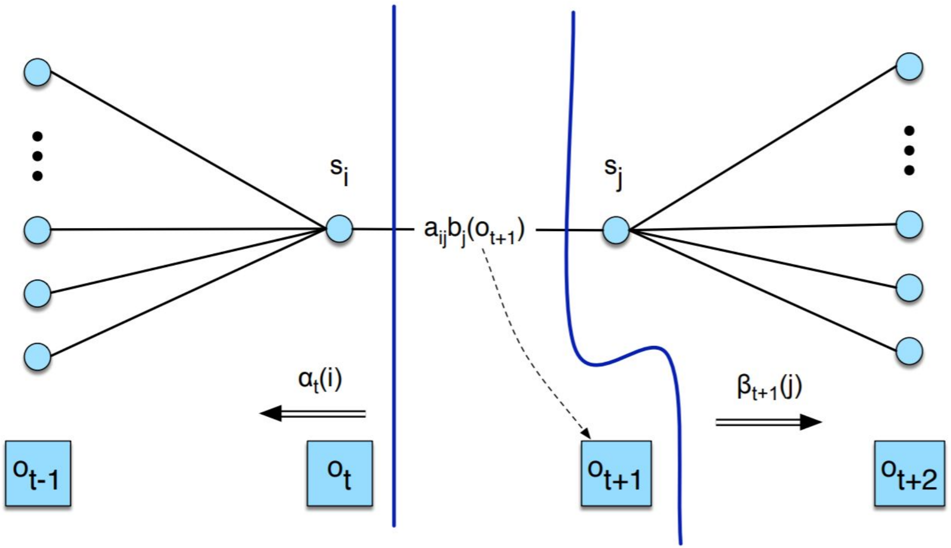
Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Backward



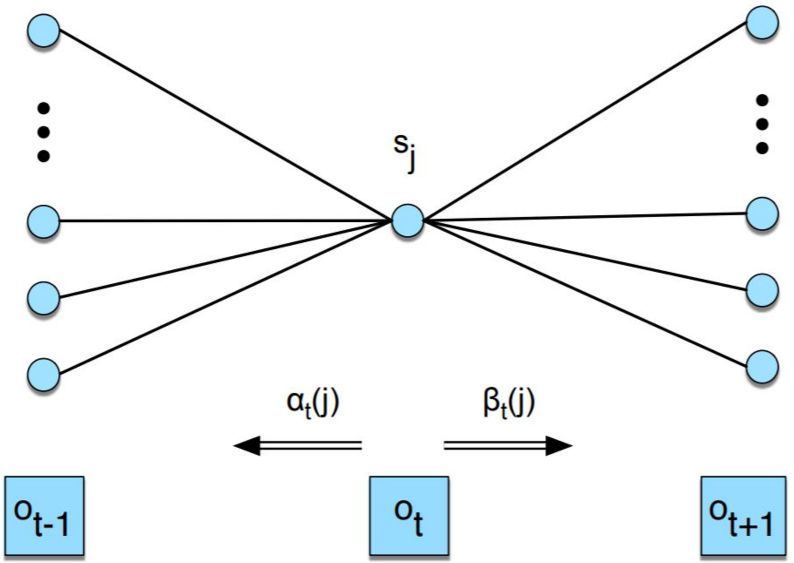
$\beta_t(i) = \sum_j \beta_{t+1}(j)\, a_{ij}\, b_j(o_{t+1})$
q_N
q_3
q_2
q_1
o_{t-1}
q_i
o_t
a_{iN}
a_{i3}
a_{i2}
a_{i1}
$\beta_{t+1}(N)$
$\beta_{t+1}(3)$
$\beta_{t+1}(2)$
$\beta_{t+1}(1)$
o_{t+1}
$b_1(o_{t+1})$
$b_2(o_{t+1})$
$b_3(o_{t+1})$
$b_N(o_{t+1})$

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\alpha_T(q_F)} \quad \forall\, t,\ i, \text{ and } j$$



probability to transition from *i* to *j* at time *t* given O

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\alpha_T(q_F)} \quad \forall\, t \text{ and } j$$



probability to being in state *j* at time *t*

Forward-Backward

function FORWARD-BACKWARD(*observations* of len T, *output vocabulary* V, *hidden state set* Q) **returns** *HMM=(A,B)*

initialize A and B
iterate until convergence
E-step

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\alpha_T(q_F)} \quad \forall\, t \text{ and } j$$

$$\xi_t(i,j) = \frac{\alpha_t(i)\,a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(q_F)} \quad \forall\, t,\ i, \text{ and } j$$

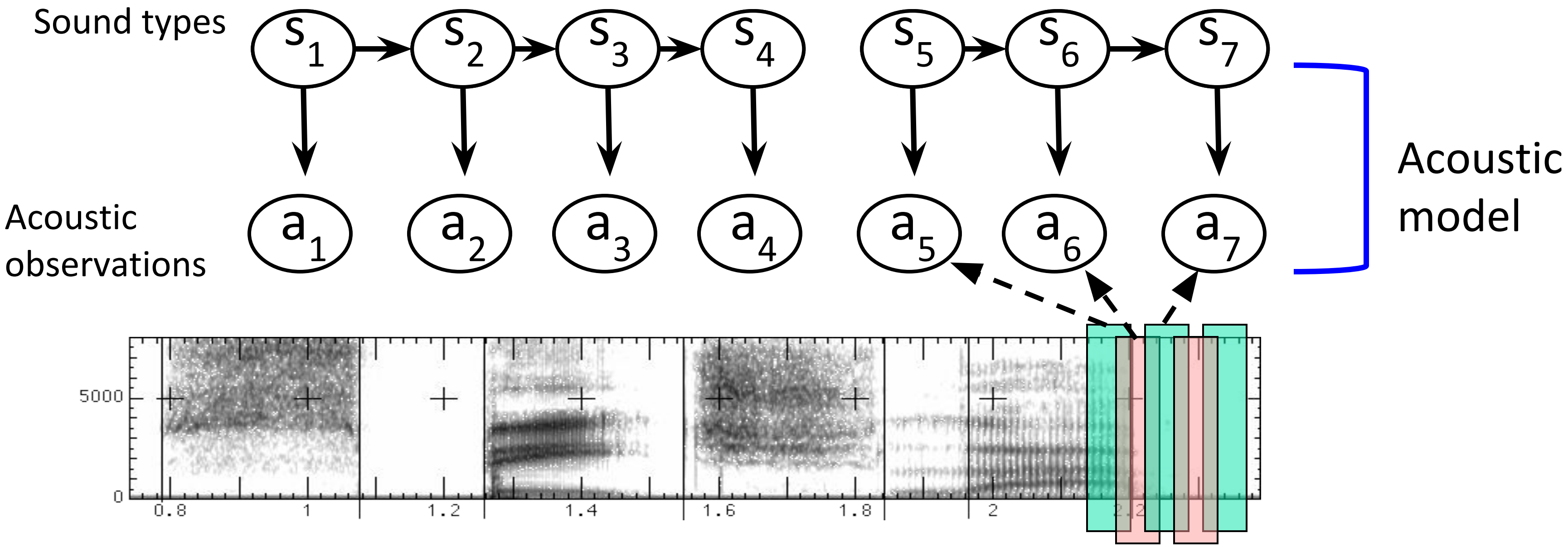
M-step

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i,k)}$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 \text{ s.t. } O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

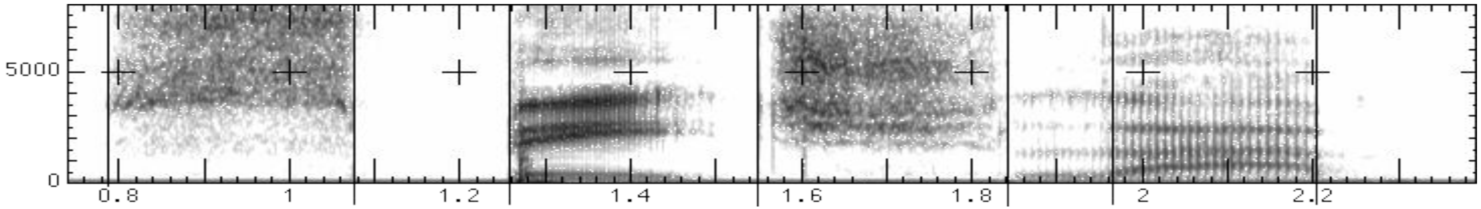
return A, B

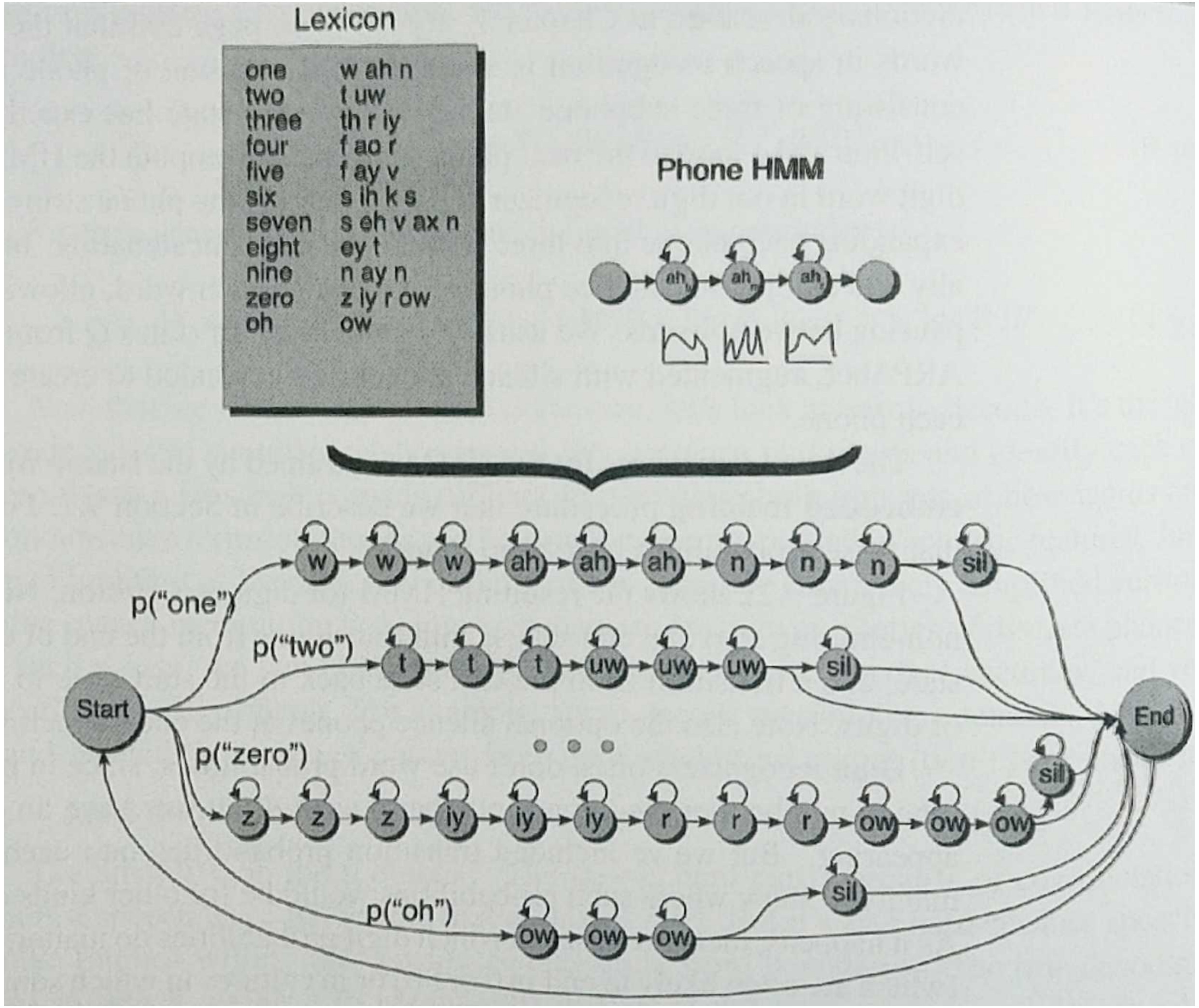
Acoustic Model



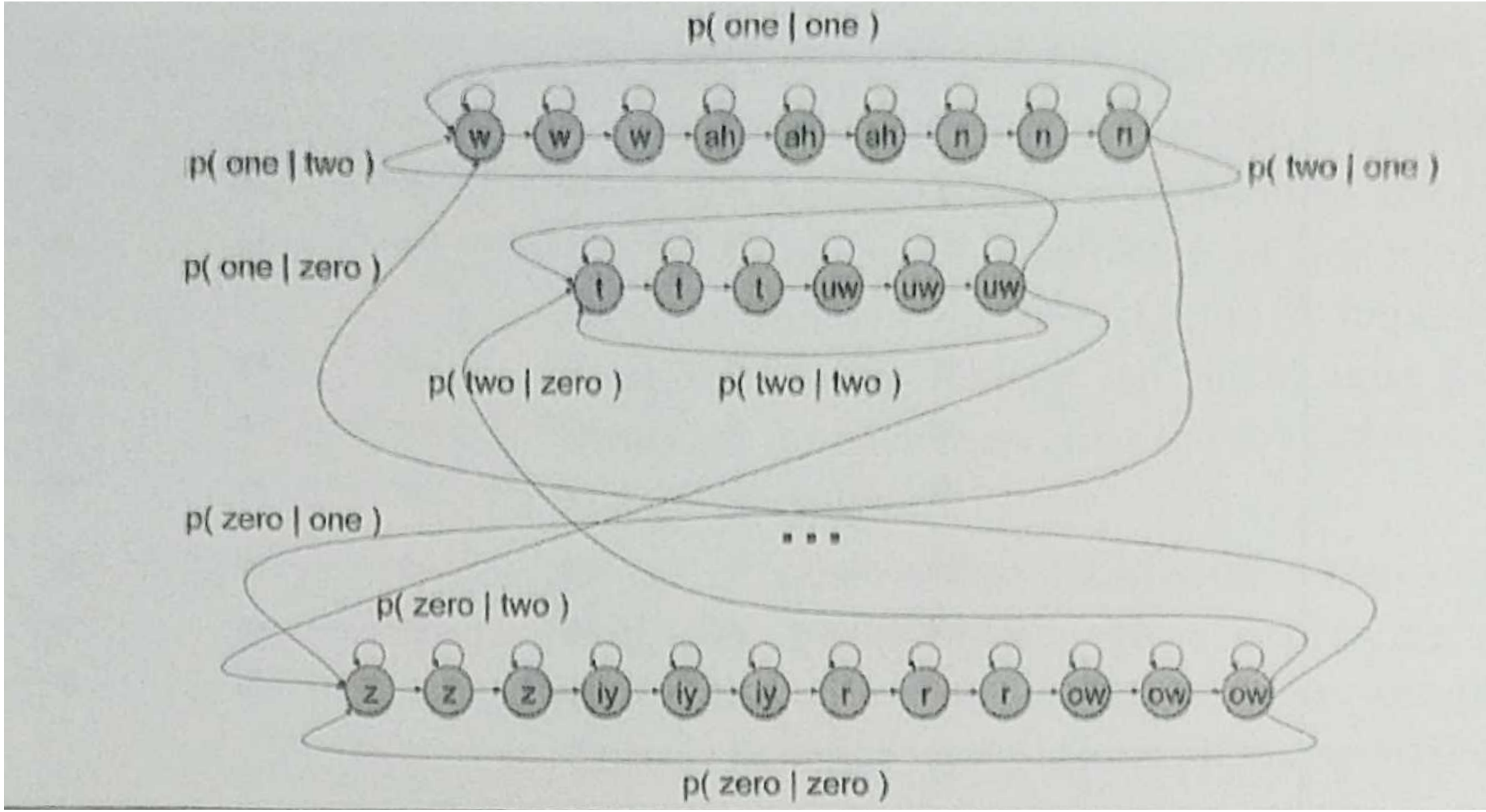
“speech lab”

ssssssssppppeeeeeeetshshshshllllaeaeaebbbbb



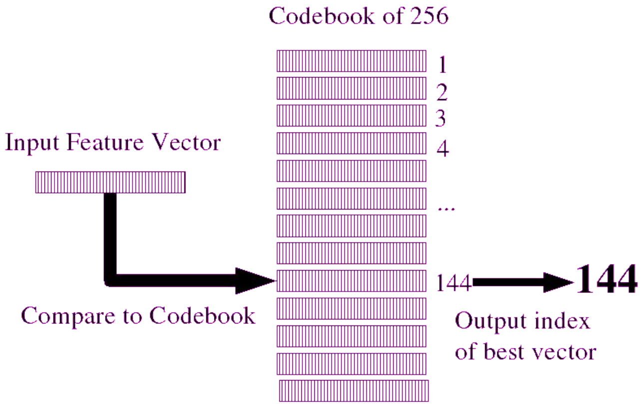


Lexicon
one w ah n
two t uw
three th r iy
four f ao r
five f ay v
six s ih k s
seven s eh v ax n
eight ey t
nine n ay n
zero z iy r ow
oh ow
Phone HMM
ah
ah
ah
p("one")
w w w ah ah ah n n n sil
p("two")
t t t uw uw uw sil
Start
End
p("zero")
z z z iy iy iy r r r ow ow ow sil
p("oh")
ow ow ow sil


p(one | one)
p(one | two)
w
w
w
ah
ah
ah
n
n
n
p(two | one)
p(one | zero)
t
t
t
uw
uw
uw
p(two | zero)
p(two | two)
p(zero | one)
...
p(zero | two)
z
z
z
iy
iy
iy
r
r
r
ow
ow
ow
p(zero | zero)

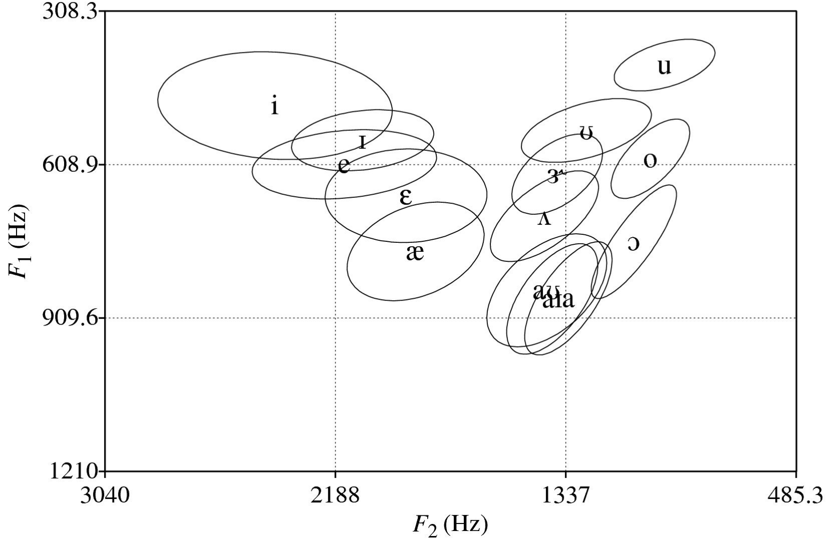
Vector Quantization

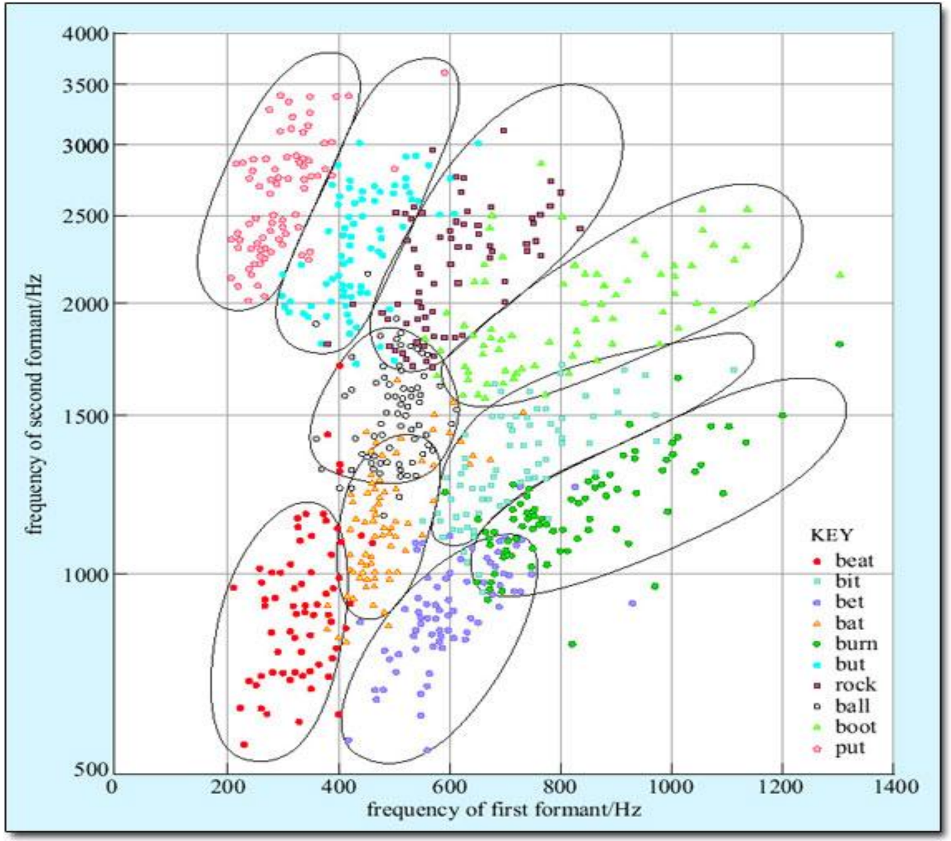
- Idea: discretization
  - Map MFCC vectors onto discrete symbols
  - Compute probabilities just by counting



- This is called vector quantization or VQ
- Not used for ASR any more
- But: useful to consider as a starting point

Issues with Codebook



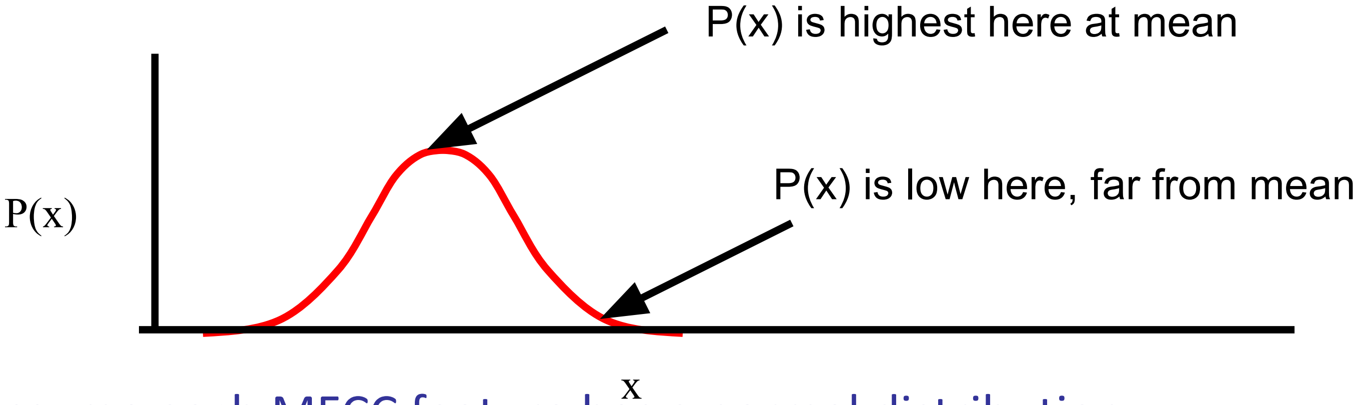


Gaussians for Acoustic Modeling

- P(x):

A Gaussian is parameterized by a mean and a variance:

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



- let's assume each MFCC feature has a normal distribution

Multivariate Gaussians

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

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

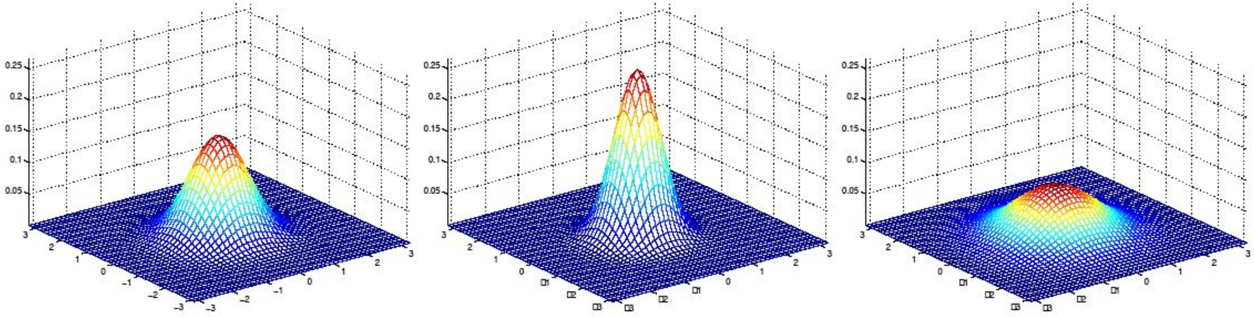
- Vector of means μ and covariance matrix Σ

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

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

Gaussians: Size of Σ

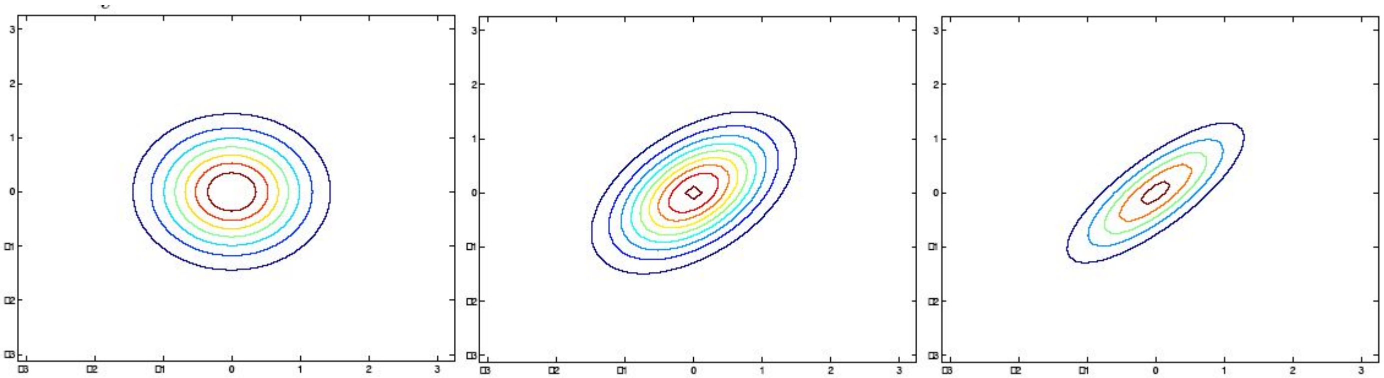
- $\mu = [0\ 0]$ $\mu = [0\ 0]$ $\mu = [0\ 0]$
- $\Sigma = I$ $\Sigma = 0.6I$ $\Sigma = 2I$
- As Σ becomes larger, Gaussian becomes more spread out; as Σ becomes smaller, Gaussian more compressed



Text and figures from Andrew Ng

Gaussians: Shape of Σ

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

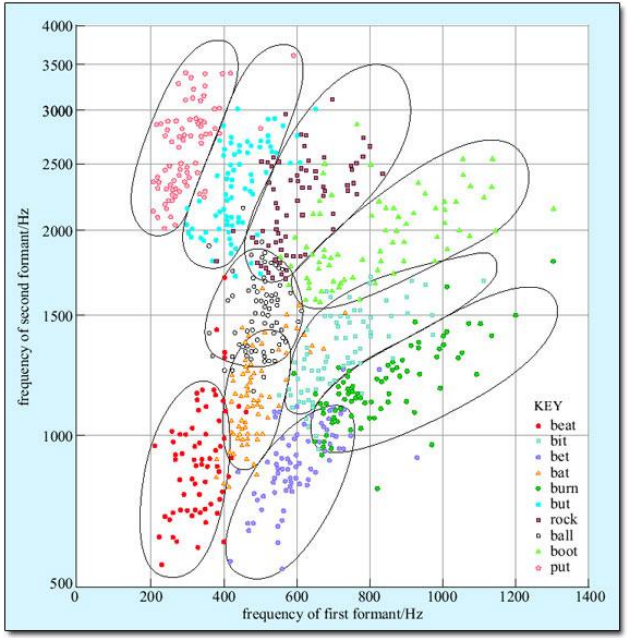


$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \quad .\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

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



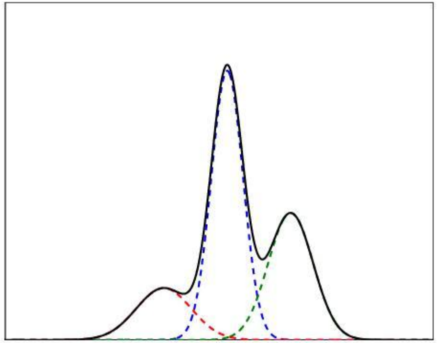
From openlearn.open.ac.uk

Mixtures of Gaussians

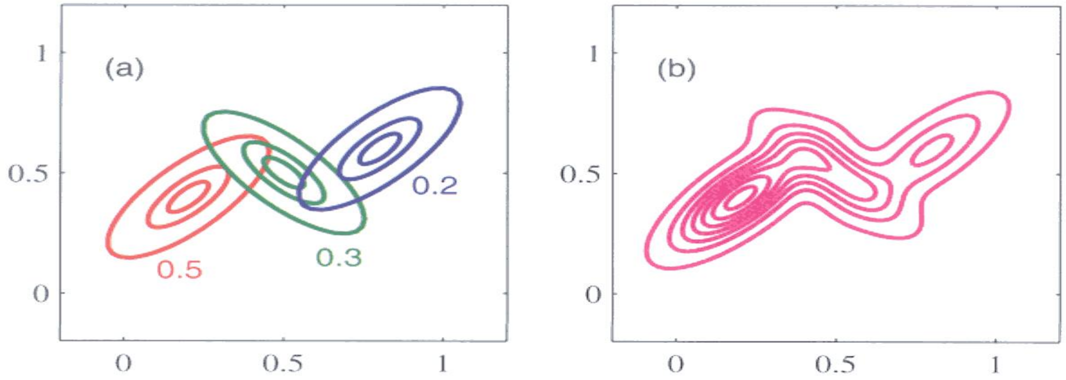
- Mixtures of Gaussians:

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

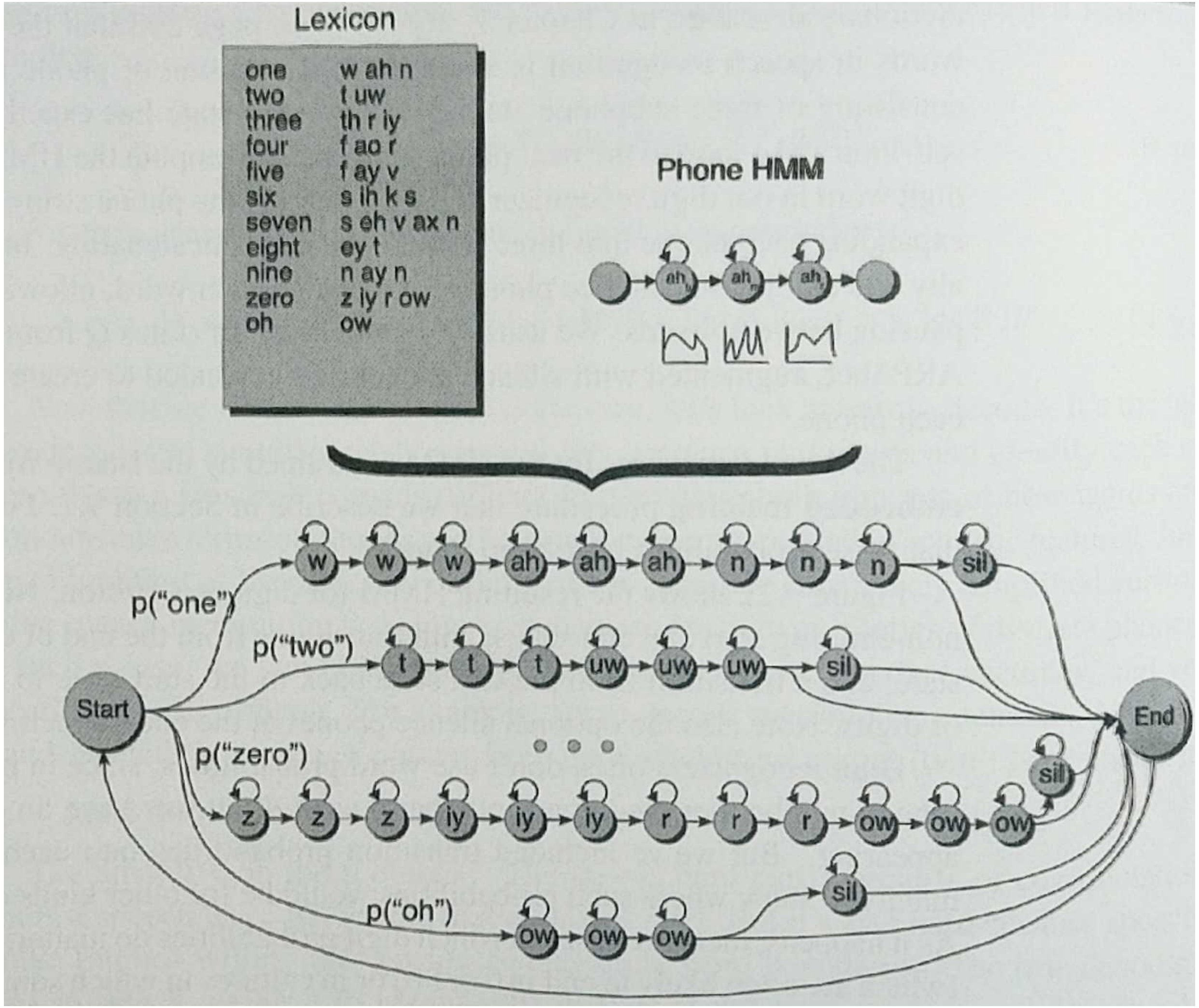
$$P(x|\mu, \mathbf{\Sigma}, \mathbf{c}) = \sum_i c_i P(x|\mu_i, \Sigma_i)$$

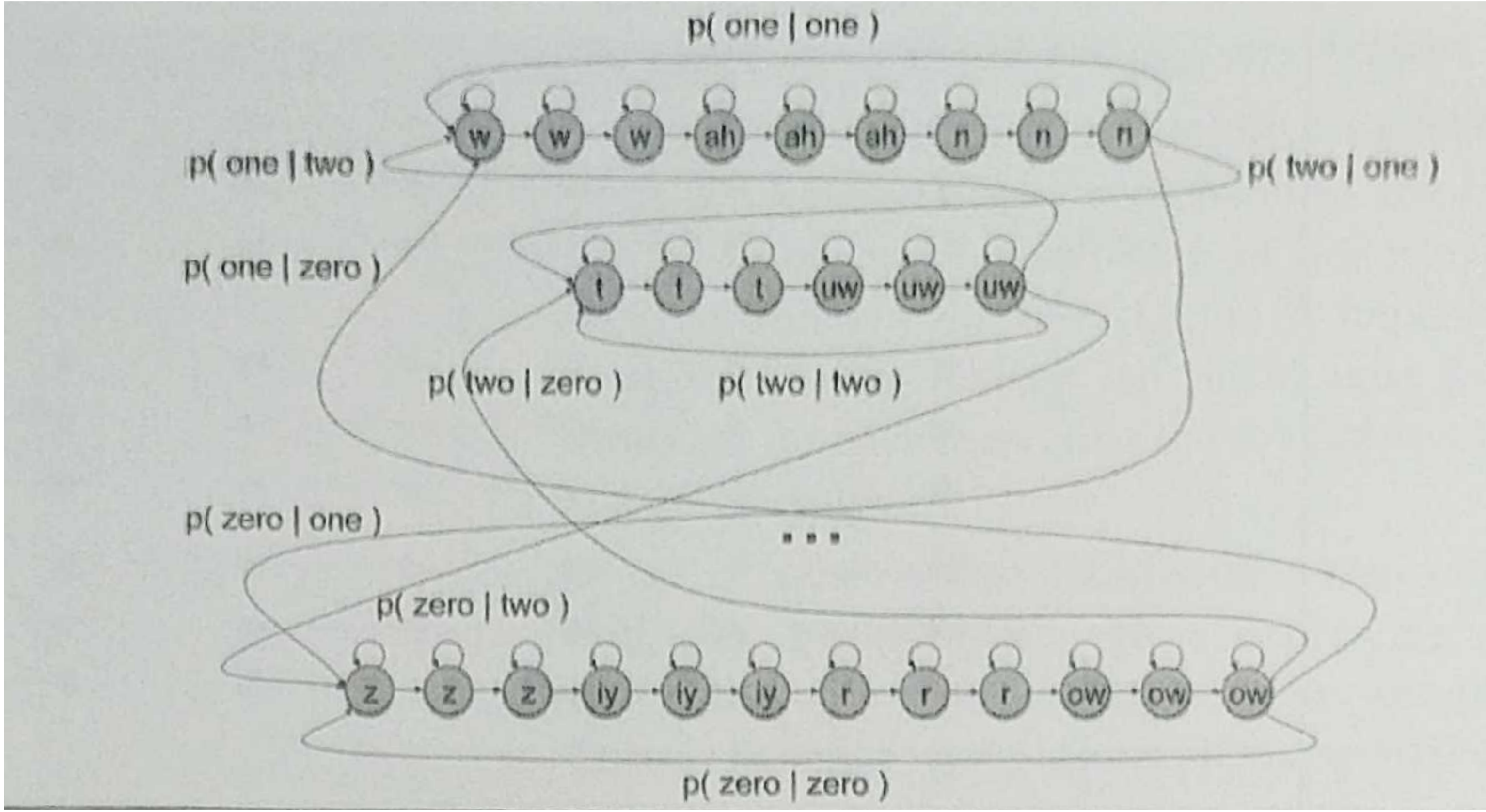


From robots.ox.ac.uk



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

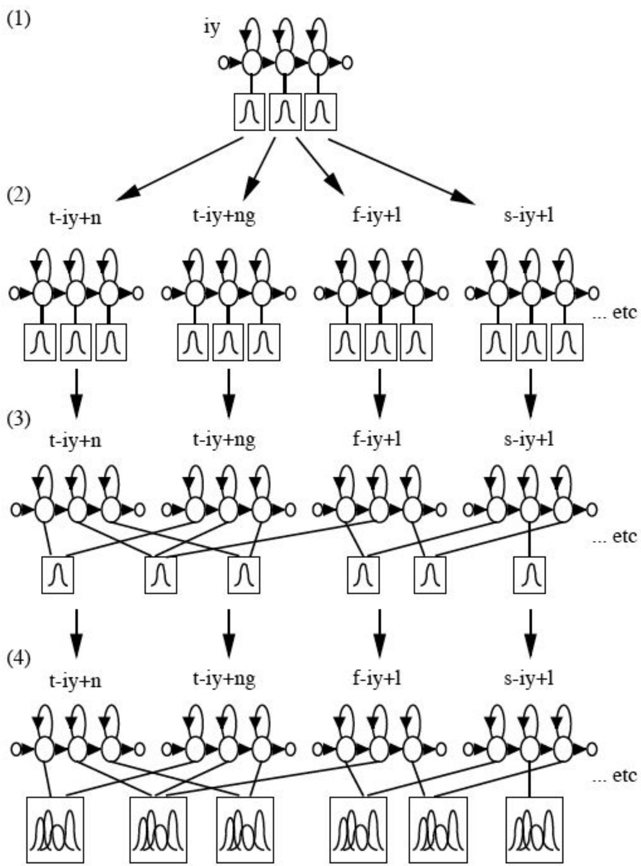

Lexicon
one w ah n
two t uw
three th r iy
four f ao r
five f ay v
six s ih k s
seven s eh v ax n
eight ey t
nine n ay n
zero z iy r ow
oh ow
Phone HMM
p("one")
p("two")
p("zero")
p("oh")
Start
End
sil


p(one | one)
w
w
w
ah
ah
ah
n
n
n
p(one | two)
p(two | one)
p(one | zero)
t
t
t
uw
uw
uw
p(two | zero)
p(two | two)
p(zero | one)
...
p(zero | two)
z
z
z
iy
iy
iy
r
r
r
ow
ow
ow
p(zero | zero)

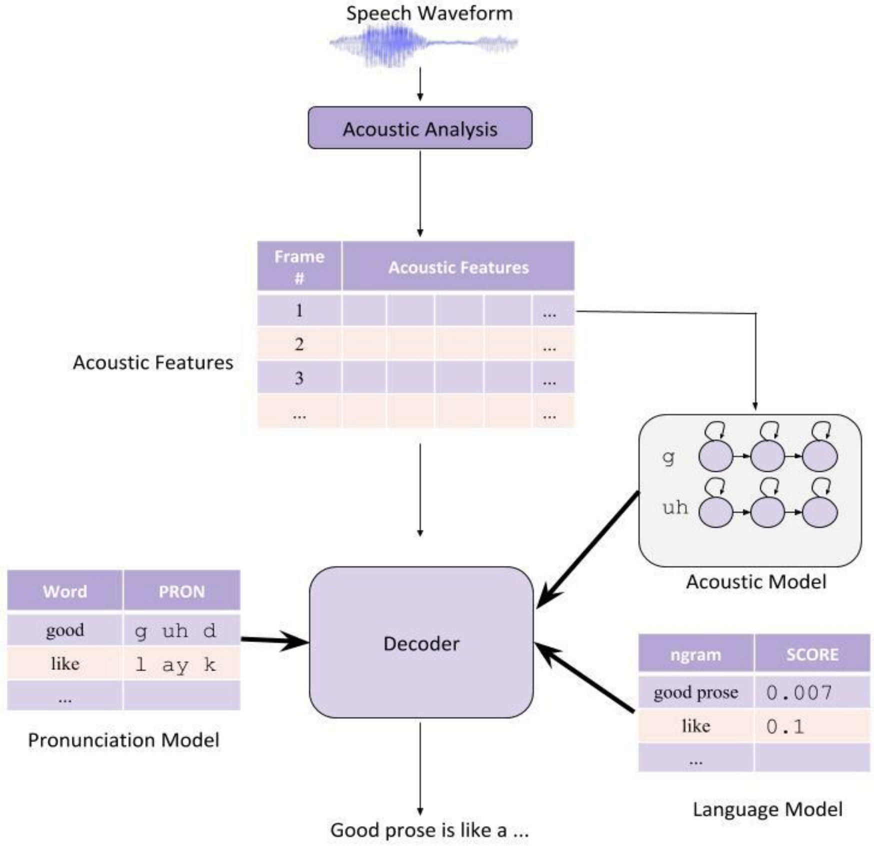
State Tying

- 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



Acoustic Modeling



Slide by Preethi Jyothi