Speech Features and Speaker Classfication

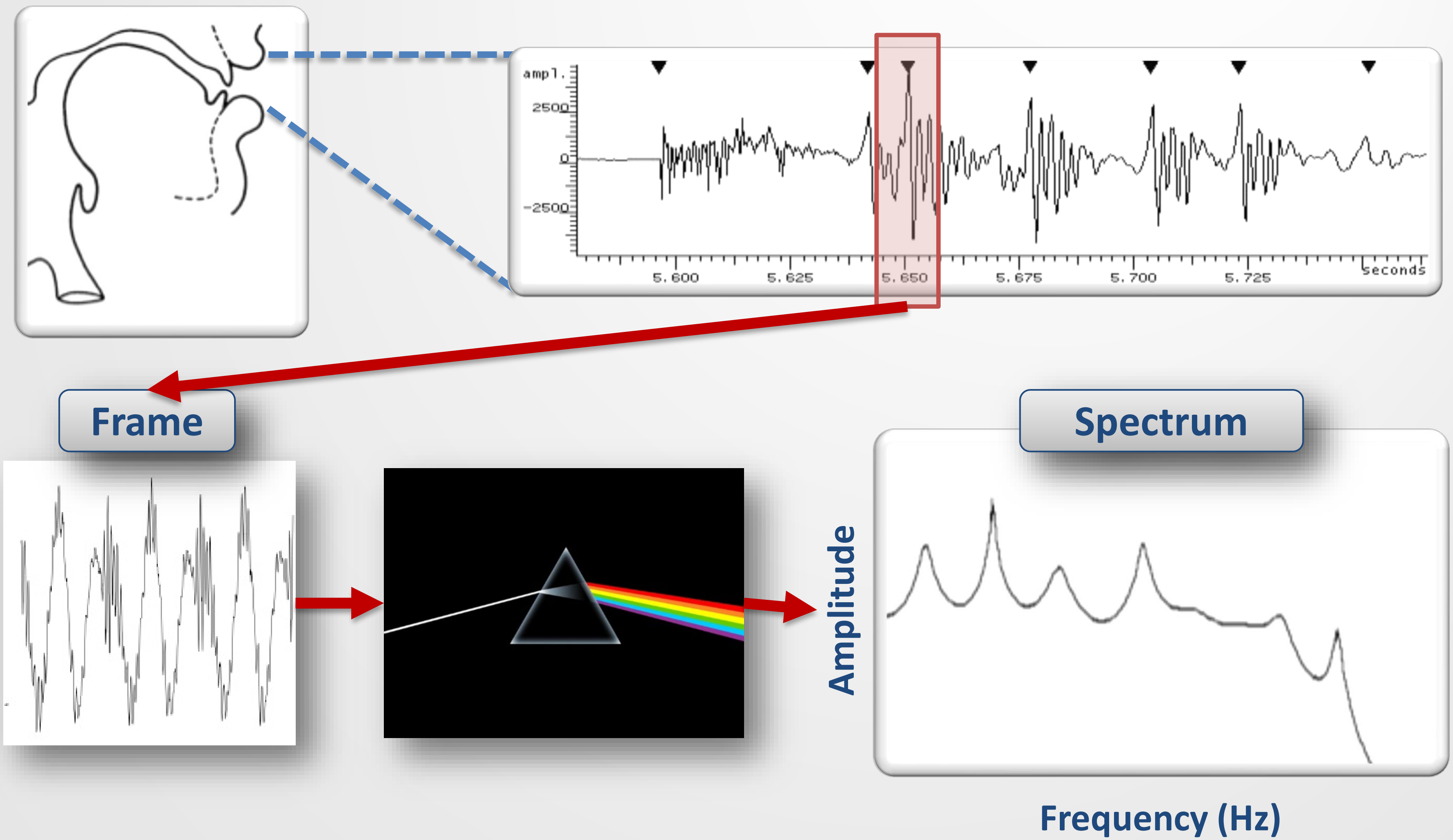
CSC401/2511 – Natural Language Computing – Winter 2024
Lecture 9
University of Toronto

Contents

- Today we will
  - Define some common feature vectors for speech processing
  - Use them as input to a GMM-based speaker classification system
- All of this is part of A3

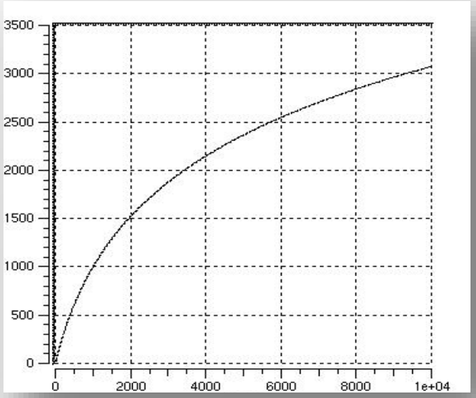
SPEECH FEATURES

Recall the spectrogram pipeline



Problems with spectrograms

- As input to speech systems, spectrograms are…
- **Too big**
  - The discrete signal is usually 16,000 samps/sec
  - 100 frames/sec x 400 samps/frame = 40,000 samps/sec!
- **Too linear**
  - Pitch perception is log-linear (recall Mels)
  - Lots of coefficients wasted on high frequencies
- **Too entangled**
  - Speaker and phoneme info is correlated



UNIVERSITY OF TORONTO

Filtering

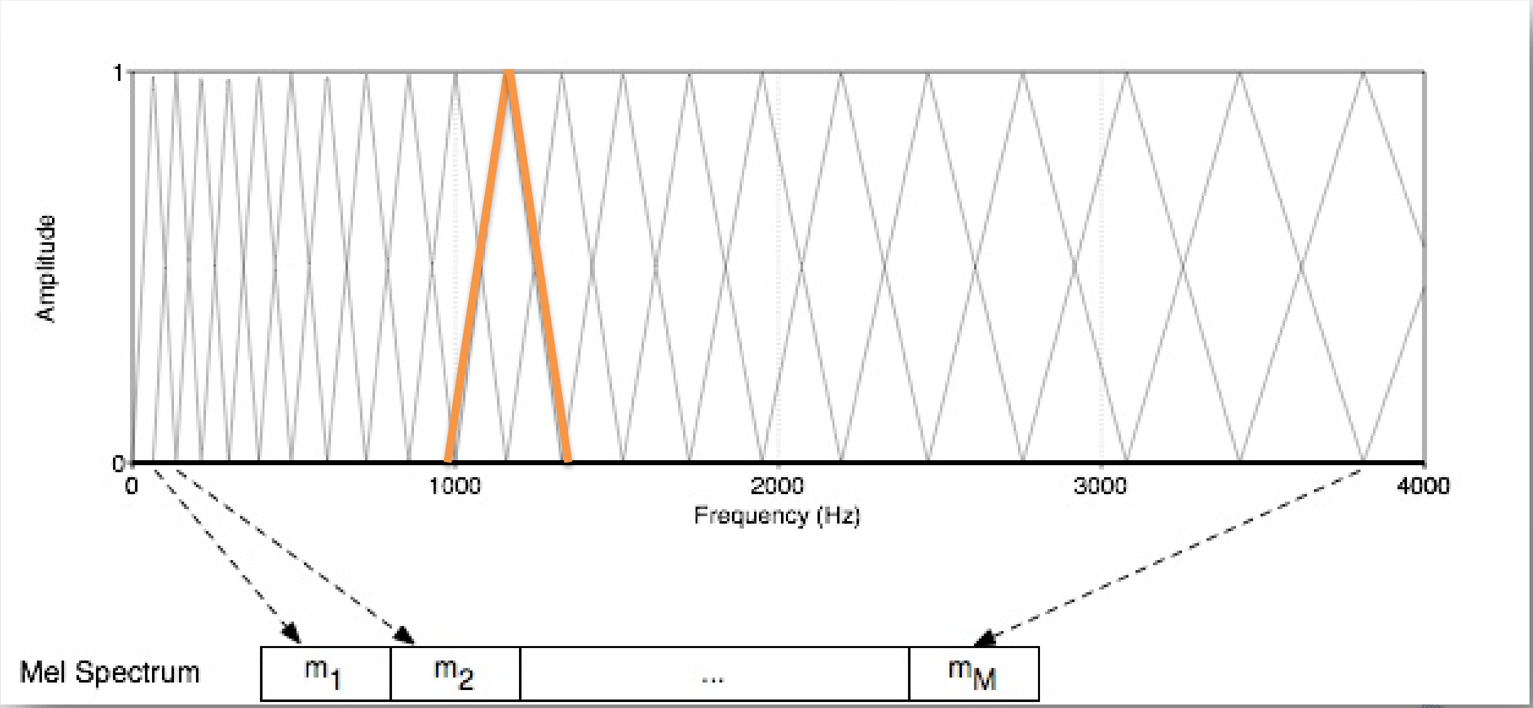
- To reduce the size of the spectra, we **filter** it with **filters** from a **filter bank**
- Each filter is a signal whose spectrum $F_m \in \mathbb{R}^N$ picks out small a range (or **band**) of frequencies
- The bands of the M filters are overlapping and span the spectrum
- A **filter coefficient** is computed as the **log** of the dot product of the **magnitude** of the frame X_t and filter F_m spectra:

  $$c_{t,m} = \log \sum_{n=1}^{N} |X_t|[n]|F_m|[n]$$
- If there are T frames, this gives us a real-valued feature matrix of size $T \times M$
  - $M = 40$ is a lot smaller than 400!

UNIVERSITY OF TORONTO

The mel-scale filter bank

- The mel-scale triangular overlapping filter bank, or **f-bank**, is a popular choice
- The filter's vertices are arranged along the mel-scale
  - Ascending frequency = wider bands



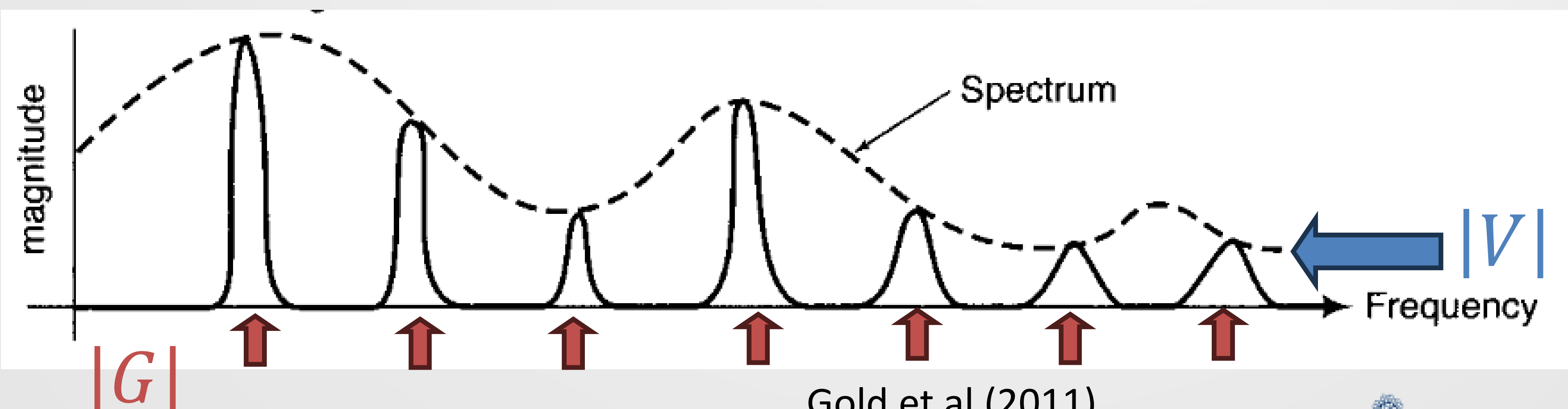
UNIVERSITY OF TORONTO

The source-filter model

- In vowels, the sound signal emitted from the glottis g is filtered by the vocal tract v
- The **source-filter model** of speech assumes

$$|X[n]| = |G[n]||V[n]|$$

- $|V|$ is responsible for the smooth shape (envelope)
- $|G|$ is responsible for all the bumps (F0 harmonics)



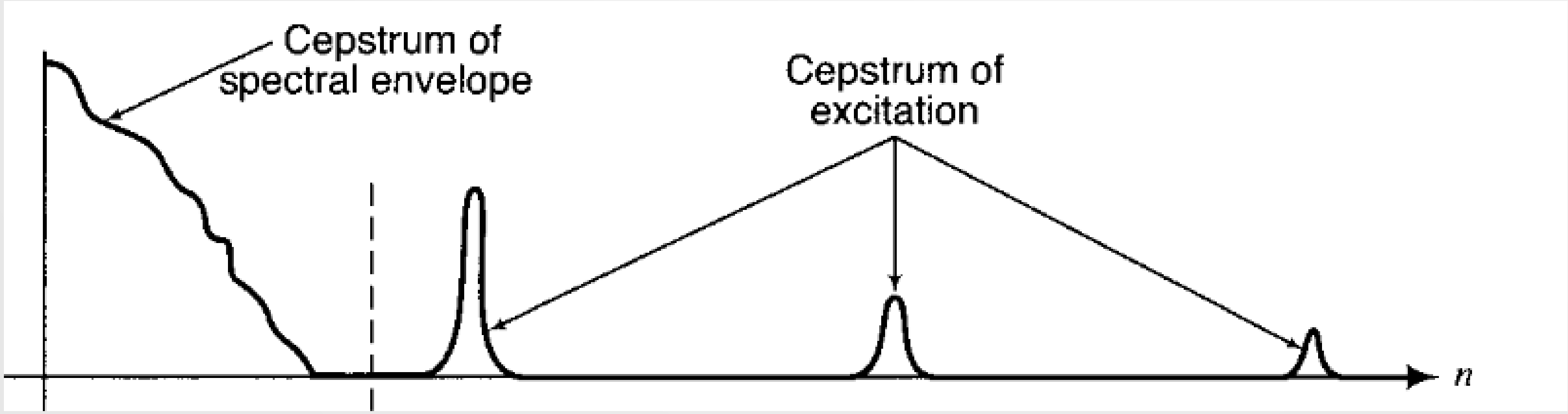
Gold et al (2011)

The cepstrum

- We can get at $|V|$ by computing the **cepstrum** $\hat{x}$
- The cepstrum is $\log|X|$ transformed by the inverse DFT
- Because $\log|X| = \log|G| + \log|V|$, and DFT^{-1} is linear

$$\hat{x}[n] = \hat{g}[n] + \hat{v}[n]$$

- $DFT^{-1} \approx DFT$, so $\hat{x}$ is like the spectrum of $\log|X|$
- $|V|$ is slower-moving than $|G|$, so $\hat{v}[n]$ is higher for lower n (lower frequency of frequency)



Gold et al (2011)

Mel-Frequency Cepstral Coefficients

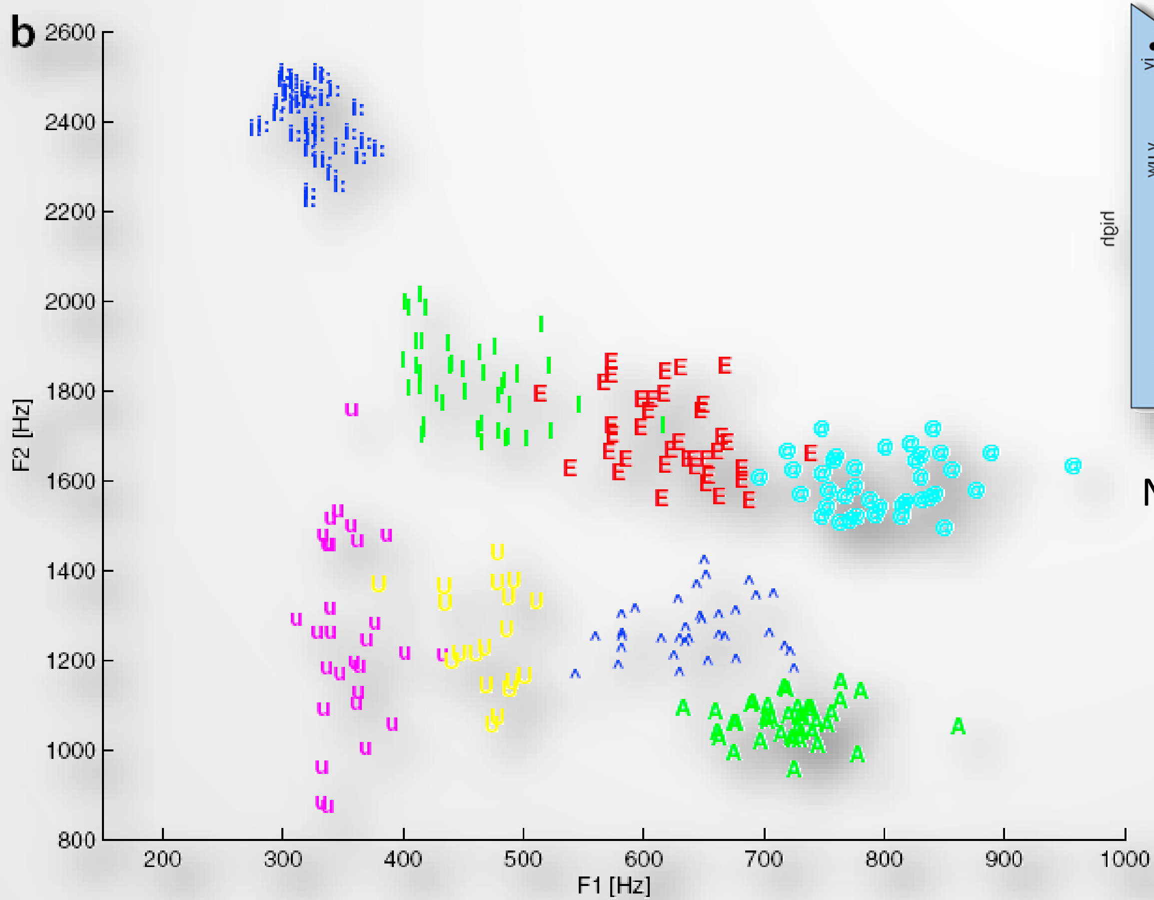
- **MFCC**s are the coefficients of the cepstrum of F-bank coefficients
- Altogether

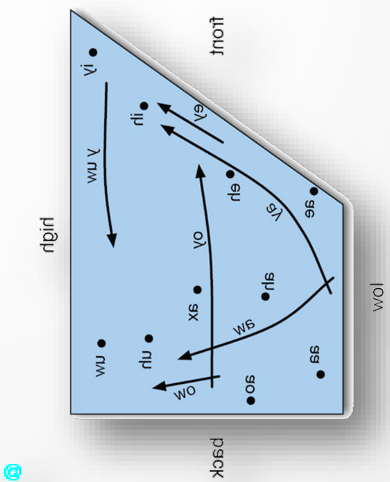
$$\textbf{Frame} \xrightarrow{\text{DFT}} \textbf{Spectrum} \xrightarrow{\log|\cdot|} \textbf{Log magnitude spectrum} \xrightarrow{\text{filter}} \textbf{F-bank coefficients} \xrightarrow{\text{DFT}^{-1}} \textbf{MFCCs}$$

- MFCCs are useful for models which can't handle speaker correlations themselves, like (diagonal) GMMs
- F-banks are better for those which can, like NNs

GAUSSIAN MIXTURES

Classifying speech sounds





Note: The vowel trapezoid's dimensions were physical

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

UNIVERSITY OF TORONTO

Classify speakers by cluster attributes

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

		Time, *t*			
		0	1	…	T
MFCC	1			…	
	2			…	
	3			…	
	…	…	…	…	…
	42			…	

Observation matrix

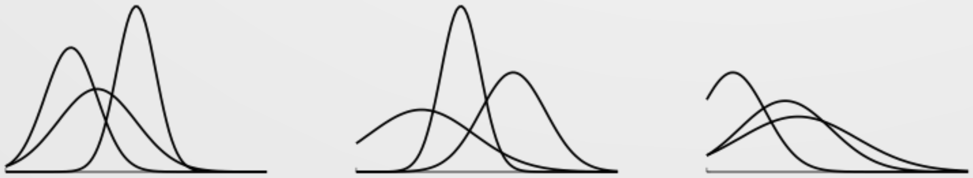
P(| ) >

P(| )

Speaker classification

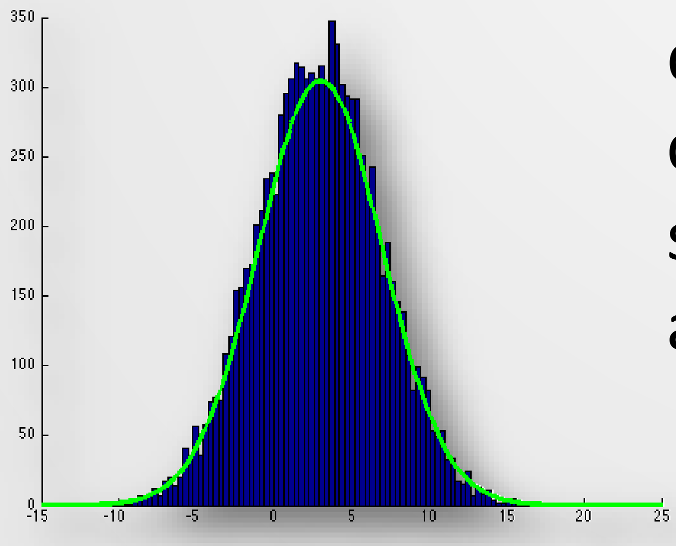
- **Speaker classification**: *n*. picking the most likely speaker among several speakers given only acoustics.

- Each **speaker** will produce speech according to **different** probability distributions.
  - We train a statistical model, given annotated data (mapping utterances to speakers).
  - We choose the speaker whose model gives the highest probability for an observation.



Fitting continuous distributions

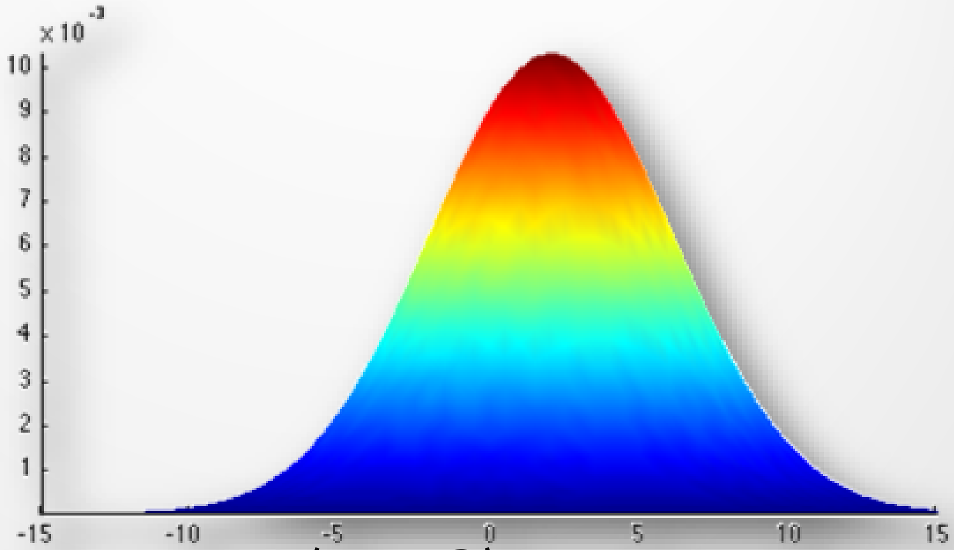
- Since we are operating with **continuous** variables, we need to **fit continuous probability** functions to a **discrete number** of observations.



- 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 μ and variance σ^2.

Univariate (1D) Gaussians

- Also known as **Normal** distributions, $N(\mu, \sigma)$



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

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

But we don't have samples for all x…

Maximum likelihood estimation

- Given data $X = \{x_1, x_2, \ldots, x_n\}$, MLE produces an estimate of the parameters $\hat{\theta}$ by maximizing the **likelihood**, $L(X, \theta)$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, L(X, \theta)$$

  where $\boldsymbol{L}(\boldsymbol{X}, \boldsymbol{\theta}) = \boldsymbol{P}(\boldsymbol{X}; \boldsymbol{\theta}) = \prod_{i=1}^{n} P(x_i; \theta)$.

- Since $L(X, \theta)$ provides a **surface** over all $\boldsymbol{\theta}$, 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 μ:

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

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

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

$$\mu = \frac{\sum_i x_i}{n}$$

- Similarly, $\sigma^2 = \frac{\sum_i (x_i-\mu)^2}{n}$

UNIVERSITY OF TORONTO

Multivariate Gaussians

- When data is ***d*-dimensional**, the input variable is

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

the **mean** is

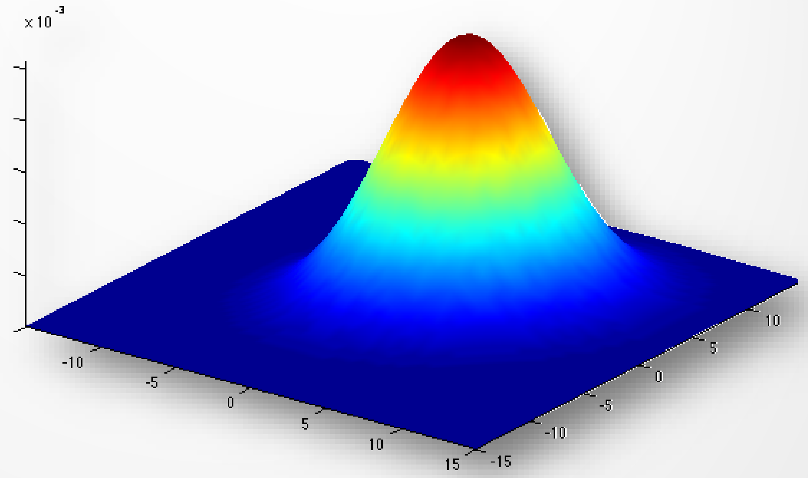
$$\vec{\mu} = E(\vec{x}) = \langle \mu[1], \mu[2], \dots, \mu[d] \rangle$$

the **covariance matrix** is

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

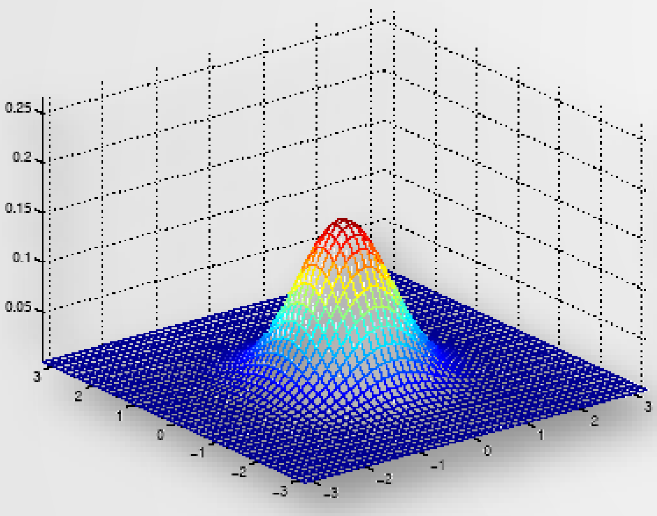
and

$$P(\vec{x}) = \frac{\exp\left(-\frac{(\vec{x}-\vec{\mu})^{\intercal}\Sigma^{-1}(\vec{x}-\vec{\mu})}{2}\right)}{(2\pi)^{\frac{d}{2}}\,|\Sigma|^{\frac{1}{2}}}$$

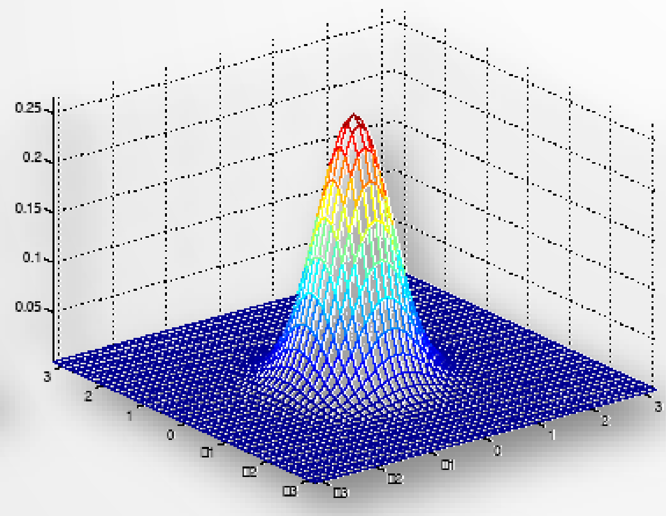


$A^{\intercal}$ is the **transpose** of A
A^{-1} is the **inverse** of A
$|A|$ is the **determinant** of A

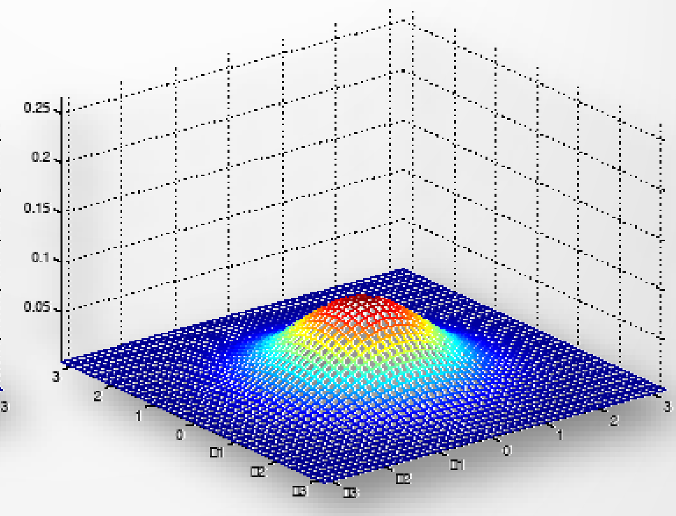
Intuitions of covariance



$$\mu = [0\ 0]$$
$$\Sigma = \mathbf{I}$$



$$\mu = [0\ 0]$$
$$\Sigma = 0.6\mathbf{I}$$

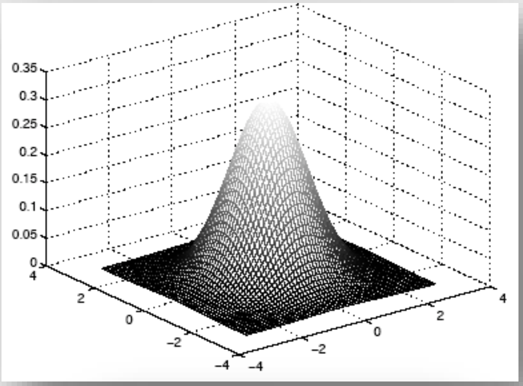


$$\mu = [0\ 0]$$
$$\Sigma = 2.0\mathbf{I}$$

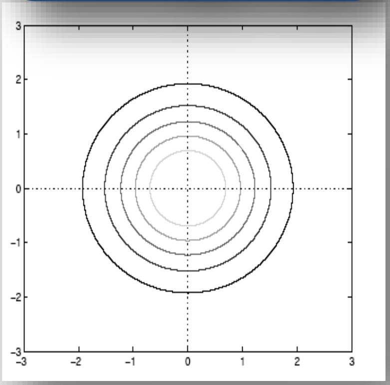
- As values in Σ become larger, the Gaussian spreads out.
- ($\mathbf{I}$ is the identity matrix)

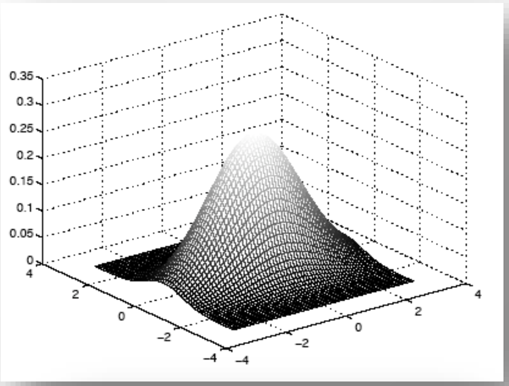
UNIVERSITY OF TORONTO

Intuitions of covariance

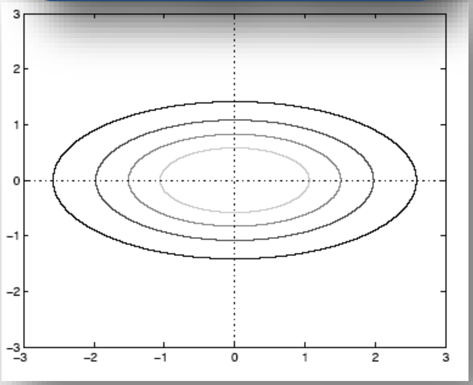


$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$





$$\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 0.6 \end{bmatrix}$$

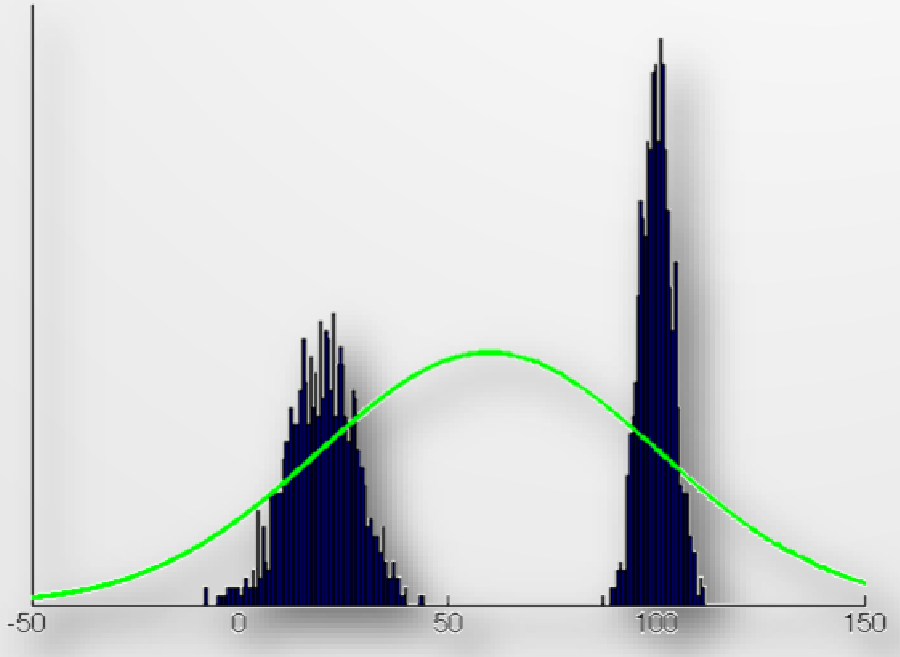


- Different values on the diagonal result in different variances in their respective dimensions

UNIVERSITY OF TORONTO

Non-Gaussian observations

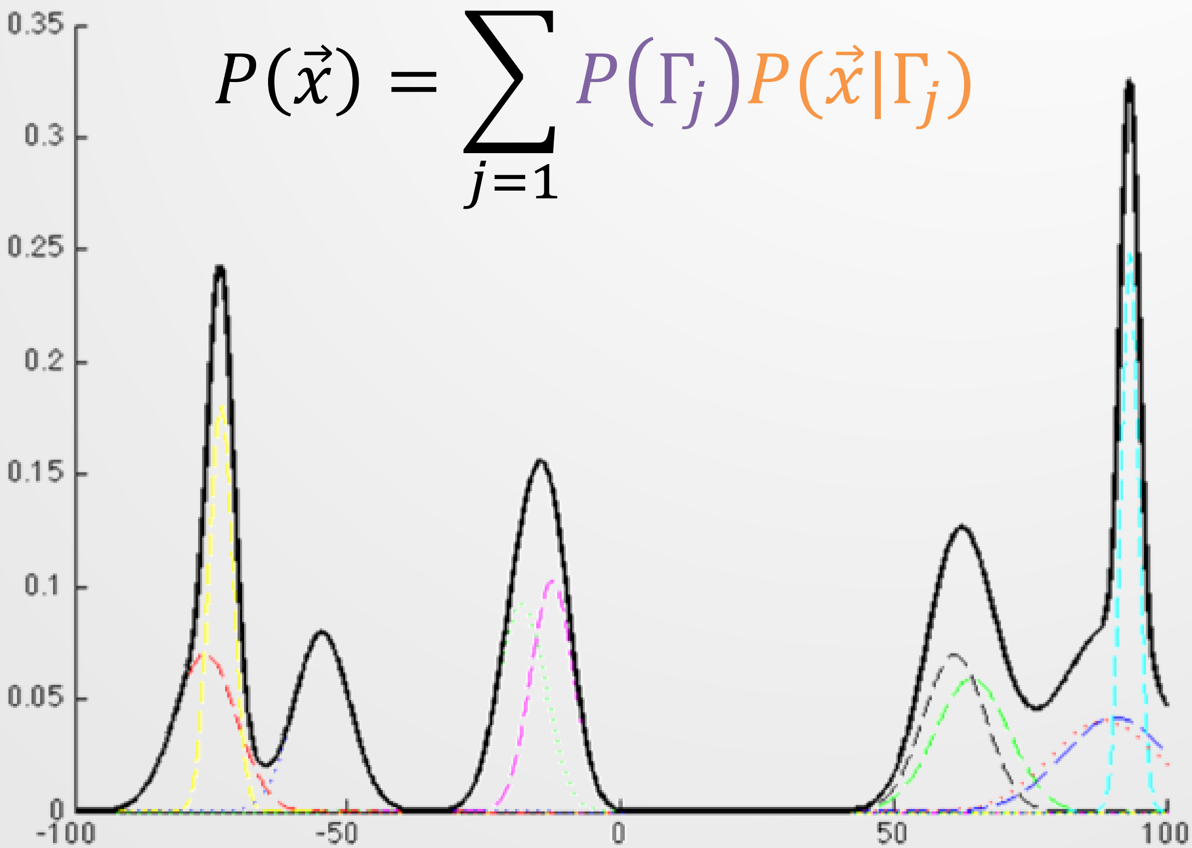
- Speech data are generally *not* unimodal.
- The observations below are **bimodal**, so fitting one Gaussian would not be representative.



Mixtures of Gaussians

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

$$P(\vec{x}) = \sum_{j=1}^{M} P(\Gamma_j) P(\vec{x} | \Gamma_j)$$



Observation likelihoods

- Assuming MFCC dimensions are independent of one another, the **covariance matrix is diagonal** – i.e., 0 off the diagonal.
- Therefore, the probability of an observation vector given a Gaussian becomes

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

- *Imagine* that a GMM first *chooses a Gaussian*, then *emits an observation* from that Gaussian.

MLE for GMMs

- Let $\boldsymbol{\omega_m} = P(\Gamma_m)$ and $\boldsymbol{b_m}(\overrightarrow{\boldsymbol{x_t}}) = P(\overrightarrow{x_t}|\Gamma_m)$,

'weight'

'component observation likelihood'

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

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

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

$$\log L(X, \theta) = \sum_{t=1}^{T} \log P_\theta(\overrightarrow{x_t}) = \sum_{t=1}^{T} \log \sum_{m=1}^{M} \omega_m b_m(\overrightarrow{x_t})$$

MLE for GMMs

- What happens when we try to find a maximum for $\mu_m[n]$?

$$\frac{\delta \log L(X,\theta)}{\delta \mu_m[n]} = \sum_{t=1}^{T} \frac{\delta}{\delta \mu_m[n]} \log \sum_{m'=1}^{M} \omega_{m'} b_{m'}(\overrightarrow{x_t}) = 0$$

$$\sum_{t=1}^{T} \frac{1}{P_\theta(\overrightarrow{x_t})} \frac{\delta}{\delta \mu_m[n]} \omega_m b_m(\overrightarrow{x_t}) = \sum_{t=1}^{T} \frac{\omega_m b_m(\overrightarrow{x_t})}{P_\theta(\overrightarrow{x_t})} \left(\frac{x_t[n] - \mu_m[n]}{\Sigma_m[n]^2} \right) = 0$$

$$\mu_m[n] = \frac{\sum_{t=1}^{T} \frac{\omega_m b_m(\overrightarrow{x_t})}{P_\theta(\overrightarrow{x_t})} x_t[n]}{\sum_{t=1}^{T} \frac{\omega_m b_m(\overrightarrow{x_t})}{P_\theta(\overrightarrow{x_t})}} = \frac{\sum_{t=1}^{T} P_\theta\left(\Gamma_m \middle| \overrightarrow{x_t}\right) x_t[n]}{\sum_{t=1}^{T} P_\theta\left(\Gamma_m \middle| \overrightarrow{x_t}\right)}$$

But this involves $\mu_m[n]$!

Learning mixtures of gaussians

- If we knew *which* Gaussian generated each sample, then $\langle \overrightarrow{\boldsymbol{\mu}_m}, \boldsymbol{\Sigma}_m \rangle$ can be learned by MLE.
- The MLE of $P(\Gamma_j)$ would likewise be the count $\frac{\#\overrightarrow{x_t} \text{ from } \Gamma_j}{T}$
- But we **don't** know this!
- Instead, we guess at "soft" mixture assignments $P_\theta(\Gamma_m | \overrightarrow{x_t})$ from another model…
- …which we got from a previous round of maximization

UNIVERSITY OF TORONTO

Expectation-Maximization for GMMs

- Overall idea:
  - First, initialize a set of model parameters.
  - “Expectation”: Compute the expected probabilities of observation, given these parameters.
  - “Maximization”: Update the parameters to maximize the aforementioned probabilities.
  - Repeat.

UNIVERSITY OF TORONTO

Expectation-Maximization for GMMs

- The **expectation step** gives us:

$$P_\theta(\Gamma_m|\overrightarrow{x_t}) = \frac{\omega_m b_m(\overrightarrow{x_t})}{P_\theta(\overrightarrow{x_t})}$$

Proportion of overall probability contributed by m

- The **maximization step** gives us:

$$\widehat{\overrightarrow{\mu_m}} = \frac{\sum_t P_\theta(\Gamma_m|\overrightarrow{x_t})\overrightarrow{x_t}}{\sum_t P_\theta(\Gamma_m|\overrightarrow{x_t})}$$

$$\widehat{\Sigma_m} = \frac{\sum_t P_\theta(\Gamma_m|\overrightarrow{x_t})\overrightarrow{x_t}^2}{\sum_t P_\theta(\Gamma_m|\overrightarrow{x_t})} - \widehat{\overrightarrow{\mu_m}}^2$$

$$\widehat{\omega_m} = \frac{1}{T}\sum_{t=1}^{T} P_\theta(\Gamma_m|\overrightarrow{x_t})$$

Recall from slide 18, MLE wants:

$$\mu = \frac{\sum_i x_i}{n}$$

$$\sigma^2 = \frac{\sum_i (x_i - \mu)^2}{n}$$

Recipe for GMM EM

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

1. Initialize:	Guess $\theta = \langle \omega_m, \overrightarrow{\mu_m}, \Sigma_m \rangle$ for $m = 1..M$ either uniformly, randomly, or by *k*-means clustering.
2. E-step:	Compute $P_\theta(\Gamma_m \vert \overrightarrow{x_t})$.
3. M-step:	Update parameters for $\langle \omega_m, \overrightarrow{\mu_m}, \Sigma_m \rangle$ with $\langle \widehat{\omega_m}, \widehat{\overrightarrow{\mu_m}}, \widehat{\Sigma_m} \rangle$ as described on slide 29.