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

  - WER = \( \frac{(S + D + I)}{N} \)

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

[Diagram showing frequency and time for different microphone positions]
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
    \[ s[n] \rightarrow X_m \quad (m = \frac{n}{H}) \]

  - Acoustic classifier
    calculates probabilities of each mutually-exclusive state \( q^i \)
    \[ p(q^i|X) \]

  - ‘Finite state acceptor’ (i.e. HMM)
    \[ \hat{Q} = \arg\max_{\{q_0, q_1, \ldots, q_L\}} p(q_0, q_1, \ldots, q_L | X_0, X_1 \ldots X_L) \]
    MAP match of allowable sequence to probabilities:
    \[ X \]
    \[ \begin{array}{ccccccc}
      0 & 1 & 2 & \ldots & \ldots & \ldots & X_L \\
    \end{array} \]
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
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:
  \[ X_m[l] = \text{IDFT} \{ \log |S[mH, k]| \} \]
  - 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**:
  \[ X[c] = \sum_{k = l_c}^{u_c} |S[k]|^2 \]
  - mel, Bark, constant-Q ...
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

![Graph of spectral smoothing](image1)

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**:
  $$s[n] = h_c * e[n]$$
  $$\log |S[n, k]| = \log |H_c[k]| + \log |E[n, k]|$$

![Graph of normalization](image2)
**Delta features**

- Want each segment to have ‘static’ feature vals
  - but some segments intrinsically dynamic!
    → calculate their derivatives - maybe steadier?

- Append $\frac{dX}{dt} + \frac{d^2X}{dt^2}$ to feature vectors

- Relates to onset sensitivity in humans?

---

**Overall feature calculation**

- MFCCs and/or RASTA-PLP

Sound spectra

\[ FFT \times[k] \]
\[ \text{Mel scale freq. warp} \]
\[ \log|\times[k]| \]
\[ \text{IFFT} \]
\[ \text{Truncate} \]
\[ \text{Subtract mean} \]
\[ \text{CMN MFCC features} \]
\[ \text{cepstra} \]

\[ FFT \times[k] \]
\[ \text{Bark scale freq. warp} \]
\[ \log|\times[k]| \]
\[ \text{Rasta band-pass} \]
\[ \text{LPC smooth} \]
\[ \text{Cepstral recursion} \]
\[ \text{Rasta-PLP cepstral features} \]

- 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 Recognition
   - Dynamic Time Warp
   - Probabilistic Formulation
4. Hidden Markov Models
3 Sequence recognition: Dynamic Time Warp (DTW)

- Framewise comparison with stored templates:

![Diagram showing sequence recognition using Dynamic Time Warp (DTW) with reference and test frames.]

- 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 $T_{xy}$

![Diagram showing the calculation of the lowest cost to $(i,j)$ and the best predecessor.]

- 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

![Diagram showing DTW-based recognition process](image)

### 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:**
  \[ M^* = \arg\max_{M_j} p(M_j | X, \Theta) \]
  - \( X \) = observed features
  - \( M_j \) = word-sequence models
  - \( \Theta \) = all current parameters
Statistical formulation (2)

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

$$M^* = \arg\max_M p(M_j | X, \Theta)$$

$$= \arg\max_M p(X | M_j, \Theta_A) p(M_j | \Theta_L)$$

- $p(X | M_j) =$ likelihood of observations under model
- $p(M_j) =$ prior probability of model
- $\Theta_A =$ acoustics-related model parameters
- $\Theta_L =$ language-related model parameters

Questions:
- what form of model to use for $p(X | M_j, \Theta_A)$?
- how to find $\Theta_A$ (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 $\rightarrow$ states $\rightarrow$ observations:

$$\text{Model } M_j \rightarrow Q_k : \quad q_1 | q_2 | q_3 | q_4 | q_5 | \ldots$$

$$\text{time} \quad | \quad \text{states}$$

$$\text{observations}$$

$$X_j^N : \quad x_1 | x_2 | x_3 | x_4 | x_5 | \ldots$$

- Probability of observations given model is:

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

- 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

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:

\[
p(q_{n+1}|q_n) = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 \\
0.8 & 0.1 & 0.1 & 0 & 0 \\
0.8 & 0.1 & 0.1 & 0 & 0 \\
0.1 & 0.7 & 0.1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]


\[
q_{n+1}
\]

\[
\begin{array}{c|ccccc}
S & A & B & C & E \\
S & 0 & 1 & 0 & 0 & 0 \\
A & 0.8 & 0.1 & 0.1 & 0 & 0 \\
q_n & B & 0.8 & 0.1 & 0 & 0 \\
C & 0.1 & 0.7 & 0.1 & 0 & 0 \\
E & 0 & 0 & 0 & 0 & 1
\end{array}
\]
**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 \( a_{ij} \)
    
    \[ p(q_n^i | q_{n-1}^j) = a_{ij} \]
  - emission distributions \( b_i(x) \)
    \[ p(x | q^i) = b_i(x) p(x | q) \]
  - (initial state probabilities \( p(q_1^i) = \pi_i \))
Markov models for sequence recognition

- **Independence of observations:**
  - observation $x_n$ depends only current state $q_n$
  
  $$p(X|Q) = p(x_1, x_2, \ldots x_N|q_1, q_2, \ldots q_N)$$
  $$= p(x_1|q_1) \cdot p(x_2|q_2) \cdot \ldots \cdot p(x_N|q_N)$$
  $$= \prod_{n=1}^{N} p(x_n|q_n) = \prod_{n=1}^{N} b_{q_n}(x_n)$$

- **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’:
  
  $$p(X|M_j) = \sum_{all \, Q_k} p(X_1^N|Q_k, M_j) \cdot p(Q_k|M_j)$$

- 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

\[ q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \]

Observations:
\[ x_1, x_2, x_3 \]

Observation likelihoods:
\[ p(x|q) \]
\[ x_1 \quad x_2 \quad x_3 \]
\[ q \{ \begin{array}{ccc}
A & 2.5 & 0.2 & 0.1 \\
B & 0.1 & 2.2 & 2.3 \\
\end{array} \]

Σ = 0.4020

All possible 3-emission paths \( Q_k \) from S to E

\[ q_0 \quad q_1 \quad q_2 \quad q_3 \quad q_4 \]

\[ p(Q|M) = \prod_n p(q_n|q_{n-1}) \]

\[ p(X|Q,M) = \prod_n p(x_n|q_n) \]

<table>
<thead>
<tr>
<th>States</th>
<th>Model M1</th>
<th>Observations</th>
<th>Observation likelihoods</th>
<th>Paths</th>
<th>Time steps n</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>0.9 0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0.7 0.2 0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0.8 0.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Observation likelihoods:
\[ x \mid 1 \quad x \mid 2 \quad x \mid 3 \]

\[ p(x) \]
\[ A \quad 2.5 \quad 0.2 \quad 0.1 \]

Σ = 0.4020

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 \( q^j \) at time step \( n \) (by any path):
  \[ \alpha_n(i) = p(x_1, x_2, \ldots x_n, q_n = q^i) = p(X^n, q^n) \]

- 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 $p(X_1^N | M) = \sum_{i=1}^{S} \alpha_N(i)$

→ Practical way to solve for $p(X | M_j)$
and hence perform recognition

\[
\begin{align*}
\alpha_1(i) &= \pi_i \cdot b_i(x_1) \\
\alpha_n(i) &= \max_j \left\{ \alpha_n(i) a_{ij} \right\} \cdot b_j(x_n+1),
\end{align*}
\]

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 $q_{n+1}^j$:
  \[
  \alpha_{n+1}(j) = \max_i \left\{ \alpha_n(i) a_{ij} \right\} \cdot b_j(x_{n+1}),
  \]
  - backtrack from final state to get best path
  - final probability is product only (no sum)
  → log-domain calculation is just summation
- Total probability often dominated by best path:
  \[
p(X, Q^* | M) = p(X | M)
  \]
Interpreting the Viterbi path

- Viterbi path assigns each $x_n$ to a state $q^i$
  - performing classification based on $b_j(x)$
  - ... 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
Model matching: Transition probabilities

Model $M'_1$

Model $M'_2$

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)*