automatic speech recognition
This lecture

• Mel-frequency cepstral coefficients (MFCCs),
• Gaussian mixture models,
• Acoustic modelling, and
• Clustering.
The Mel-scale filter bank

• To *mimic* the response of the *human ear* (and because it empirically improves speech recognition), we often discretize the spectrum using $M$ filters.
  • Uniform spacing before 1 kHz, logarithmic after 1 kHz
Source and filter

- The acoustics of speech are produced by a glottal pulse waveform (the source) passing through a vocal tract whose shape modifies that wave (the filter).

- The shape of the vocal tract is more important to phoneme recognition.
  - We to separate the source from the filter in the acoustics.
Source and filter

• Since speech is assumed to be the output of a linear time invariant system, it can be written as the **convolution** of the input and an impulse response.
  • For speech signal $x[n]$, glottal signal $g[n]$, and vocal tract transfer $v[n]$ with spectra $X[z]$, $G[z]$, and $V[z]$, respectively, this means
    $$x[n] = g[n] \ast v[n]$$
    $$X[z] = G[z]V[z]$$
    $$\log X[z] = \log G[z] + \log V[z]$$

• **Convolution**, $x \ast y$, is beyond the scope of this course. It is basically a modification of one signal by another.
The cepstrum

• We separate the source and the filter by representing the log of the spectrum as if it were a time domain signal.
  • the log of the spectrum $\log X[z]$ is a sum of the source and the filter log spectra, i.e., a superposition – finding its spectrum will allow us to extract these components.

• Cepstrum: \textit{n.} the spectrum of the log of the spectrum.
  • Fun fact: ‘ceps’ is the reverse of ‘spec’.
The cepstrum

• The domain of the cepstrum is **quefrency** (a play on the word ‘frequency’).
The cepstrum

This is due to the vocal tract shape

This is due to the glottis

Pictures from John Coleman (2005)
Mel-frequency cepstral coefficients (MFCCs) are the most popular representation of speech used in ASR. They are the spectra of the logarithms of the mel-scaled filtered spectra of the windows of the waveform.
Advantages of MFCCs

- The cepstrum produces **highly uncorrelated features** (every dimension is useful).
  - This includes a **separation** of the **source** and **filter**.

- In practice, the cepstrum is **easier to learn** than the spectrum for phoneme recognition.

- Aside: There is an efficient method to compute cepstra called the **discrete cosine transform**.
MFCCs in practice

- An observation vector of MFCCs often consists of:
  - The **first 13 cepstral coefficients** (i.e., the first 13 dimensions produced by this method),
  - An additional **overall energy** measure,
  - The **velocities** ($\delta$) of each of those 14 dimensions,
    - (i.e., the slope of each dimension at a given time),
    - The **accelerations** ($\delta\delta$) of each of those 14 dimensions.
- The result is that at a timeframe $t$ we have an observation MFCC vector of $(13+1)\times3=42$ dimensions.
  - This vector is what is used by our ASR systems...
Classifying speech sounds

- Speech sounds tend to cluster. This graph shows vowels, each in their own colour, according to the 1st two formants.
Classifying speakers

- Similarly, all of the speech produced by one speaker will cluster differently in **MFCC space** than speech from another speaker.
- We can decide if a given observation comes from one speaker or another.

\[
P(\text{one speaker}) > P(\text{another speaker})
\]

<table>
<thead>
<tr>
<th>MFCC</th>
<th>Time, ( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 0 ) 1 ( \ldots ) ( T )</td>
</tr>
<tr>
<td>2</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>3</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots ) ( \ldots ) ( \ldots ) ( \ldots )</td>
</tr>
<tr>
<td>42</td>
<td>( \ldots )</td>
</tr>
</tbody>
</table>
Fitting continuous distributions

• In the past, we used discrete probability functions.
• Since we are now operating with continuous variables, we need to fit continuous probability functions to a discrete number of observations.

• If we assume the 1-dimensional data in this histogram is Normally distributed, we can fit a continuous Gaussian function simply in terms of the mean $\mu$ and variance $\sigma^2$. 
Comparing continuous distributions

- Moreover, if we observe a particular value in this univariate space, e.g., \( x = 15 \), we can say which of several distributions is most likely to have produced it.
- Here, distribution B is more likely to have produced \( x = 15 \) because \( P(x; B) > P(x; A) \).
Good fits

• Given some fixed **training data**, we want to be able to fit continuous probability functions that **best match** our observations.

• The data in **this histogram** is **more likely** to have been produced from the parameterization on the **left**.
Univariate (1D) Gaussians

• Often called **Normal** distributions, $N(\mu, \sigma)$

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

• The parameters we can modify are $\theta = \langle \mu, \sigma^2 \rangle$
  
  • $\mu = E(x) = \int x \cdot P(x) \, dx$ (**mean**)
  
  • $\sigma^2 = E\left((x - \mu)^2\right) = \int (x - \mu)^2 P(x) \, dx$ (**variance**)

Maximum likelihood estimation

• Given data $X = \{x_1, x_2, ..., x_n\}$, MLE produces an estimate of the parameters $\hat{\theta}$ by maximizing the likelihood, $L(X, \theta)$:
  $$\hat{\theta} = \text{argmax}_\theta L(X, \theta)$$
  where $L(X, \theta) = P(X; \theta) = \prod_{i=1}^n P(x_i; \theta)$.

• Since $L(X, \theta)$ provides a surface over all $\theta$, in order to find the highest likelihood, we look at the derivative
  $$\frac{\delta}{\delta \theta} L(X, \theta) = 0$$
  to see at which point the likelihood stops growing.
MLE with univariate Gaussians

• Estimate $\mu$:

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

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

$$\frac{\delta}{\delta \mu} \log L(X, \mu) = \frac{\sum_{i}(x_i - \mu)}{\sigma^2} = 0$$

$$\mu = \frac{\sum_{i} x_i}{n}$$

• Similarly, $\sigma^2 = \frac{\sum_{i}(x_i - \mu)^2}{n}$
Multivariate Gaussians

• When data is \( d \)-dimensional, the input variable is
  \[ \mathbf{x} = \langle x[1], x[2], \ldots, x[d] \rangle \]
  the mean is
  \[ \mathbf{\mu} = E(\mathbf{x}) = \langle \mu[1], \mu[2], \ldots, \mu[d] \rangle \]
  the covariance matrix is
  \[ \Sigma[i, j] = E(x[i]x[j]) - \mu[i]\mu[j] \]
  and
  \[
  P(\mathbf{x}) = \frac{\exp\left(-\frac{(\mathbf{x} - \mathbf{\mu})^\top \Sigma^{-1}(\mathbf{x} - \mathbf{\mu})}{2}\right)}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}}
  \]
  \( A^\top \) is the transpose of \( A \)
  \( A^{-1} \) is the inverse of \( A \)
  \(|A|\) is the determinant of \( A \)
Intuitions of covariance

- As values in $\Sigma$ become larger, the Gaussian spreads out.
- ($I$ is the identity matrix – 0 except for 1s on the diagonal)
Intuitions of covariance

• Different values on the diagonal result in different variances in their respective dimensions
Non-Gaussian observations

- Speech data is generally *not* unimodal – it’s more complex.
- The observations below are **bimodal**, so fitting one Gaussian would not be representative.
  - E.g., if you usually keep your keys in your desk or on your table, it makes no sense looking for them floating in the air midway between the two.
Mixtures of Gaussians

- Gaussian mixture models (GMMs) are a weighted linear combination of \( M \) component Gaussians, \( \langle \Gamma_1, \Gamma_2, \ldots, \Gamma_M \rangle \):

\[
P(\tilde{x}) = \sum_{j=1}^{M} P(\Gamma_j)P(\tilde{x}|\Gamma_j)
\]
Observation likelihoods

• If the covariance matrix is **diagonal** (it often is for simplicity), then the probability of an observation vector given a Gaussian from slide 11 becomes

\[
P(\mathbf{x}|\Gamma_m) = \exp\left(-\frac{1}{2} \sum_{i=1}^{d} \frac{(x[i] - \mu_m[i])^2}{\Sigma_m[i]}\right) \frac{1}{(2\pi)^{\frac{d}{2}} \left(\prod_{i=1}^{d} \Sigma_m[i]\right)^{\frac{1}{2}}}
\]

• If the MFCC dimensions are independent of one another, the covariance matrix will be diagonal – i.e., 0 off the diagonal.
Mixtures of Gaussians

- If we knew which Gaussian generated each sample, we could learn $P(\Gamma_j)$ with MLE, but that data is hidden, so we must use...

$$P(\tilde{x}) = \sum_{j=1}^{M} P(\Gamma_j)P(\tilde{x}|\Gamma_j)$$
Expectation-Maximization for GMMs

• If $\omega_m = P(\Gamma_m)$ and $b_m(x_t) = P(x_t | \Gamma_m)$,

$$P_\theta(x_t) = \sum_{m=1}^{M} \omega_m b_m(x_t)$$

where $\theta = \langle \omega_m, \mu_m, \Sigma_m \rangle$ for $m = 1..M$

• To estimate $\theta$, we solve $\nabla_\theta \log L(X, \theta) = 0$ where

$$\log L(X, \theta) = \sum_{t=1}^{T} \log P_\theta(x_t) = \sum_{t=1}^{T} \log \sum_{m=1}^{M} \omega_m b_m(x_t)$$
Expectation-Maximization for GMMs

• We differentiate the log likelihood function w.r.t. \( \mu_m[n] \) and set this to 0 to find the value of \( \mu_m[n] \) at which the likelihood stops growing.

\[
\frac{\delta \log L(X, \theta)}{\delta \mu_m[n]} = \sum_{t=1}^{N} \frac{1}{P_{\theta}(\bar{x}_t)} \left[ \frac{\delta}{\delta \mu_m[n]} \omega_m b_m(\bar{x}_t) \right] = 0
\]

• Aside: why does the differential at 0 give us the point at which \( L(X, \theta) \) stops growing (rather than stops shrinking)?
  • The partial derivative of the continuous likelihood function is assumed to be 0 only at \( \pm \infty \), which cannot be found analytically.
Expectation-Maximization for GMMs

- The **expectation step** gives us:

\[ b_m(x_t) = P(x_t | \Gamma_m) \]

\[ P(\Gamma_m | x_t; \theta) = \omega_m \frac{P(\theta(x_t))}{P(\theta(x_t))} b_m(x_t) \]

- The **maximization step** gives us:

\[ \widehat{\mu_m} = \frac{\sum_t P(\Gamma_m | x_t; \theta) x_t}{\sum_t P(\Gamma_m | x_t; \theta)} \]

\[ \widehat{\Sigma_m} = \frac{\sum_t P(\Gamma_m | x_t; \theta) x_t^2}{\sum_t P(\Gamma_m | x_t; \theta)} - \widehat{\mu_m}^2 \]

\[ \omega_m \approx \frac{1}{T} \sum_{t=1}^{T} P(\Gamma_m | x_t; \theta) \]
Some specificity...

• In the previous slide, the square of a vector, $\vec{a}^2$, is elementwise (i.e., \(v=a . ^2\) in Matlab)
  • E.g., $[2 \ 3 \ 4]^2 = [4 \ 9 \ 16]$

• Since $\Sigma$ is diagonal, it can be represented as a vector.

• Can $\sigma_m^2 = \frac{\sum_t P(\Gamma_m|\vec{x}_t; \theta)\vec{x}_t^2}{\sum_t P(\Gamma_m|\vec{x}_t; \theta)} - \vec{\mu}_m^2$ become negative?
  • No.
    • This is left as an exercise, but only if you’re interested.
Speaker recognition

- **Speaker recognition**: *n.* the identification of a speaker among several speakers given only some acoustics.

- Each **speaker** will produce speech according to **different** probability distributions.
  - We train a **Gaussian mixture model** for each speaker, given annotated data (mapping utterances to speakers).
  - A given sound is more likely to come from one of these speakers.
Recipe for GMM EM

• For each speaker, we learn a GMM given all $T$ frames of their training data.

1. Initialize: Guess $\theta = \langle \omega_m, \mu_m, \Sigma_m \rangle$ for $m = 1..M$ either uniformly, randomly, or by $k$-means clustering.

2. E-step: Compute $b_m(x_t)$ and $P(\Gamma_m | \overline{x_t}; \theta)$.

3. M-step: Update parameters for $\langle \omega_m, \mu_m, \Sigma_m \rangle$ as described on slide 20.

• (see the Reynolds & Rose (1995) paper on the course webpage for details)
Clustering

- **Quantization** involves turning possibly **multi-variate** and **continuous** representations into **univariate discrete** symbol.
  - Reduced storage and computation costs.
  - Potentially tremendous loss of information.

- Observation $X$ is in Cluster One, so we replace it with 1.

- Clustering is **unsupervised** learning.
  - Number and form of clusters often unknown.
Clustering

• What defines a particular cluster?
  • Is there some prototype representing each cluster?

• What defines membership in a cluster?
  • Usually, some distance metric $d(x, y)$ (e.g., Euclidean distance).

• How well do clusters represent unseen data?
  • How is a new point assigned to a cluster?
  • How do we modify that cluster as a result?
**K-means clustering**

- Used to group data into \( K \) clusters, \( \{C_1, \ldots, C_K\} \).

- Each cluster is represented by the mean of its assigned data.
  - (sometimes it’s called the cluster’s centroid).

- Iterative algorithm converges to local optimum:
  1. **Select** \( K \) initial cluster means \( \{\mu_1, \ldots, \mu_K\} \).
  2. **Until** (stopping criterion),
     a) **Assign** each data sample to closest cluster
        \[ x \in C_i \quad \text{if} \quad d(x, \mu_i) \leq d(x, \mu_j), \quad \forall i \neq j \]
     b) **Update** \( K \) means from assigned samples
        \[ \mu_i = E(x) \quad \forall x \in C_i, \quad 1 \leq i \leq K \]
K-means example ($K = 3$)

- Initialize with a random selection of 3 data samples.
- Euclidean distance metric $d(x, \mu)$
**K-means stopping condition**

- The total **distortion**, $\mathcal{D}$, is the sum of squared error,

$$
\mathcal{D} = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2
$$

- $\mathcal{D}$ decreases between $n^{th}$ and $(n + 1)^{th}$ iteration.

- We can stop training when $\mathcal{D}$ falls below some threshold $\mathcal{T}$.

$$
1 - \frac{\mathcal{D}(n + 1)}{\mathcal{D}(n)} < \mathcal{T}
$$
Acoustic clustering example

• 12 clusters of spectra, after training.
Number of clusters

• The number of true clusters is unknown.
• We can iterate through various values of $K$.
  • As $K$ approaches the size of the data, $D$ approaches 0...

\[ K = 2 \]
\[ K = 4 \]
Hierarchical clustering

- Hierarchical clustering clusters data into hierarchical ‘class’ structure.
- Two types: top-down (divisive) or bottom-up (agglomerative).
- Often based on greedy formulations.
- Hierarchical structure can be used for hypothesizing classes.
Divisive clustering

- Creates hierarchy by successively splitting clusters into smaller groups.
Agglomerative clustering

• **Agglomerative clustering** starts with $N$ ‘seed’ clusters and iteratively combines these into a hierarchy.

• On each iteration, the two most similar clusters are **merged** together to form a new **meta-cluster**.

• After $N - 1$ iterations, the hierarchy is complete.

• Often, when the similarity scores of new meta-clusters are tracked, the resulting graph (i.e., **dendogram**) can yield insight into the natural grouping of data.
Dendogram example
Stepwise-optimal clustering

• Agglomerative clustering can create clusters with **high distortion** (distance of data to ‘means’).
• **Stepwise-optimal clustering** merges the two clusters that result in the **smallest increase** in distortion on each iteration.
  • The metric for minimizing distortion, given that class \( C_i \) has \( n_i \) tokens is
    \[
    \sqrt{\frac{n_i n_j}{n_i + n_j}} \| \mu_i - \mu_j \|
    \]
  • This tends to combine small clusters with large clusters before merging clusters of similar sizes.
Speaker clustering

- 23 female and 53 male speakers from TIMIT.
- Data are vectors of average F1 and F2 for 9 vowels.
- Distance $d(C_i, C_j)$ is average of distances between members.
Acoustic-phonetic hierarchy

(this is basically an upside-down dendogram)
Word clustering
Next week

• Automatic speech recognition
  • ...with HMMS.

• Information Retrieval, part 1.