Speech Processing and Understanding

CSC401 Assignment 3
Agenda

• Background
  • Speech technology, in general
  • Acoustic phonetics

• Assignment 3
  • Speaker Recognition: Gaussian mixture models
  • Speech Recognition: Word-error rates with Levenshtein distance.
Applications of Speech Technology

**Telephony**

**Multimodality & HCI**

**Dictation**

- Buy ticket... AC490... yes
- My hands are in the air.
- Put this there.

**Emerging...**

- Data mining/indexing.
- Assistive technology.
- Conversation.
Formants in sonorants

• However, formants are insufficient features for use in speech recognition generally...
Mel-frequency cepstral coefficients

- In real speech data, the spectrogram is often transformed to a representation that more closely represents human auditory response and is more amenable to accurate classification.

- MFCCs are ‘spectra of spectra’. They are the discrete cosine transform of the logarithms of the nonlinearly Mel-scaled powers of the Fourier transform of windows of the original waveform.
Challenges in speech data

- Co-articulation and dropped phonemes.
- (Intra-and-Inter-) Speaker variability.
- No word boundaries.
- Slurring, disfluency (e.g., ‘um’).
- Signal Noise.
- Highly dimensional.
Phonemes

- Words are formed by **phonemes** (aka ‘phones’),
  e.g., ‘pod’ = /p aa d/

- Words have different pronunciations. and in practice we can never be certain of which phones were uttered, nor their start/stop points.
Phonetic alphabets

• International Phonetic Association (IPA)
  • Can represent sounds in all languages
  • Contains non-ASCII characters

• ARPAbet
  • One of the earliest attempts at encoding English for early speech recognition.

• TIMIT/CMU
  • Very popular among modern databases for speech recognition.
Example phonetic alphabets

<table>
<thead>
<tr>
<th>IPA</th>
<th>CMU</th>
<th>TIMIT</th>
<th>Example</th>
<th>IPA symbol name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ɑ]</td>
<td>AA</td>
<td>a</td>
<td>father, hot</td>
<td>script a</td>
</tr>
<tr>
<td>[æ]</td>
<td>AE</td>
<td>aë</td>
<td>had</td>
<td>digraph</td>
</tr>
<tr>
<td>[ə]</td>
<td>AH0</td>
<td>ax</td>
<td>sofa</td>
<td>schwa (common in unstressed syllables)</td>
</tr>
<tr>
<td>[ʌ]</td>
<td>AH1</td>
<td>ah</td>
<td>but</td>
<td>turned v</td>
</tr>
<tr>
<td>[ɔː]</td>
<td>AO</td>
<td>ao</td>
<td>caught</td>
<td>open o – Note, many speakers of Am. Eng. do not distinguish between [ɔː] and [ɑ]. If your “caught” and “cot” sound the same, you do not.</td>
</tr>
<tr>
<td>[ɛ]</td>
<td>EH</td>
<td>eh</td>
<td>head</td>
<td>epsilon</td>
</tr>
<tr>
<td>[ɪ]</td>
<td>IH</td>
<td>ih</td>
<td>hid</td>
<td>small capital I</td>
</tr>
<tr>
<td>[iː]</td>
<td>IY</td>
<td>iy</td>
<td>heed</td>
<td>lowercase i</td>
</tr>
<tr>
<td>[ʊ]</td>
<td>UH</td>
<td>uh</td>
<td>hood, book</td>
<td>upsilon</td>
</tr>
<tr>
<td>[uː]</td>
<td>UW</td>
<td>uw</td>
<td>boot</td>
<td>lowercase u</td>
</tr>
<tr>
<td>[aɪ]</td>
<td>AY</td>
<td>ay</td>
<td>hide</td>
<td></td>
</tr>
<tr>
<td>[au]</td>
<td>AW</td>
<td>aw</td>
<td>how</td>
<td></td>
</tr>
<tr>
<td>[eɪ]</td>
<td>EY</td>
<td>ey</td>
<td>today</td>
<td></td>
</tr>
<tr>
<td>[ou]</td>
<td>OW</td>
<td>ow</td>
<td>hoed</td>
<td></td>
</tr>
<tr>
<td>[ɔɪ]</td>
<td>OY</td>
<td>oy</td>
<td>joy, ahoy</td>
<td></td>
</tr>
<tr>
<td>[ɛː]</td>
<td>ER0</td>
<td>axr</td>
<td>herself</td>
<td>schwar (schwa changed by following r)</td>
</tr>
<tr>
<td>[ɔː]</td>
<td>ER1</td>
<td>er</td>
<td>bird</td>
<td>reverse epsilon right hook</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IPA</th>
<th>CMU</th>
<th>TIMIT</th>
<th>Example</th>
<th>IPA symbol name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ŋ]</td>
<td>NG</td>
<td>ng</td>
<td>sing song</td>
<td>eng or angma</td>
</tr>
<tr>
<td>[ʃ]</td>
<td>SH</td>
<td>sh</td>
<td>sheet, wish</td>
<td>esh or long s</td>
</tr>
<tr>
<td>[tʃ]</td>
<td>CH</td>
<td>ch</td>
<td>cheese</td>
<td></td>
</tr>
<tr>
<td>[j]</td>
<td>Y</td>
<td>y</td>
<td>yellow</td>
<td>lowercase j</td>
</tr>
<tr>
<td>[ʒ]</td>
<td>ZH</td>
<td>zh</td>
<td>vision</td>
<td>long z or yogh</td>
</tr>
<tr>
<td>[dʒ]</td>
<td>JH</td>
<td>jh</td>
<td>judge</td>
<td></td>
</tr>
<tr>
<td>[ð]</td>
<td>DH</td>
<td>dh</td>
<td>thee, this</td>
<td>eth</td>
</tr>
</tbody>
</table>

- The other consonants are transcribed as you would expect
  - i.e., p, b, m, t, d, n, k, g, s, z, f, v, w, h
Agenda

• Background
  • Speech technology, in general
  • Acoustic phonetics
• Assignment 3
  • Speaker Recognition: Gaussian mixture models
  • Speech Recognition: Word-error rates with Levenshtein distance.
Assignment 3

• Two parts:
  • **Speaker identification**: Determine which of 30 speakers an unknown test sample of speech comes from, given Gaussian mixture models you will train for each speaker.
  • **Speech recognition**: Compute word-error rates for speech recognition systems using Levenshtein distance.
Speaker Data

- 32 speakers (e.g., S-3C, S-5A).
- Each speaker has up to 12 training utterances.
  - e.g., /u/csc401/A3/data/S-3C/0.wav
- Each utterance has 3 files:
  - *.wav: The original wave file.
  - *.mfcc.npy: The MFCC features in NumPy format
  - *.txt: Sentence-level transcription.
Speaker Data (cont.)

- All you need to know: A speech utterance is an Nxd matrix
  - Each row represents the features of a d-dimensional point in time.
  - There are N rows in a sequence of N frames.
  - The data is in numpy arrays *.mfcc.npy
- To read the files: np.load('1.mfcc.npy')

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>frames</td>
<td>X₁[1]</td>
<td>X₁[2]</td>
<td>...</td>
<td>X₁[d]</td>
</tr>
<tr>
<td>1</td>
<td>X₂[1]</td>
<td>X₂[2]</td>
<td>...</td>
<td>X₂[d]</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Note: The table represents the structure of the data matrix with N rows and d columns.
Speaker Data (cont.)

- You are given human transcriptions in transcripts.txt
- You are also given Kaldi and Google transcriptions in transcripts.*.txt.
- Ignore any symbols that are not words.
Agenda

• Background
  • Speech technology, in general
  • Acoustic phonetics

• Assignment 3
  • Speaker Recognition: Gaussian mixture models
  • Speech Recognition: Word-error rates with Levenshtein distance.
Speaker Recognition

• The data is randomly split into training and testing utterances. We don’t know which speaker produced which test utterance.

• Every speaker occupies a characteristic part of the acoustic space.

• We want to learn a probability distribution for each speaker that describes their acoustic behaviour.
  • Use those distributions to identify the speaker-dependent features of some unknown sample of speech data.
Some background: fitting to data

- Given a set of observations $X$ of some random variable, we wish to know how $X$ was generated.
- Here, we assume that the data was sampled from a Gaussian Distribution (validated by data).
- Given a new data point ($x=15$), it is more likely that $x$ was generated by $B$. 
Finding parameters: 1D Gaussians

- Often called *Normal* distributions

\[ p(x) = \frac{\exp \left( -\frac{(x-\mu)^2}{2\sigma^2} \right)}{\sqrt{2\pi}\sigma} \]

\[ \mu = E(x) = \int x p(x) dx \]

\[ \sigma^2 = E((x - \mu)^2) = \int (x - \mu)^2 p(x) dx \]

- The parameters we can adjust to fit the data are \( \mu \) and \( \sigma^2 \):

\[ \theta = \langle \mu, \sigma \rangle \]
Maximum likelihood estimation

- Given data:\[ X = \{x_1, x_2, \ldots, x_n\}\]
- and Parameter set:\[ \theta\]
- Maximum likelihood attempts to find the parameter set that maximizes the likelihood of the data.

\[
L(X, \theta) = p(X \mid \theta) = p(x_1, x_2, \ldots, x_n \mid \theta) = \prod_{i=1}^{n} p(x_i \mid \theta)
\]

- The likelihood function \( L(X, \theta) \) provides a surface over all possible parameterizations. In order to find the Maximum Likelihood, we set the derivative to zero:

\[
\frac{\partial}{\partial \theta} L(X, \theta) = 0
\]
MLE - 1D Gaussian

• Estimate $\hat{\mu}$

$$L(X, \mu) = p(X \mid \mu) = \prod_{i=1}^{n} p(x_i \mid \mu) = \prod_{i=1}^{n} \exp \left( -\frac{(x_i - \mu)^2}{2\sigma^2} \right) \frac{1}{\sqrt{2\pi\sigma}}$$

$$\log L(X, \mu) = -\frac{\sum (x_i - \mu)^2}{2\sigma^2} - n \log \sqrt{2\pi\sigma}$$

$$\frac{\partial}{\partial \mu} \log L(X, \mu) = \frac{\sum (x_i - \mu)}{\sigma^2} = 0$$

$$\hat{\mu} = \frac{\sum x_i}{n}$$

• A similar approach gives the MLE estimate of $\hat{\sigma}^2$ :

$$\hat{\sigma}^2 = \frac{\sum (x_i - \hat{\mu})^2}{n}$$
Multidimensional Gaussians

- When your data is \(d\)-dimensional, the input variable is

\[
\vec{x} = \langle x[1], x[2], \ldots, x[d] \rangle
\]

the mean vector is

\[
\vec{\mu} = E(\vec{x}) = \langle \mu[1], \mu[2], \ldots, \mu[d] \rangle
\]

the covariance matrix is

\[
\Sigma = E((\vec{x} - \vec{\mu})(\vec{x} - \vec{\mu})^T)
\]

with \(\Sigma[i, j] = E(x[i]x[j]) - \mu[i]\mu[j]\)

and

\[
p(\vec{x}) = \frac{\exp \left(\frac{-(\vec{x} - \vec{\mu})^T \Sigma^{-1}(\vec{x} - \vec{\mu})}{2}\right)}{(2\pi)^{d/2} |\Sigma|^{1/2}}
\]
Non-Gaussian data

- Our speaker data does not behave unimodally.
  - i.e., we can't use just 1 Gaussian per speaker.
- E.g., observations below occur mostly bimodally, so fitting 1 Gaussian would not be representative.
Gaussian mixtures

- Gaussian mixtures are a weighted linear combination of $M$ component gaussians.

$$p(\mathbf{x}) = \sum_{j=1}^{M} p(\Gamma_j) p(\mathbf{x} | \Gamma_j)$$
MLE for Gaussian mixtures

- For notational convenience, \( \omega_m = p(\Gamma_m), \ b_m(x^T_t) = p(x^T_t \mid \Gamma_m) \)

- So
  \[
p_{\Theta}(x^T_t) = \sum_{m=1}^{M} \omega_m b_m(x^T_t), \ \Theta = \langle \omega_m, \mu^m, \Sigma_m \rangle, \ m = 1, \ldots, M
  \]

  \[
b_m(x^T_t) = \exp \left( -\frac{1}{2} \sum_{i=1}^{d} \frac{(x^T_t[i] - \mu^m[i])^2}{\sigma^2_m[i]} \right) \left( \frac{(2\pi)^{d/2}}{\prod_{i=1}^{d} \sigma^2_m[i]} \right)^{1/2}
  \]

- To find \( \hat{\Theta} \), we solve \( \nabla_{\Theta} \log L(X, \Theta) = 0 \) where

  \[
  \log L(X, \Theta) = \sum_{t=1}^{N} \log p_{\Theta}(x^T_t) = \sum_{t=1}^{N} \log \left( \sum_{m=1}^{M} \omega_m b_m(x^T_t) \right)
  \]

...see Appendix for more
MLE for Gaussian mixtures (pt. 2)

- Given
  \[
  \frac{\partial \log L(X, \Theta)}{\partial \mu_m[n]} = \sum_{t=1}^{N} \frac{1}{p_\Theta(x_t^\top)} \left[ \frac{\partial}{\partial \mu_m[n]} \omega_m b_m(x_t^\top) \right]
  \]

- Since
  \[
  \frac{\partial}{\partial \mu_m[n]} b_m(x_t^\top) = b_m(x_t^\top) \frac{x_t[n] - \mu_m[n]}{\sigma^2_m[n]}
  \]

- We obtain \(\mu_m[n]\) by solving for \(\mu_m[n]\) in :
  \[
  \frac{\partial \log L(X, \Theta)}{\partial \mu_m[n]} = \sum_{t=1}^{N} \frac{\omega_m}{p_\Theta(x_t^\top)} b_m(x_t^\top) \frac{x_t[n] - \mu_m[n]}{\sigma^2_m[n]} = 0
  \]

and:

\[
\begin{align*}
  b_m(x_t^\top) &= p(x_t^\top | \Gamma_m) \\
p(\Gamma_m | x_t^\top, \Theta) &= \frac{\omega_m}{p_\Theta(x_t^\top)} b_m(x_t^\top) \\
\mu_m[n] &= \frac{\sum_t p(\Gamma_m | x_t^\top, \Theta) x_t[n]}{\sum_t p(\Gamma_m | x_t^\top, \Theta)}
\end{align*}
\]
Recipe for GMM ML estimation

- Do the following for each speaker individually. Use all the frames available in their respective **Training** directories.

1. **Initialize**: Guess $\Theta = \langle \omega_m, \mu_m, \Sigma_m \rangle$, $m = 1, \ldots, M$ with M random vectors in the data, or by performing M-means clustering.

2. **Compute likelihood**: Compute $b_m(x_t)$ and $P(\Gamma_m \mid x_t, \Theta)$

3. **Update parameters**:

   $$\hat{\omega}_m = \frac{1}{T} \sum_{t=1}^{T} p(\Gamma_m \mid x_t, \Theta)$$

   $$\hat{\mu}_m = \frac{\sum_t p(\Gamma_m \mid x_t, \Theta)x_t}{\sum_t p(\Gamma_m \mid x_t, \Theta)}$$

   $$\hat{\Sigma}_m = \frac{\sum_t p(\Gamma_m \mid x_t, \Theta)x_t^2}{\sum_t p(\Gamma_m \mid x_t, \Theta)} - \hat{\mu}_m$$

   $$\log p(X \mid \hat{\Theta}_{i+1}) - \log p(X \mid \hat{\Theta}_i) < \epsilon$$

Repeat 2&3 until converges
Cheat sheet

\[
b_m(\vec{x}_t) = p(\vec{x}_t \mid \Gamma_m)
\]

\[
b_m(\vec{x}_t) = \exp\left(-\frac{1}{2} \sum_{i=1}^{d} \frac{(x_t[i] - \mu_{m[i]})^2}{\sigma_{m[i]}^2}\right)
\left(2\pi\right)^{d/2} \left(\prod_{i=1}^{d} \sigma_{m[i]}^2\right)^{1/2}
\]

Probability of observing \(x_t\) in the \(m^{th}\) Gaussian

\[
\omega_m = p(\Gamma_m)
\]

Prior probability of the \(m^{th}\) Gaussian

\[
p(\Gamma_m \mid \vec{x}_t, \Theta) = \frac{\omega_m}{p_\Theta(\vec{x}_t)} b_m(\vec{x}_t)
\]

Probability of the \(m^{th}\) Gaussian, given \(x_t\)

\[
p_\Theta(\vec{x}_t) = \sum_{m=1}^{M} \omega_m b_m(\vec{x}_t)
\]

Probability of \(x_t\) in the GMM
Initializing theta

\[ \Theta = (\omega_1, \mu_1, \Sigma_1, \omega_2, \mu_2, \Sigma_2, \ldots, \omega_M, \mu_M, \Sigma_M) \]

- Initialize each \( \mu_m \) to a random vector from the data.
- Initialize \( \Sigma_m \) to a `random' diagonal matrix (or identity matrix).
- Initialize \( \omega_m \) randomly, with these constraints:
  \[ 0 \leq \omega_m \leq 1 \]
  \[ \sum_{m} \omega_m = 1 \]
- A good choice would be to set \( \omega_m \) to \( 1/M \)
Over-fitting in Gaussian Mixture Models

• Singularities in likelihood function when a component ‘collapses’ onto a data point:

\[ \mathcal{N}(x_n|x_n, \sigma_j^2 I) = \frac{1}{(2\pi)^{1/2} \sigma_j} \]

then consider \( \sigma_j \to 0 \)

• Likelihood function gets larger as we add more components (and hence parameters) to the model – not clear how to choose the number \( K \) of components

Solutions:
• Ensure that the variances don’t get too small.
• Bayesian GMMs

* Slide borrowed from Chris Bishop’s presentation
Your Task

• For each speaker, train a GMM, using the EM algorithm, assuming diagonal covariance.
• Identify the speaker of each test utterance.
• Experiment with the number of mixture elements in the models, the improvement threshold, number of possible speakers, etc.
• Comment on the results
Practical tips for MLE of GMMs

- We assume diagonal covariance matrices. This reduces the number of parameters and can be sufficient in practice given enough components.

- Numerical Stability: Compute likelihoods in the log domain (especially when calculating the likelihood of a sequence of frames).

$$\log b_m(x_t) = -\sum_{n=1}^{d} \frac{(x_t[n] - \mu_m[n])^2}{2\sigma_m^2[n]} - \frac{d}{2} \log 2\pi - \frac{1}{2} \log \prod_{n=1}^{d} \sigma_m^{-2}[n]$$

- Here, $x_t$, $\mu_m$ and $\sigma_m^{-2}$ are d-dimensional vectors.
Practical tips (pt. 2)

• Efficiency: Pre-compute terms not dependent on $\vec{x}_t$

\[
\log b_m(\vec{x}_t) = - \sum_{n=1}^{d} \left( \frac{1}{2} \vec{x}_t[n]^2 \sigma_m^{-2}[n] - \mu_m[n] \vec{x}_t[n] \sigma_m^{-2}[n] \right) \\
- \left( \sum_{n=1}^{d} \frac{\mu_m[n]^2}{2\sigma_m^2[n]} + \frac{d}{2} \log 2\pi + \frac{1}{2} \log \prod_{n=1}^{d} \sigma_m^2[n] \right)
\]
Agenda

- Background
  - Speech technology, in general
  - Acoustic phonetics
- Assignment 3
  - Speaker Recognition: Gaussian mixture models
  - Speech Recognition: Word-error rates with Levenshtein distance.
Word-error rates

- If somebody said
  REF: how to recognize speech
but an ASR system heard
  HYP: how to wreck a nice beach
how do we measure the error that occurred?

- One measure is \(#\text{CorrectWords}/#\text{HypothesisWords}\)
e.g., 2/6 above

- Another measure is \((S+I+D)/#\text{ReferenceWords}\)
  - S: \# Substitution errors (one word for another)
  - I: \# Insertion errors (extra words)
  - D: \# Deletion errors (words that are missing).
Computing Levenshtein Distance

• In the example
  REF: how to recognize speech.
  HYP: how to wreck a nice beach
  How do we count each of S, I, and D?

• If “wreck” is a substitution error, what about “a” and “nice”?
Computing Levenshtein Distance

• In the example
  REF: how to recognize speech.
  HYP: how to wreck a nice beach
How do we count each of S, I, and D?
If “wreck” is a substitution error, what about “a” and “nice”? 

• **Levenshtein distance:**
  Initialize R[0,0] = 0, and R[i,j] = ∞ for all i=0 or j=0
  for i=1..n (#ReferenceWords)
    for j=1..m (#Hypothesis words)
      R[i,j] = min(R[i-1,j] + 1 (deletion)
      R[i-1,j-1]          (only if words match)
      R[i-1,j-1]+1        (only if words differ)
      R[i,j-1] + 1        ) (insertion)
  
  Return 100*R(n,m)/n
## Levenshtein Example

<table>
<thead>
<tr>
<th></th>
<th>how</th>
<th>to</th>
<th>wreck</th>
<th>a</th>
<th>nice</th>
<th>beach</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>how</td>
<td>∞</td>
<td>0 → 1</td>
<td>2 → 3</td>
<td>4 → 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>∞</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>recognize</td>
<td>∞</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>speech</td>
<td>∞</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Levenshtein example

<table>
<thead>
<tr>
<th></th>
<th>how</th>
<th>to</th>
<th>wreck</th>
<th>a</th>
<th>nice</th>
<th>beach</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>how</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>recognize</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>speech</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Levenshtein example

<table>
<thead>
<tr>
<th></th>
<th>how</th>
<th>to</th>
<th>wreck</th>
<th>a</th>
<th>nice</th>
<th>beach</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>how</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>to</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>recognize</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>speech</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Levenshtein example

<table>
<thead>
<tr>
<th></th>
<th>how</th>
<th>to</th>
<th>wreck</th>
<th>a</th>
<th>nice</th>
<th>beach</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>word</th>
<th>how</th>
<th>to</th>
<th>wreck</th>
<th>a</th>
<th>nice</th>
<th>beach</th>
</tr>
</thead>
<tbody>
<tr>
<td>how</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>to</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>recognize</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>speech</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Word-error rate is $4/4 = 100\%$

2 substitutions, 2 insertions
Appendices
Multidimensional Gaussians, pt. 2

- If the $i^{th}$ and $j^{th}$ dimensions are statistically independent,
  \[
  E(x[i]x[j]) = E(x[i])E(x[j])
  \]
  and
  \[
  \Sigma[i, j] = 0
  \]

- If all dimensions are statistically independent, $\Sigma[i, j] = 0$, $\forall i \neq j$ and the covariance matrix becomes diagonal, which means
  \[
  p(\bar{x}) = \prod_{i=1}^{d} p(x[i])
  \]
  where
  \[
  p(x[i]) \sim N(\mu[i], \Sigma[i, i])
  \]
  \[
  \Sigma[i, i] = \sigma^2[i]
  \]
MLE example - dD Gaussians

- The MLE estimates for parameters $\Theta = \langle \theta_1, \theta_2, \ldots, \theta_d \rangle$ given i.i.d. training data $X = \langle \vec{x}_1, \ldots, \vec{x}_n \rangle$ are obtained by maximizing the joint likelihood

$$L(X, \Theta) = p(X \mid \Theta) = p(\vec{x}_1, \ldots, \vec{x}_n \mid \Theta) = \prod_{i=1}^{n} p(\vec{x}_i \mid \Theta)$$

- To do so, we solve $\nabla_\Theta L(X, \Theta) = 0$ where

$$\nabla_\Theta = \left\langle \frac{\partial}{\partial \theta_1}, \ldots, \frac{\partial}{\partial \theta_d} \right\rangle$$

- Giving

$$\hat{\mu} = \frac{\sum_{t=1}^{n} \vec{x}_t}{n} \quad \hat{\Sigma} = \frac{\sum_{t=1}^{n} (\vec{x}_t - \hat{\mu}) (\vec{x}_t - \hat{\mu})^T}{n}$$
MLE for Gaussian mixtures (pt1.5)

- Given $\log L(X, \Theta) = \sum_{t=1}^{N} \log p_{\Theta}(\vec{x}_t)$ and $p_{\Theta}(\vec{x}_t) = \sum_{m=1}^{M} \omega_m b_m(\vec{x}_t)$

- Obtain an ML estimate, $\mu_m \hat{\rightarrow}$, of the mean vector by maximizing $\log L(X, \mu_m)$ w.r.t. $\mu_m[n]$

\[
\frac{\partial \log L(X, \Theta)}{\partial \mu_m[n]} = \sum_{t=1}^{N} \frac{\partial}{\partial \mu_m[n]} \log p_{\Theta}(\vec{x}_t) = \sum_{t=1}^{N} \frac{1}{p_{\Theta}(\vec{x}_t)} \left[ \frac{\partial}{\partial \mu_m[n]} \omega_m b_m(\vec{x}_t) \right]
\]

- Why?
  - d of sum = sum of d
  - d rule for $\log_e$
  - d wrt $\mu_m$ is 0 for all other mixtures in the sum in $p_{\Theta}(\vec{x}_t)$