EE E6820: Speech & Audio Processing & Recognition

Lecture 9: Speech Recognition

1. **Recognizing Speech**
2. **Feature Calculation**
3. **Sequence Recognition**
4. **Hidden Markov Models**

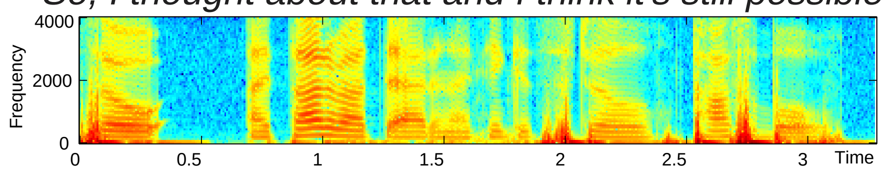
Dan Ellis <dpwe@ee.columbia.edu>
http://www.ee.columbia.edu/~dpwe/e6820/

Columbia University Dept. of Electrical Engineering

Spring 2006

1 Recognizing Speech

"So, I thought about that and I think it's still possible"



- **What kind of information might we want from the speech signal?**
  - words
  - phrasing, 'speech acts' (prosody)
  - mood / emotion
  - speaker identity
- **What kind of processing do we need to get at that information?**
  - time scale of feature extraction
  - signal aspects to capture in features
  - signal aspects to *exclude* from features

Speech recognition as Transcription

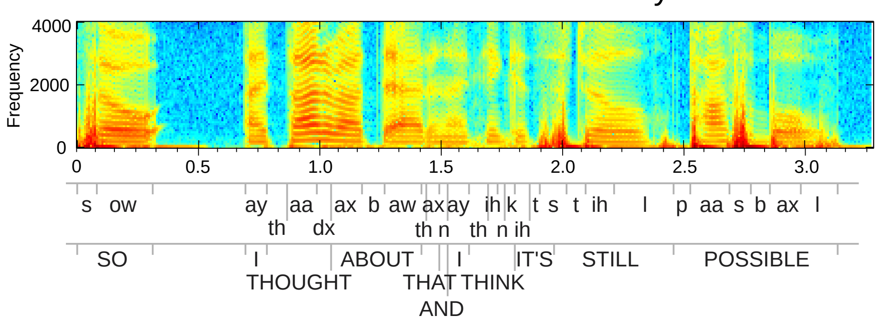
- **Transcription = “speech to text”**
  - find a word string to match the utterance
- **Best suited to small vocabulary tasks**
  - voice dialing, command & control etc.
- **Gives neat objective measure:**
  word error rate (WER) %
  - can be a sensitive measure of performance
- **Three kinds of errors:**

Reference:	THE	CAT	SAT	ON	THE		MAT
Recognized:	–	CAT	SAT	AN	THE	A	MAT
	Deletion			*Substitution*		*Insertion*	

  - WER = (S + D + I) / N

Problems: Within-speaker variability

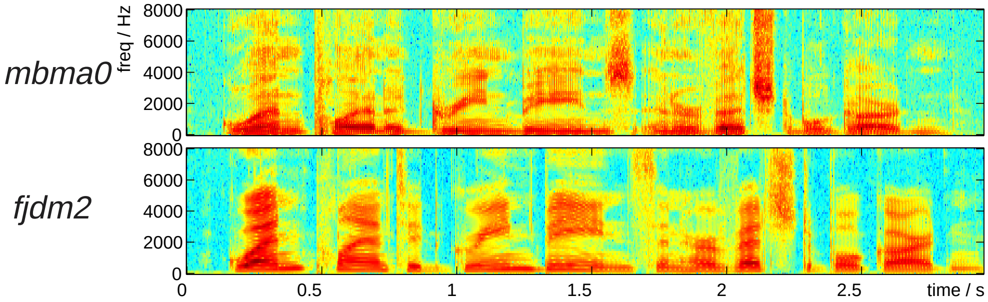
- **Timing variation:**
  - word duration varies enormously



  - fast speech ‘reduces’ vowels
- **Speaking style variation:**
  - careful/casual articulation
  - soft/loud speech
- **Contextual effects:**
  - speech sounds vary with context, role:
    “How **do** you **do**?”

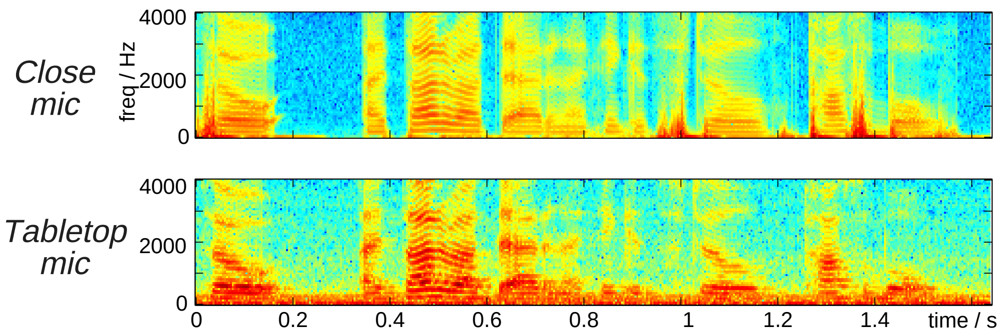
Between-speaker variability

- **Accent variation**
  - regional / mother tongue
- **Voice quality variation**
  - gender, age, huskiness, nasality
- **Individual characteristics**
  - mannerisms, speed, prosody



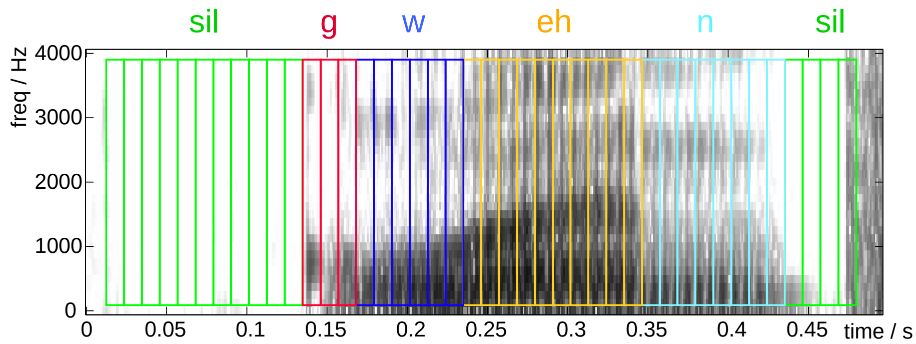
Environment variability

- **Background noise**
  - fans, cars, doors, papers
- **Reverberation**
  - ‘boxiness’ in recordings
- **Microphone/channel**
  - huge effect on relative spectral gain



How to recognize speech?

- **Cross correlate templates?**
  - waveform?
  - spectrogram?
  - time-warp problems
- **Match short-segments & handle time-warp later**
  - model with slices of ~ 10 ms
  - pseudo-stationary model of words:



  - other sources of variation...

Probabilistic formulation

- **Probability that segment label is correct**
  - gives standard form of speech recognizers:

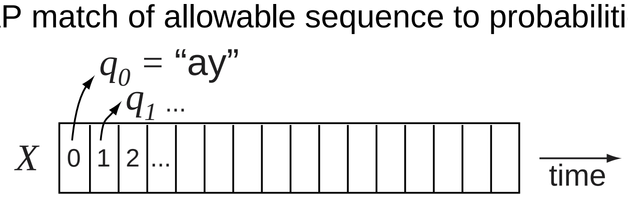
- **Feature calculation**
  transforms signal into easily-classified domain $s[n] \rightarrow X_m \quad \left(m = \frac{n}{H}\right)$

- **Acoustic classifier**
  calculates probabilities of each mutually-exclusive state q^i $\quad p(q^i | X)$

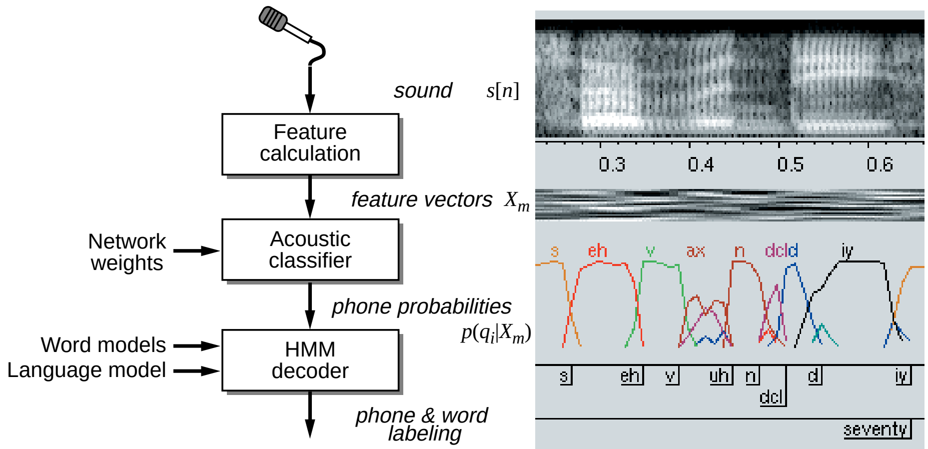
- **'Finite state acceptor' (i.e. HMM)**

$$\hat{Q} = \underset{\{q_0, q_1, \ldots q_L\}}{\operatorname{argmax}} p(q_0, q_1, \ldots q_L | X_0, X_1 \ldots X_L)$$

MAP match of allowable sequence to probabilities:



Standard speech recognizer structure



- **Questions:**
  - what are the best features?
  - how do we do the acoustic classification?
  - how do we find/match the state sequence?

Outline

1. Recognizing Speech
2. **Feature Calculation**
   - Spectrogram, MFCCs & PLP
   - Improving robustness
3. Sequence Recognition
4. Hidden Markov Models

2 Feature Calculation

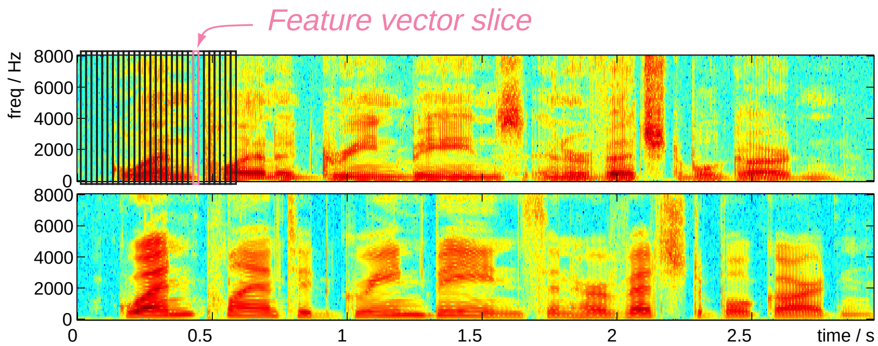
- **Goal: Find a representational space most suitable for classification**
  - waveform: voluminous, redundant, variable
  - spectrogram: better, still quite variable
  - ...?
- **Pattern Recognition: Representation is upper bound on performance**
  - maybe we *should* use the waveform...
  - or, maybe the representation can do *all* the work
- **Feature calculation is intimately bound to classifier**
  - pragmatic strengths and weaknesses
- **Features develop by slow evolution**
  - current choices more historical than principled

Features (1): Spectrogram

- **Plain STFT as features e.g.**

$$X_m[k] = S[mH, k] = \sum_n s[n + mH] \cdot w[n] \cdot e^{-(j2\pi kn)/N}$$

- **Consider examples:**



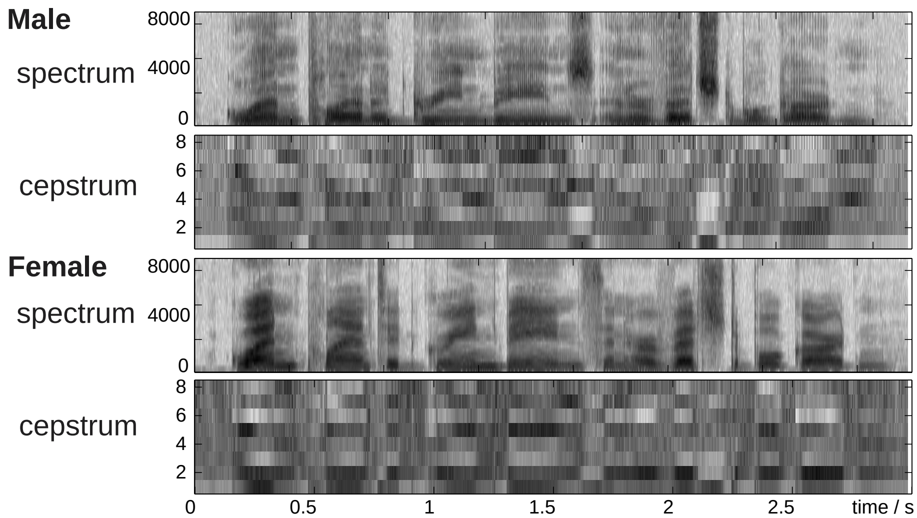
- **Similarities between corresponding segments**
  - but still large differences

Features (2): Cepstrum

- **Idea: Decorrelate, summarize spectral slices:**

$$X_m[l] = IDFT\{\log|S[mH, k]|\}$$

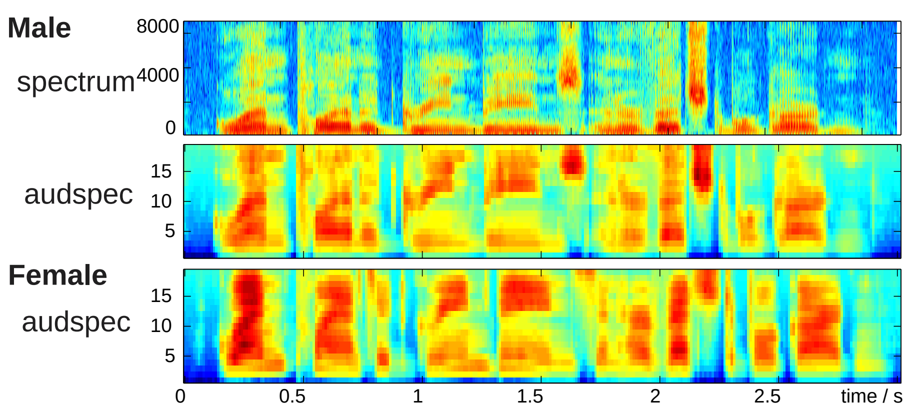
  - good for Gaussian models
  - greatly reduce feature dimension



Features (3): Frequency axis warp

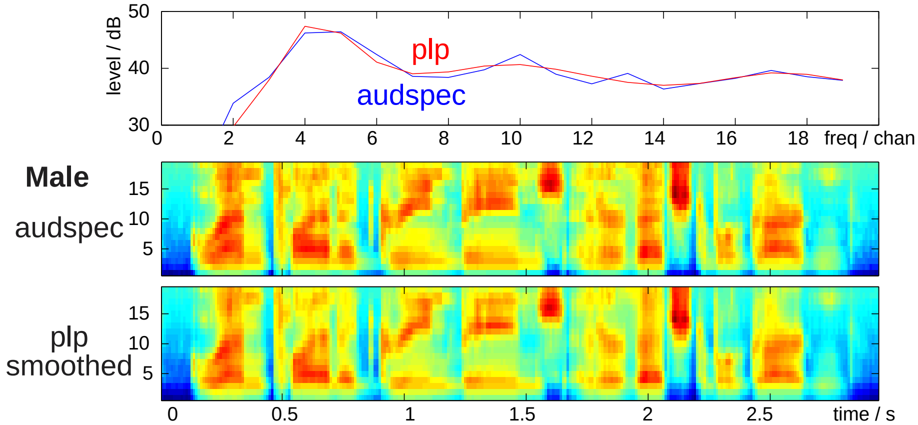
- **Linear frequency axis gives equal 'space' to 0-1 kHz and 3-4 kHz**
  - but perceptual importance very different
- **Warp frequency axis closer to perceptual axis:**
  - mel, Bark, constant-Q ...

$$X[c] = \sum_{k=l_c}^{u_c} |S[k]|^2$$



Features (4): Spectral smoothing

- **Generalizing across different speakers is helped by smoothing (i.e. *blurring*) spectrum**
- **Truncated cepstrum is one way:**
  - MSE approx to $\log|S[k]|$
- **LPC modeling is a little different:**
  - MSE approx to $|S[k]| \rightarrow$ prefers detail at peaks



Features (5): Normalization along time

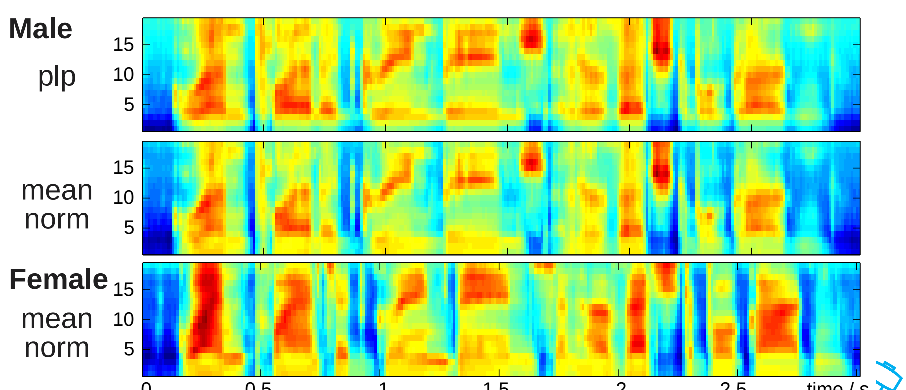
- **Idea: feature variations, not absolute level**
- **Hence: calculate average level & subtract it:**

$$Y[n,k] = X[n,k] - \text{mean}_n\{X[n,k]\}$$

- **Factors out fixed channel frequency response:**

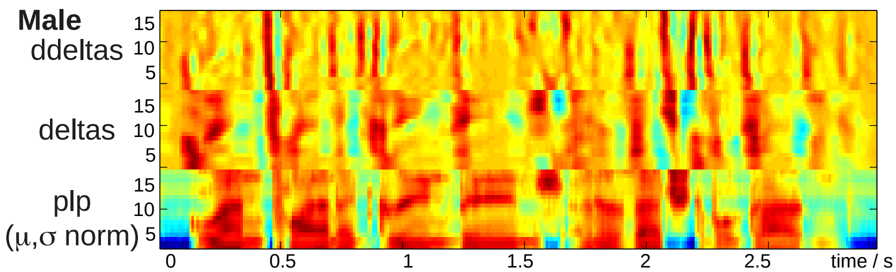
$$s[n] = h_c * e[n]$$

$$\log|S[n,k]| = \log|H_c[k]| + \log|E[n,k]|$$



Delta features

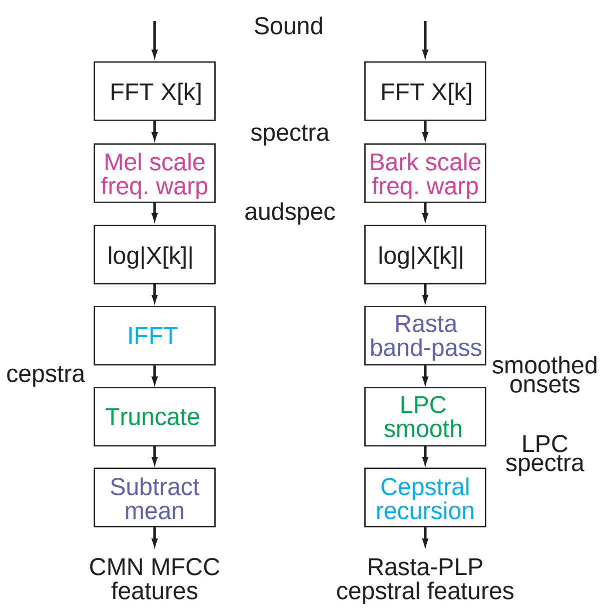
- **Want each segment to have ‘static’ feature vals**
  - but some segments intrinsically dynamic!
  - →calculate their derivatives - maybe steadier?
- **Append** dX/dt **(+** d^2X/dt^2**) to feature vectors**



- **Relates to onset sensitivity in humans?**

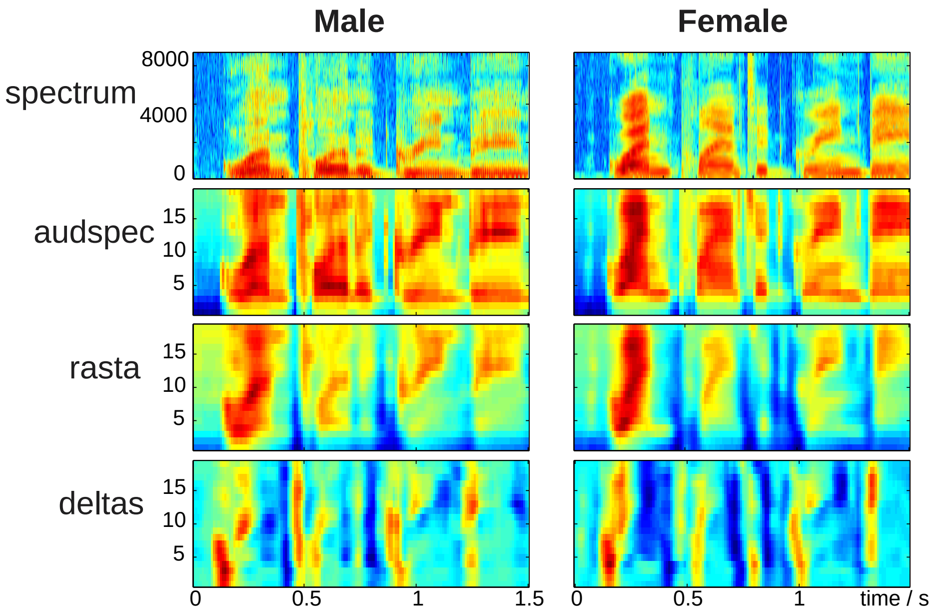
Overall feature calculation

- **MFCCs and/or RASTA-PLP**



- **Key attributes:**
  - spectral, auditory scale
  - decorrelation
  - smoothed (spectral) detail
  - normalization of levels

Features summary



- **Normalize same phones**
- **Contrast different phones**

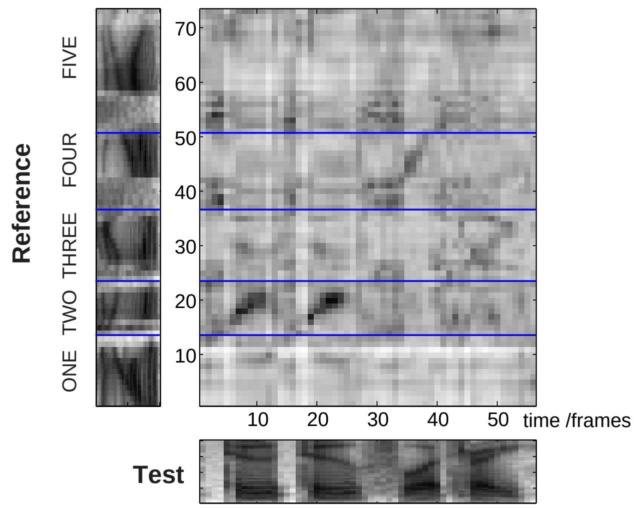
Outline

1. Recognizing Speech
2. Feature Calculation
3. **Sequence Recogntion**
   - Dynamic Time Warp
   - Probabilistic Formulation
4. Hidden Markov Models

Sequence recognition: Dynamic Time Warp (DTW)

3

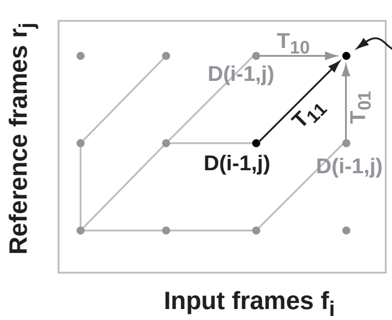
- Framewise comparison with stored templates:



  - distance metric?
  - comparison across templates?

Dynamic Time Warp (2)

- **Find lowest-cost constrained path:**
  - matrix $d(i,j)$ of distances between input frame f_i and reference frame r_j
  - allowable predecessors & transition costs T_{xy}



Lowest cost to (i,j)

$$D(i,j) = d(i,j) + \min\left\{\begin{array}{l} D(i-1,j) + T_{10} \\ D(i,j-1) + T_{01} \\ D(i-1,j-1) + T_{11} \end{array}\right\}$$

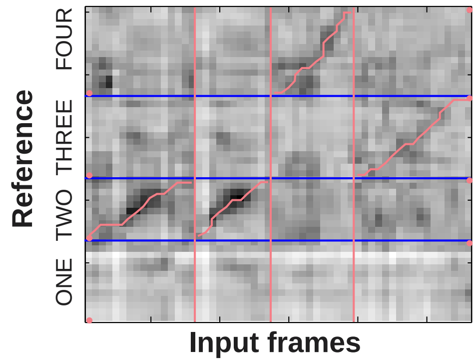
Local match cost

Best predecessor (including transition cost)

- **Best path via traceback from final state**
  - store predecessors for each (i,j)

DTW-based recognition

- **Reference templates for each possible word**
- **For isolated words:**
  - mark endpoints of input word
  - calculate scores through each template (+prune)



  - continuous speech: link together word ends
- **Successfully handles timing variation**
  - recognize speech at reasonable cost

Statistical sequence recognition

- **DTW limited because it's hard to optimize**
  - interpretation of distance, transition costs?
- **Need a theoretical foundation: Probability**
- **Formulate recognition as MAP choice among models:**

$$M^* = \underset{M_j}{\operatorname{argmax}}\ p(M_j | X, \Theta)$$

  - X = observed features
  - M_j = word-sequence models
  - Θ = all current parameters

Statistical formulation (2)

- **Can rearrange via Bayes' rule (& drop $p(X)$):**

$$M^* = \underset{M_j}{\operatorname{argmax}}\ p(M_j | X, \Theta)$$

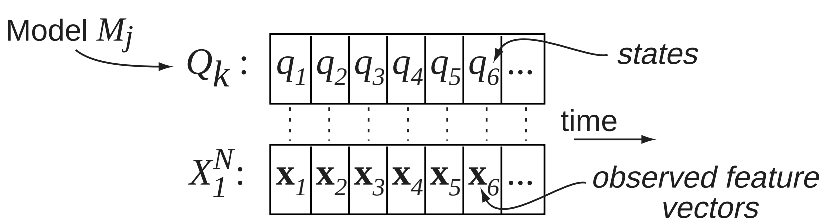
$$= \underset{M_j}{\operatorname{argmax}}\ p(X | M_j, \Theta_A) p(M_j | \Theta_L)$$

  - $p(X \mid M_j)$ = likelihood of observations under model
  - $p(M_j)$ = prior probability of model
  - Θ_A = acoustics-related model parameters
  - Θ_L = language-related model parameters

- **Questions:**
  - what form of model to use for $p(X | M_j, \Theta_A)$?
  - how to find Θ_A (training)?
  - how to solve for M_j (decoding)?

State-based modeling

- **Assume discrete-state model for the speech:**
  - observations are divided up into time frames
  - model → states → observations:



- **Probability of observations given model is:**

$$p(X | M_j) = \sum_{\text{all } Q_k} p(X_1^N | Q_k, M_j) \cdot p(Q_k | M_j)$$

  - sum over all possible state sequences Q_k

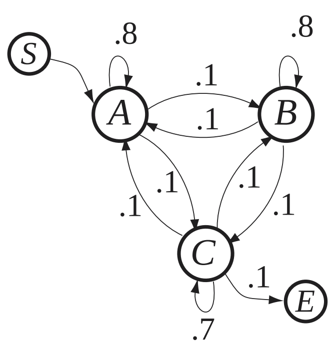
- **How do observations depend on states?**
  How do state sequences depend on model?

Outline

1. Recognizing Speech
2. Feature Calculation
3. Sequence Recognition
4. **Hidden Markov Models (HMM)**
   - generative Markov models
   - hidden Markov models
   - model fit likelihood
   - HMM examples

3 Markov models

- **A (first order) Markov model is a finite-state system whose behavior depends only on the current state**
- **E.g. generative Markov model:**

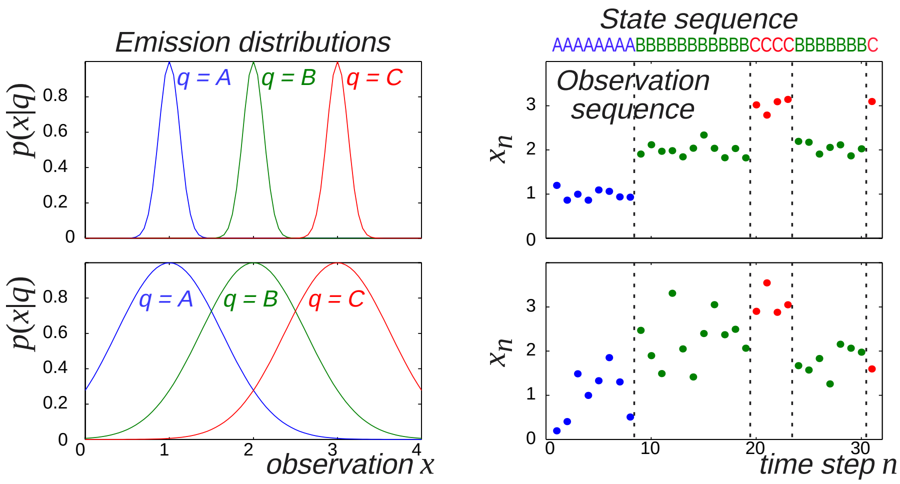


$$
p(q_{n+1}|q_n) \quad
\begin{array}{c|ccccc}
 & & & q_{n+1} & & \\
 & S & A & B & C & E \\
\hline
S & 0 & 1 & 0 & 0 & 0 \\
A & 0 & .8 & .1 & .1 & 0 \\
q_n \;\; B & 0 & .1 & .8 & .1 & 0 \\
C & 0 & .1 & .1 & .7 & .1 \\
E & 0 & 0 & 0 & 0 & 1
\end{array}
$$

S A A A A A A A A B B B B B B B B B C C C C B B B B B B C E

Hidden Markov models

- **= Markov model where state sequence $Q = \{q_n\}$ is not directly observable (= 'hidden')**
- **But, observations X *do* depend on Q:**
  - x_n is rv that depends on current state: $p(x|q)$



  - can still tell *something* about state seq...

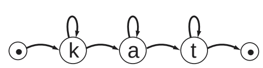
(Generative) Markov models (2)

- **HMM is specified by:**

- states q^i

- transition probabilities a_{ij}

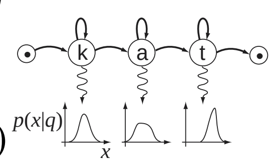
$$p(q_n^j | q_{n-1}^i) \equiv a_{ij}$$



	k	a	t	•
•	1.0	0.0	0.0	0.0
k	0.9	0.1	0.0	0.0
a	0.0	0.9	0.1	0.0
t	0.0	0.0	0.9	0.1

- emission distributions $b_i(x)$

$$p(x|q^i) \equiv b_i(x)$$



+ (initial state probabilities $p(q_1^i) \equiv \pi_i$)

Markov models for sequence recognition

- **Independence of observations:**
  - observation x_n depends only current state q_n

$$p(X|Q) = p(x_1, x_2, \ldots x_N | q_1, q_2, \ldots q_N)$$
$$= p(x_1|q_1) \cdot p(x_2|q_2) \cdot \ldots p(x_N|q_N)$$
$$= \prod_{n=1}^{N} p(x_n|q_n) = \prod_{n=1}^{N} b_{q_n}(x_n)$$

- **Markov transitions:**
  - transition to next state q_{i+1} depends only on q_i

$$p(Q|M) = p(q_1, q_2, \ldots q_N | M)$$
$$= p(q_N|q_1 \ldots q_{N-1}) p(q_{N-1}|q_1 \ldots q_{N-2}) \ldots p(q_2|q_1) p(q_1)$$
$$= p(q_N|q_{N-1}) p(q_{N-1}|q_{N-2}) \ldots p(q_2|q_1) p(q_1)$$
$$= p(q_1) \prod_{n=2}^{N} p(q_n|q_{n-1}) = \pi_{q_1} \prod_{n=2}^{N} a_{q_{n-1} q_n}$$

Model-fit calculation

- **From 'state-based modeling':**

$$p(X|M_j) = \sum_{\text{all } Q_k} p(X_1^N | Q_k, M_j) \cdot p(Q_k|M_j)$$

- **For HMMs:**

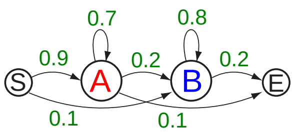
$$p(X|Q) = \prod_{n=1}^{N} b_{q_n}(x_n)$$
$$p(Q|M) = \pi_{q_1} \cdot \prod_{n=2}^{N} a_{q_{n-1} q_n}$$

- **Hence, solve for M^* :**
  - calculate $p(X|M_j)$ for each available model, scale by prior $p(M_j) \rightarrow p(M_j|X)$

- **Sum over *all* Q_k ???**

Summing over all paths

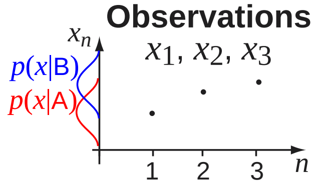
Model M_1



	S	A	B	E
S	•	0.9	0.1	•
A	•	0.7	0.2	0.1
B	•	•	0.8	0.2
E	•	•	•	1

Observations

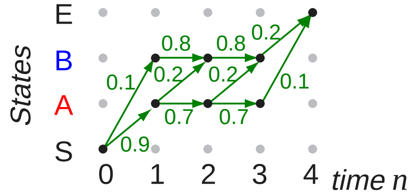
x_1, x_2, x_3



Observation likelihoods

$p(x\|q)$		x_1	x_2	x_3
q	A	2.5	0.2	0.1
	B	0.1	2.2	2.3

Paths



All possible 3-emission paths Q_k from S to E

q_0	q_1	q_2	q_3	q_4	$p(Q \mid M) = \prod_n p(q_n \| q_{n-1})$	$p(X \mid Q,M) = \prod_n p(x_n \| q_n)$	$p(X,Q \mid M)$
S	A	A	A	E	.9 x .7 x .7 x .1 = **0.0441**	2.5 x 0.2 x 0.1 = 0.05	0.0022
S	A	A	B	E	.9 x .7 x .2 x .2 = 0.0252	2.5 x 0.2 x 2.3 = 1.15	0.0290
S	A	B	B	E	.9 x .2 x .8 x .2 = 0.0288	2.5 x 2.2 x 2.3 = 12.65	**0.3643**
S	B	B	B	E	.1 x .8 x .8 x .2 = 0.0128	0.1 x 2.2 x 2.3 = 0.506	0.0065
					Σ = 0.1109		$\boxed{\Sigma = p(X \mid M) = \mathbf{0.4020}}$

(length 3 paths only)

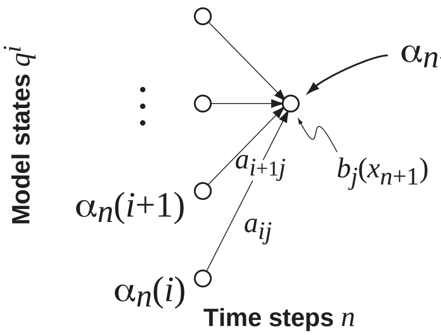
The 'forward recursion'

- **Dynamic-programming-like technique to calculate sum over all Q_k**
- **Define $\alpha_n(i)$ as the probability of getting to state q^i at time step n (by any path):**

$$\alpha_n(i) = p(x_1, x_2, \ldots x_n, q_n = q^i) \equiv p(X_1^n, q_n^i)$$

- **Then $\alpha_{n+1}(j)$ can be calculated recursively:**

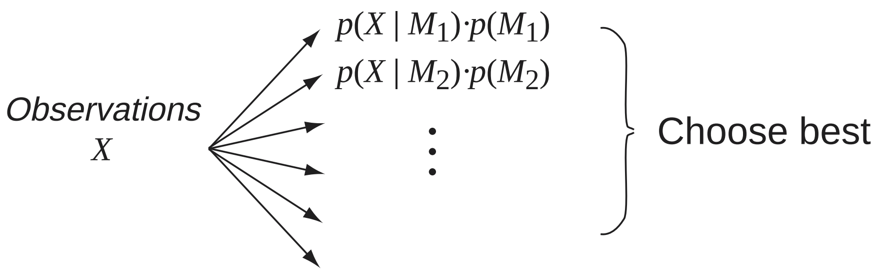
$$\alpha_{n+1}(j) = \left[\sum_{i=1}^{S} \alpha_n(i) \cdot a_{ij}\right] \cdot b_j(x_{n+1})$$



Forward recursion (2)

- **Initialize** $\alpha_1(i) = \pi_i \cdot b_i(x_1)$
- **Then total probability** $p(X_1^N | M) = \sum_{i=1}^{S} \alpha_N(i)$

$\rightarrow$ **Practical way to solve for** $p(X \mid M_j)$
and hence perform recognition



Optimal path

- **May be interested in actual q_n assignments**
  - which state was ‘active’ at each time frame
  - e.g. phone labelling (for training?)
- **Total probability is over *all* paths...**
- **... but can also solve for single *best* path = “Viterbi” state sequence**
- **Probability along best path to state q_{n+1}^j:**

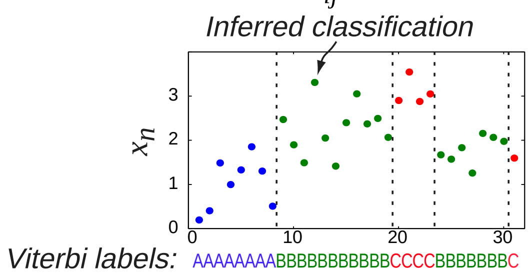
$$\alpha^*_{n+1}(j) = \left[\max_i \left\{ \alpha^*_n(i) a_{ij} \right\} \right] \cdot b_j(x_{n+1})$$

  - backtrack from final state to get best path
  - final probability is product only (no sum) $\rightarrow$ log-domain calculation is just summation
- **Total probability often dominated by best path:**

$$p(X, Q^* | M) \approx p(X | M)$$

Interpreting the Viterbi path

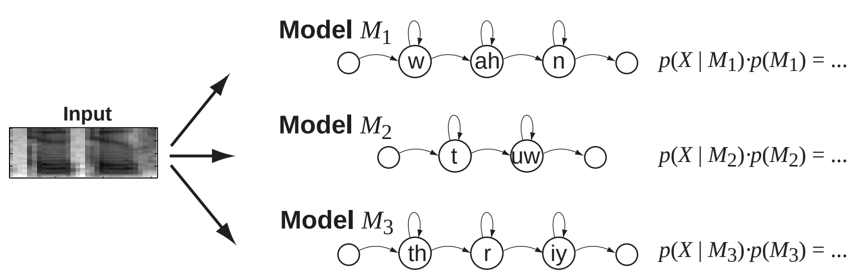
- **Viterbi path assigns each x_n to a state q^i**
  - performing classification based on $b_i(x)$
  - ... at the same time as applying transition constraints a_{ij}



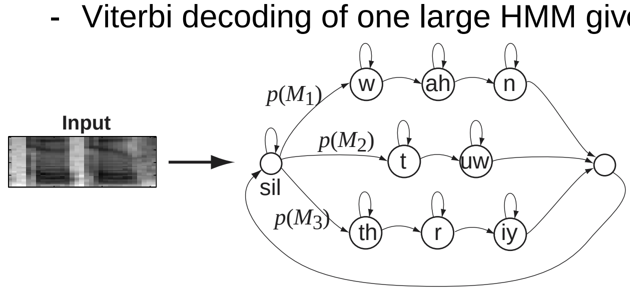
- **Can be used for segmentation**
  - train an HMM with ‘garbage’ and ‘target’ states
  - decode on new data to find ‘targets’, boundaries
- **Can use for (heuristic) training**
  - e.g. train classifiers based on labels...

Recognition with HMMs

- **Isolated word**
  - choose best $p(M|X) \propto p(X|M)p(M)$

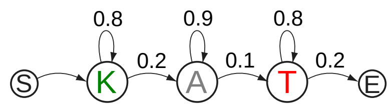


- **Continuous speech**
  - Viterbi decoding of one large HMM gives words

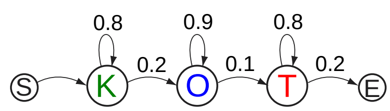


HMM examples: Different state sequences

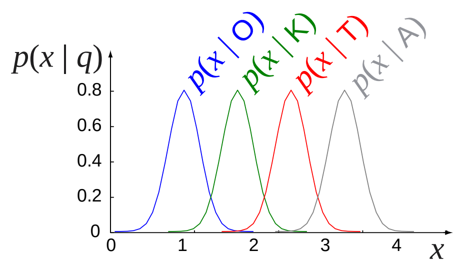
Model M_1



Model M_2

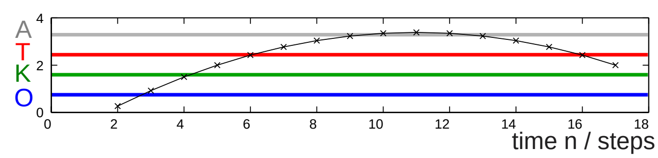


Emission distributions

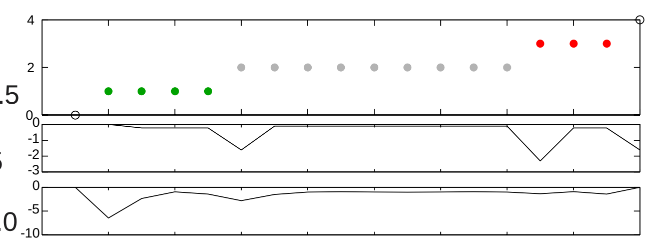


Model matching: Emission probabilities

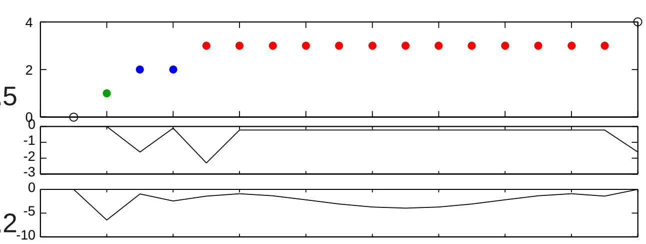
Observation sequence x_n



Model M_1
$\log p(X \mid M) = -32.1$
state alignment
$\log p(X, Q^* \mid M) = -33.5$
log trans.prob
$\log p(Q^* \mid M) = -7.5$
log obs.l'hood
$\log p(X \mid Q^*, M) = -26.0$

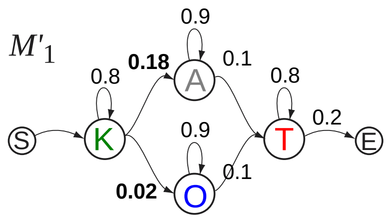


Model M_2
$\log p(X \mid M) = -47.0$
state alignment
$\log p(X, Q^* \mid M) = -47.5$
log trans.prob
$\log p(Q^* \mid M) = -8.3$
log obs.l'hood
$\log p(X \mid Q^*, M) = -39.2$

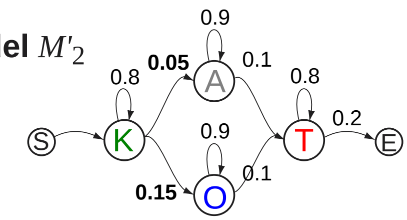


Model matching: Transition probabilities

Model M'_1

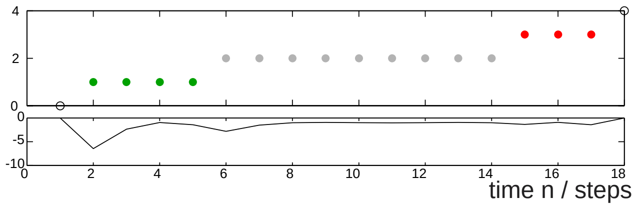


Model M'_2



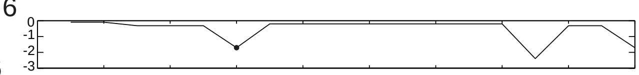
state alignment

log obs.l'hood
$\log p(X \mid Q^*, M) = -26.0$



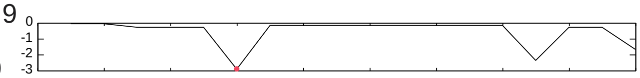
Model M'_1 $\log p(X \mid M) = -32.2$
$\log p(X, Q^* \mid M) = -33.6$

log trans.prob
$\log p(Q^* \mid M) = -7.6$



Model M'_2 $\log p(X \mid M) = -33.5$
$\log p(X, Q^* \mid M) = -34.9$

log trans.prob
$\log p(Q^* \mid M) = -8.9$



Summary

- **Speech signal is highly variable**
  - need models that absorb variability
  - hide what we can with robust features
- **Speech is modeled as a sequence of features**
  - need temporal aspect to recognition
  - best time-alignment of templates = DTW
- **Hidden Markov models are rigorous solution**
  - self-loops allow temporal dilation
  - exact, efficient likelihood calculations

Parting thought:
How to set the HMM parameters? (training)