# Building Speech Systems

K Sri Rama Murty

IIT Hyderabad

ksrm@ee.iith.ac.in

November 17, 2022

# Why Speech?

- Speech is the most natural forms of communication
  - Does not require any special training

# Why Speech?

- Speech is the most natural forms of communication
  - Does not require any special training
- Flexible
  - Hands/Eyes free communication
  - Radio survived the era of HD television!
  - Assistive technologies for blind
  - Interaction while driving - Google maps instructions

# Why Speech?

- Speech is the most natural forms of communication
  - Does not require any special training
- Flexible
  - Hands/Eyes free communication
  - Radio survived the era of HD television!
  - Assistive technologies for blind
  - Interaction while driving - Google maps instructions
- Efficient
  - Conveys lot more information than mere text content

# Why Speech?

- Speech is the most natural forms of communication
  - Does not require any special training
- Flexible
  - Hands/Eyes free communication
  - Radio survived the era of HD television!
  - Assistive technologies for blind
  - Interaction while driving - Google maps instructions
- Efficient
  - Conveys lot more information than mere text content
- Economical
  - Inexpensive transmission and reception of information
  - Voice communication is the reason behind success of mobile phones

# Information in Speech Signal

- Linguistic Information
  - Information that can be represented by a set of discrete symbols
  - Conveys textual message in the speech signal
  - Language recognition, speech recognition, search for specific words

# Information in Speech Signal

- Linguistic Information
  - Information that can be represented by a set of discrete symbols
  - Conveys textual message in the speech signal
  - Language recognition, speech recognition, search for specific words
- Paralinguisitc Information
  - Supplemental information that is not inferable from written text
  - Added by speaker to modify & supplement the linguistic information
  - Intention, attitude, emphasis, speaking style, etc.
  - It can be consciously controlled by the speaker

# Information in Speech Signal

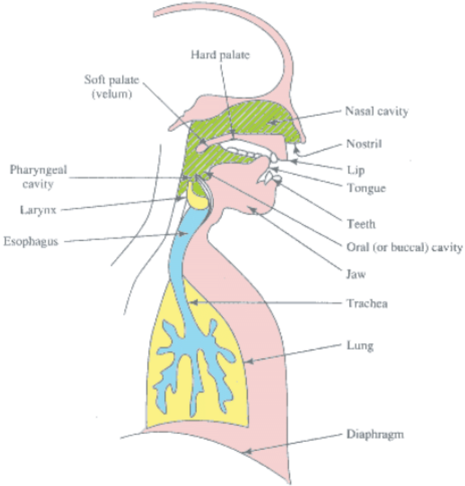
- Linguistic Information
  - Information that can be represented by a set of discrete symbols
  - Conveys textual message in the speech signal
  - Language recognition, speech recognition, search for specific words
- Paralinguisitc Information
  - Supplemental information that is not inferable from written text
  - Added by speaker to modify & supplement the linguistic information
  - Intention, attitude, emphasis, speaking style, etc.
  - It can be consciously controlled by the speaker
- Nonlinguistic Information
  - Factors that cannot, generally, be controlled by the speaker
  - Gender, speaker identity, age, physical build, emotional state, health, idiosyncrasy (filler words, mother-tongue), etc.

# Information in Speech Signal

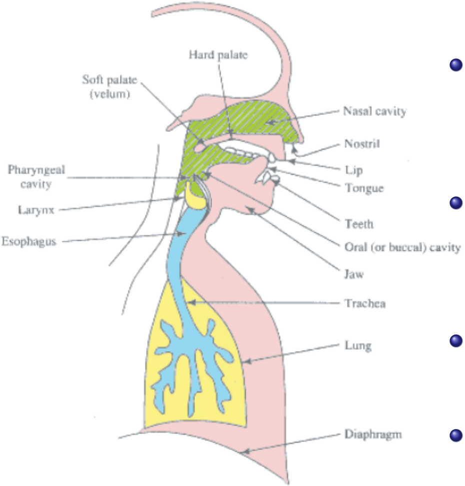
- Linguistic Information
  - Information that can be represented by a set of discrete symbols
  - Conveys textual message in the speech signal
  - Language recognition, speech recognition, search for specific words
- Paralinguisitc Information
  - Supplemental information that is not inferable from written text
  - Added by speaker to modify & supplement the linguistic information
  - Intention, attitude, emphasis, speaking style, etc.
  - It can be consciously controlled by the speaker
- Nonlinguistic Information
  - Factors that cannot, generally, be controlled by the speaker
  - Gender, speaker identity, age, physical build, emotional state, health, idiosyncrasy (filler words, mother-tongue), etc.
- Background Information - Acoustic environment around the speaker

# Acoustic Theory of Speech Production

# Speech Production Mechanism

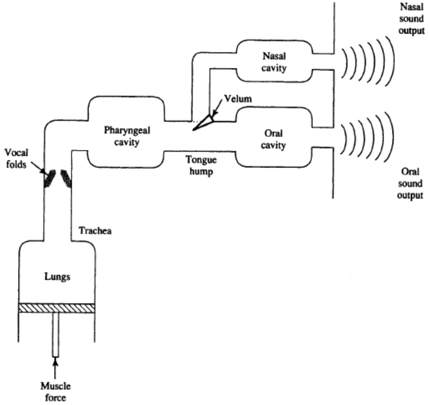


# Speech Production Mechanism



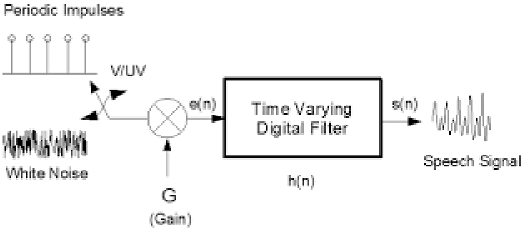

- Speech is the most sophisticated motor activity in the human body
- The motions are lightening fast and totally fluid, yet highly accurate
- Nasal cavity is much larger than oral cavity
- Animation

# Block Diagram of Speech Production

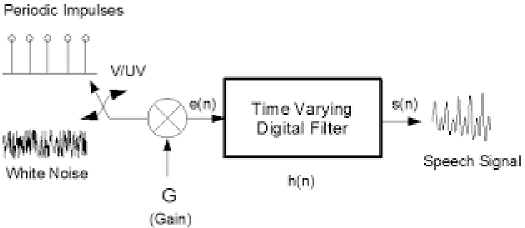



- Lungs act as source of energy
- Vocal folds chop the airflow from lungs into quasi periodic puffs of excitation
- Shape of the vocal tract determines the sound that is produced
- Velum is big enough is decouple nasal cavity, but not oral cavity

# Source-Filter Model of Speech Production

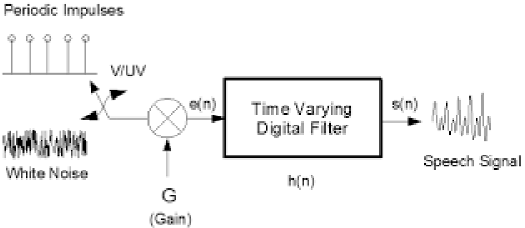


# Source-Filter Model of Speech Production



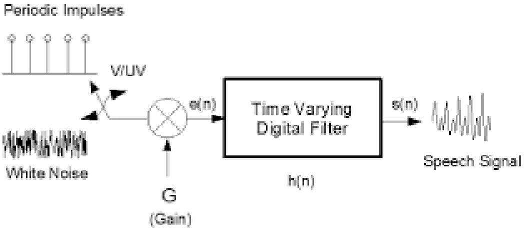

- Speech is an outcome of time-varying vocal tract filter driven by a time-varying excitation

# Source-Filter Model of Speech Production



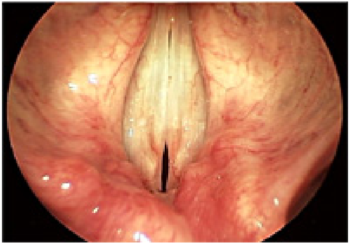

- Speech is an outcome of time-varying vocal tract filter driven by a time-varying excitation
- The state of the vocal cords, the positions, shapes and sizes of the various articulators – all change slowly over time, thereby producing the desired speech sounds

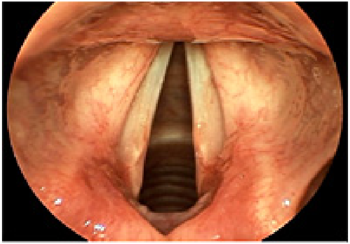
# Source-Filter Model of Speech Production



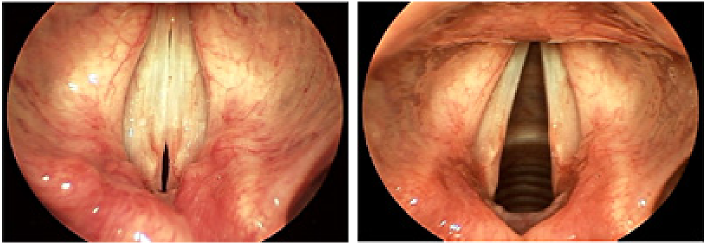

- Speech is an outcome of time-varying vocal tract filter driven by a time-varying excitation
- The state of the vocal cords, the positions, shapes and sizes of the various articulators – all change slowly over time, thereby producing the desired speech sounds
- Need to determine the state of VT from waveform - inverse problem

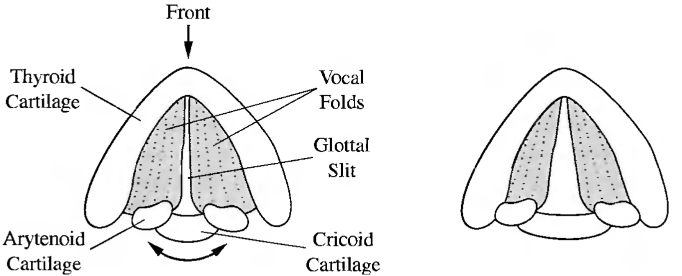
# Vocal Folds - View & Operation





# Vocal Folds - View & Operation

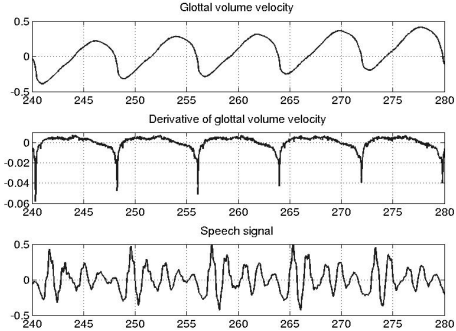





Video - 1 Video - 2

# Voice Source - Male vs Female

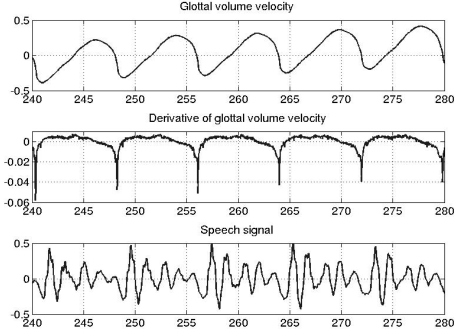
Male Speaker




EGG signal Speech signal

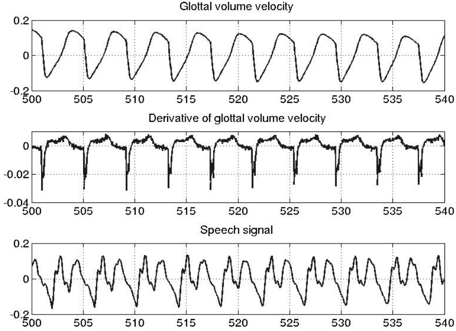
# Voice Source - Male vs Female

Male Speaker



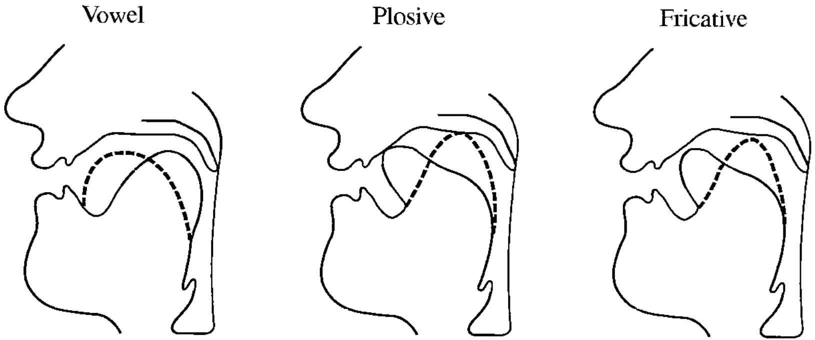

EGG signal Speech signal

Female Speaker

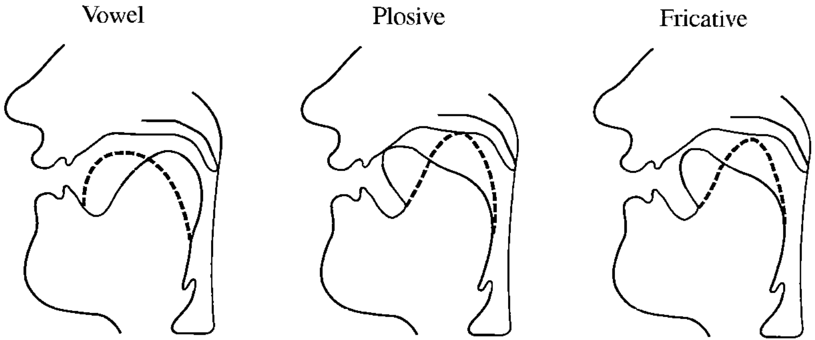



EGG signal Speech signal

# Important Vocal-Tract Configurations

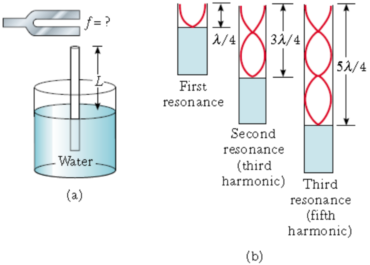


# Important Vocal-Tract Configurations

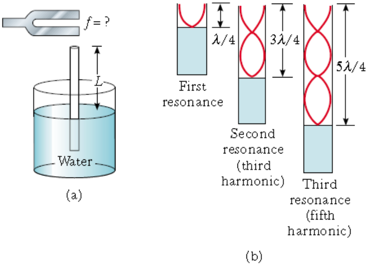



- Vowel - Relatively Open vocal-tract
- Plosive (Stop consonant) - VT is closed at some point
- Fricative - Constricted vocal tract

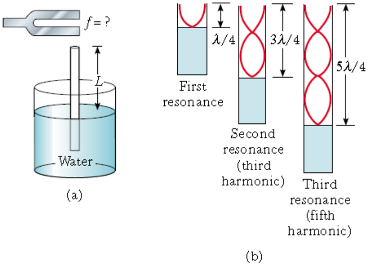
# Resonances of a Uniform Tube



# Resonances of a Uniform Tube

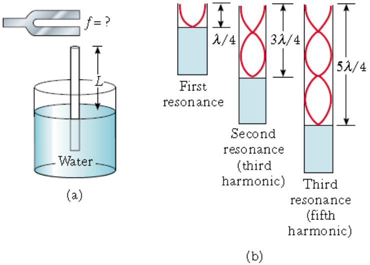


# Resonances of a Uniform Tube



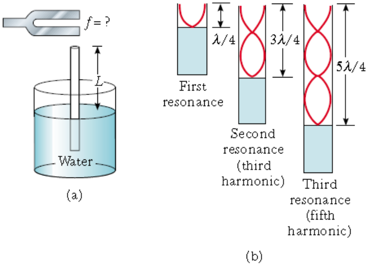

- The air column has only certain natural frequencies

# Resonances of a Uniform Tube



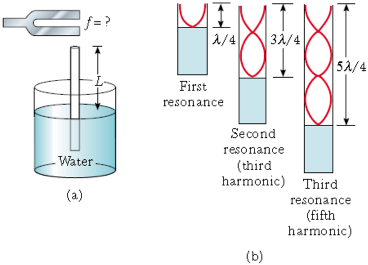

- The air column has only certain natural frequencies
- If the tuning fork has the right frequency, the air column in the tube resonates loudly

# Resonances of a Uniform Tube



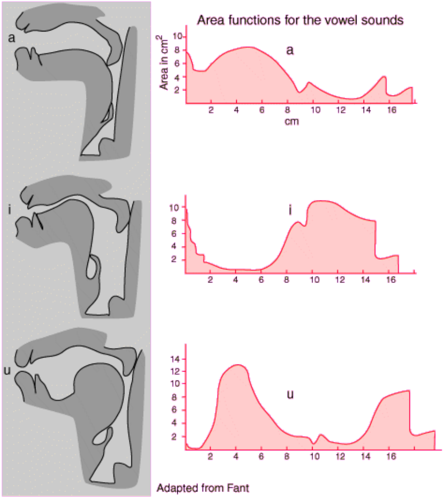

- The air column has only certain natural frequencies
- If the tuning fork has the right frequency, the air column in the tube resonates loudly
- A standing wave pattern - maximum air displacement at open end & no air displacement at closed end.

# Resonances of a Uniform Tube

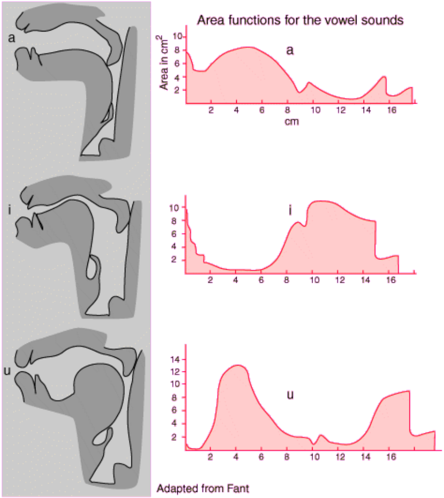



- The air column has only certain natural frequencies
- If the tuning fork has the right frequency, the air column in the tube resonates loudly
- A standing wave pattern - maximum air displacement at open end & no air displacement at closed end.
- The tube acts like an acoustic filter - the frequency response of which depends on its dimensions

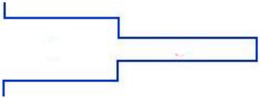
# Two-Tube Vowel Models

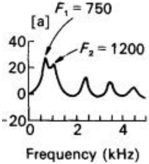


# Two-Tube Vowel Models

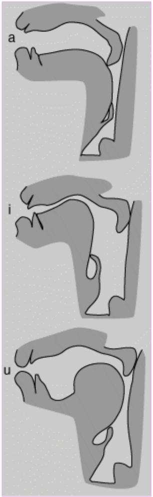


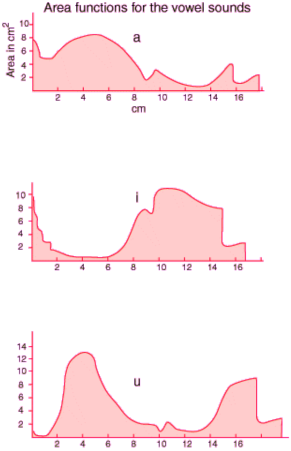

Adapted from Fant



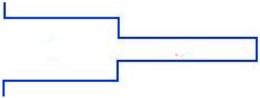


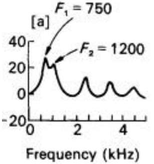
# Two-Tube Vowel Models

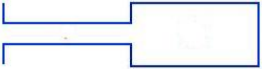


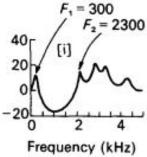




Adapted from Fant

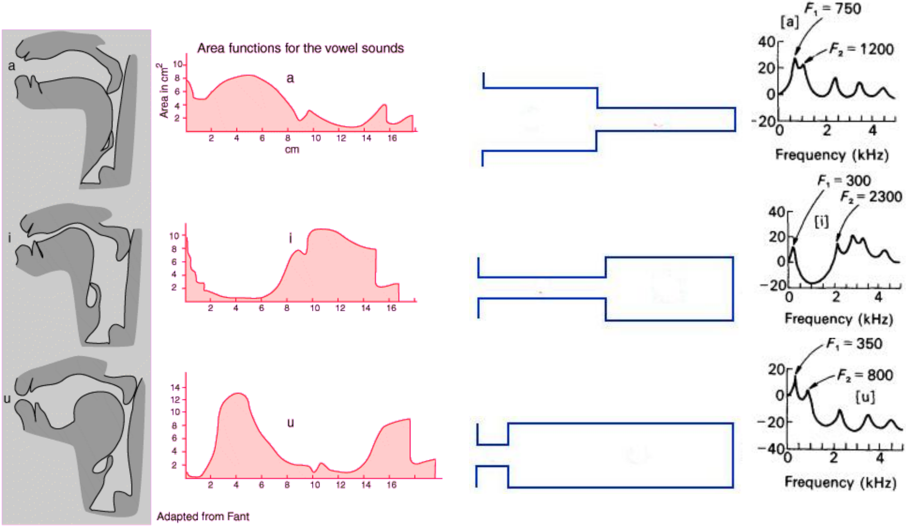






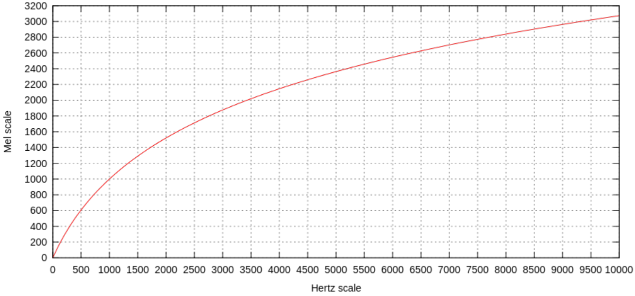
# Two-Tube Vowel Models



# Feature Extraction

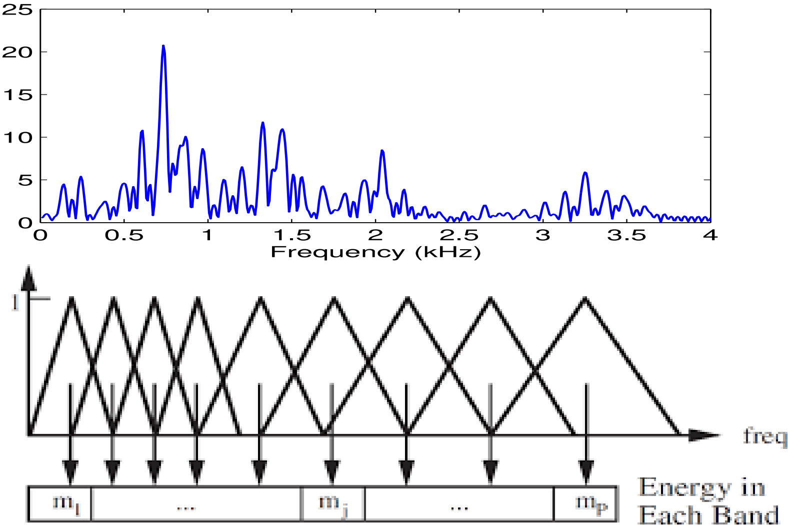
# Mel Scale

- Human beings can resolve low-frequency sounds better
- Mel scale incorporates this feature of human perception
- Mel scale is linear up to 1000 Hz, and logarithmic after that
- Filters are placed uniformly-spaced along Mel scale

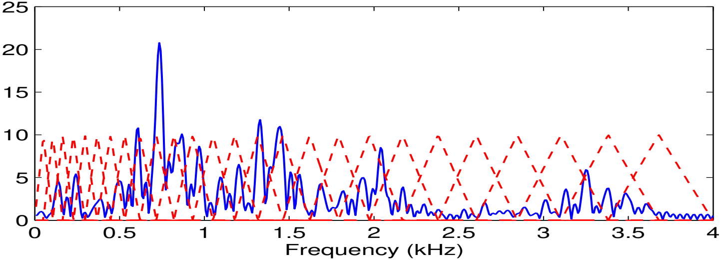



$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

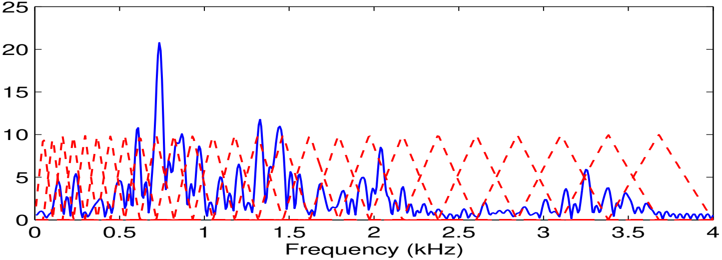
# Mel Filters

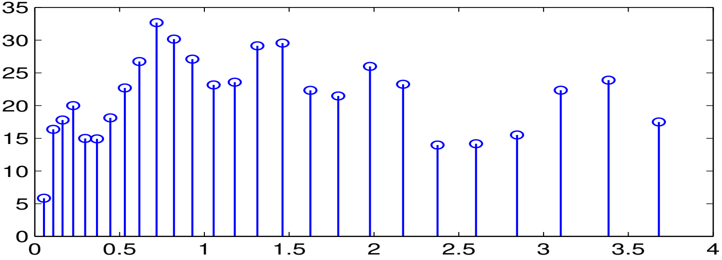

25
20
15
10
5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
Frequency (kHz)
1
freq
$m_1$
...
$m_j$
...
$m_P$
Energy in
Each Band

# Mel-Filter Bank Energy Coefficients



# Mel-Filter Bank Energy Coefficients



# Mel Frequency Cepstral Coefficients

MFCCs from waveform

- 20-30 ms window: $s[n]$
- N-point DFT - $S[k]$
- Squared magnitude $|S[k]|^2$
- **S**: symmetric half of $|S[k]|^2$

# Mel Frequency Cepstral Coefficients

MFCCs from waveform

- 20-30 ms window: $s[n]$
- N-point DFT - $S[k]$
- Squared magnitude $|S[k]|^2$
- $\mathbf{S}$: symmetric half of $|S[k]|^2$
- Energy over Mel-filter bands

$$\mathbf{S}_m = \mathbf{W}_{M \times \frac{N}{2}} \mathbf{S}_{\frac{N}{2} \times 1}$$

$\mathbf{W}$ is the Mel-weight matrix

## Mel Frequency Cepstral Coefficients

MFCCs from waveform

- 20-30 ms window: $s[n]$
- N-point DFT - $S[k]$
- Squared magnitude $|S[k]|^2$
- $\mathbf{S}$: symmetric half of $|S[k]|^2$
- Energy over Mel-filter bands

$$\mathbf{S}_m = \mathbf{W}_{M\times\frac{N}{2}}\mathbf{S}_{\frac{N}{2}\times 1}$$

  $\mathbf{W}$ is the Mel-weight matrix
- Log energy - $\mathbf{S}_l = \log \mathbf{S}_m$
- Compute IDCT: $\mathbf{S}_c = \mathbf{D}_{M\times M}\mathbf{S}_l$
- Retain first few coefficients $\hat{\mathbf{S}}_c$

# Mel Frequency Cepstral Coefficients

MFCCs from waveform

- 20-30 ms window: $s[n]$
- N-point DFT - $S[k]$
- Squared magnitude $|S[k]|^2$
- $\mathbf{S}$: symmetric half of $|S[k]|^2$
- Energy over Mel-filter bands

$$\mathbf{S}_m = \mathbf{W}_{M \times \frac{N}{2}} \mathbf{S}_{\frac{N}{2} \times 1}$$

$\mathbf{W}$ is the Mel-weight matrix

- Log energy - $\mathbf{S}_l = \log \mathbf{S}_m$
- Compute IDCT: $\mathbf{S}_c = \mathbf{D}_{M \times M} \mathbf{S}_l$
- Retain first few coefficients $\hat{\mathbf{S}}_c$

Waveform from MFCCs

- MFCCs $\hat{\mathbf{S}}_c$ are given
- Append zeros to length $M$
- Compute DCT: $\hat{\mathbf{S}}_l = \mathbf{D}^{-1} \hat{\mathbf{S}}_c$
- Exponential: $\hat{\mathbf{S}}_m = \exp \hat{\mathbf{S}}_l$

# Mel Frequency Cepstral Coefficients

MFCCs from waveform

- 20-30 ms window: $s[n]$
- N-point DFT - $S[k]$
- Squared magnitude $|S[k]|^2$
- $\mathbf{S}$: symmetric half of $|S[k]|^2$
- Energy over Mel-filter bands

$$\mathbf{S}_m = \mathbf{W}_{M\times\frac{N}{2}}\mathbf{S}_{\frac{N}{2}\times 1}$$

$\mathbf{W}$ is the Mel-weight matrix

- Log energy - $\mathbf{S}_l = \log \mathbf{S}_m$
- Compute IDCT: $\mathbf{S}_c = \mathbf{D}_{M\times M}\mathbf{S}_l$
- Retain first few coefficients $\hat{\mathbf{S}}_c$

Waveform from MFCCs

- MFCCs $\hat{\mathbf{S}}_c$ are given
- Append zeros to length $M$
- Compute DCT: $\hat{\mathbf{S}}_l = \mathbf{D}^{-1}\hat{\mathbf{S}}_c$
- Exponential: $\hat{\mathbf{S}}_m = \exp \hat{\mathbf{S}}_l$
- Pseudo inverse: $\hat{\mathbf{S}} = \mathbf{W}^{-1}\hat{\mathbf{S}}_m$
- Square root: $|S[k]|$

# Mel Frequency Cepstral Coefficients

MFCCs from waveform

- 20-30 ms window: $s[n]$
- N-point DFT - $S[k]$
- Squared magnitude $|S[k]|^2$
- $\mathbf{S}$: symmetric half of $|S[k]|^2$
- Energy over Mel-filter bands

$$\mathbf{S}_m = \mathbf{W}_{M \times \frac{N}{2}} \mathbf{S}_{\frac{N}{2} \times 1}$$

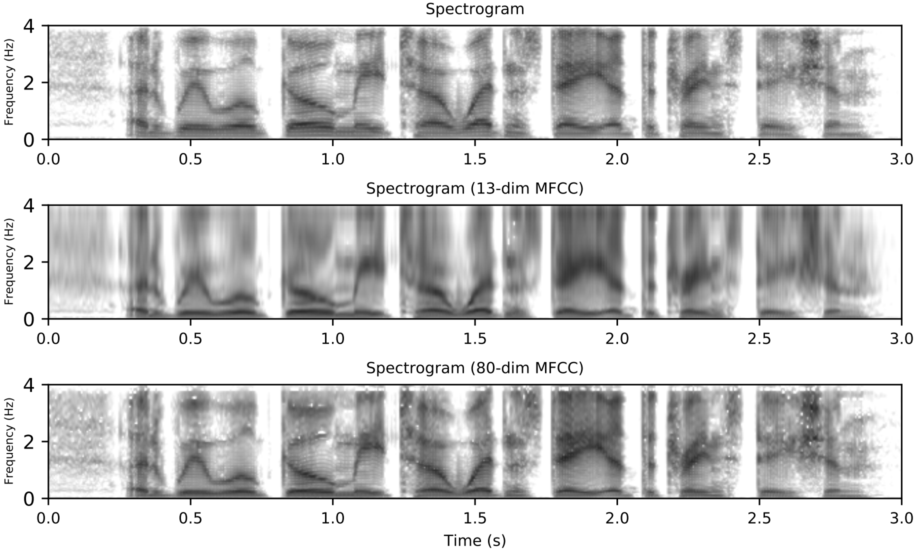
$\mathbf{W}$ is the Mel-weight matrix

- Log energy - $\mathbf{S}_l = \log \mathbf{S}_m$
- Compute IDCT: $\mathbf{S}_c = \mathbf{D}_{M \times M} \mathbf{S}_l$
- Retain first few coefficients $\hat{\mathbf{S}}_c$

Waveform from MFCCs

- MFCCs $\hat{\mathbf{S}}_c$ are given
- Append zeros to length $M$
- Compute DCT: $\hat{\mathbf{S}}_l = \mathbf{D}^{-1} \hat{\mathbf{S}}_c$
- Exponential: $\hat{\mathbf{S}}_m = \exp \hat{\mathbf{S}}_l$
- Pseudo inverse: $\hat{\mathbf{S}} = \mathbf{W}^{-1} \hat{\mathbf{S}}_m$
- Square root: $|S[k]|$
- Phase information is missing
- Griffin-Lim algo.: $e^{j\angle S[k]}$
- Inverse DFT: $\hat{s}[n]$

# Effect of Cepstral Order



# Typical Feature Configurations

- Statistical models
  - MFCCs are preferred representation as the dimensions can be assumed to be uncorrelated.

# Typical Feature Configurations

- Statistical models
    - MFCCs are preferred representation as the dimensions can be assumed to be uncorrelated.
    - Speech recognition: 13 MFCCs at 8 kHz (24 MFCCs at 16 kHz)
    - Speaker recognition: 20 MFCCs at 8 kHz (26 MFCCs at 16kHz)
    - Language recognition: 7 MFCCs at 8 kHz (12 MFCCs at 16 kHz)
    - Speech synthesis: 80-128 MFCCs at 16 kHz.
    - Delta and acceleration coefficients are explicitly added

# Typical Feature Configurations

- Statistical models
  - MFCCs are preferred representation as the dimensions can be assumed to be uncorrelated.
  - Speech recognition: 13 MFCCs at 8 kHz (24 MFCCs at 16 kHz)
  - Speaker recognition: 20 MFCCs at 8 kHz (26 MFCCs at 16kHz)
  - Language recognition: 7 MFCCs at 8 kHz (12 MFCCs at 16 kHz)
  - Speech synthesis: 80-128 MFCCs at 16 kHz.
  - Delta and acceleration coefficients are explicitly added
- Neural network models
  - Prefers dependent dimensions to capture higher-order correlations

# Typical Feature Configurations

- Statistical models
  - MFCCs are preferred representation as the dimensions can be assumed to be uncorrelated.
  - Speech recognition: 13 MFCCs at 8 kHz (24 MFCCs at 16 kHz)
  - Speaker recognition: 20 MFCCs at 8 kHz (26 MFCCs at 16kHz)
  - Language recognition: 7 MFCCs at 8 kHz (12 MFCCs at 16 kHz)
  - Speech synthesis: 80-128 MFCCs at 16 kHz.
  - Delta and acceleration coefficients are explicitly added
- Neural network models
  - Prefers dependent dimensions to capture higher-order correlations
  - 40-dimensional Mel-frequency energy coefficients are used.
  - Temporal context window of CNNs and RNNs implicitly capture the temporal dynamics

# Typical Feature Configurations

- Statistical models
  - MFCCs are preferred representation as the dimensions can be assumed to be uncorrelated.
  - Speech recognition: 13 MFCCs at 8 kHz (24 MFCCs at 16 kHz)
  - Speaker recognition: 20 MFCCs at 8 kHz (26 MFCCs at 16kHz)
  - Language recognition: 7 MFCCs at 8 kHz (12 MFCCs at 16 kHz)
  - Speech synthesis: 80-128 MFCCs at 16 kHz.
  - Delta and acceleration coefficients are explicitly added
- Neural network models
  - Prefers dependent dimensions to capture higher-order correlations
  - 40-dimensional Mel-frequency energy coefficients are used.
  - Temporal context window of CNNs and RNNs implicitly capture the temporal dynamics
  - Recent DNN approaches extract features from the raw-waveform.

# Feature Extraction using Librosa

- Extract MFCC feature vectors using Librosa library

```
import librosa
y, sr = librosa.load('speech.wav', sr=16000)
mfcc = librosa.feature.mfcc(y=y, sr=sr)
mfcc_delta = librosa.feature.delta(mfcc)
mfcc_delta2 = librosa.feature.delta(mfcc, order=2)
```

- Concatenate MFCC and delta features to form 60-dim vector

# Speaker Recognition

# Speaker Recognition

- Task of recognizing a speaker from his/her voice
  - **Identification:** Whether an unknown speaker matches one in the list?
  - **Verification:** Whether an unknown speaker is who she claims to be?

# Speaker Recognition

- Task of recognizing a speaker from his/her voice
  - **Identification:** Whether an unknown speaker matches one in the list?
  - **Verification:** Whether an unknown speaker is who she claims to be?
- Applications of voice biometric
  - Remotely recognizable & contactless
  - Authenticating mobile devices, voice mails, telebanking etc.
  - Forensic applications

# Speaker Recognition

- Task of recognizing a speaker from his/her voice
  - **Identification:** Whether an unknown speaker matches one in the list?
  - **Verification:** Whether an unknown speaker is who she claims to be?
- Applications of voice biometric
  - Remotely recognizable & contactless
  - Authenticating mobile devices, voice mails, telebanking etc.
  - Forensic applications
- Not an easy task: Humans make 23% errors on unfamiliar voices

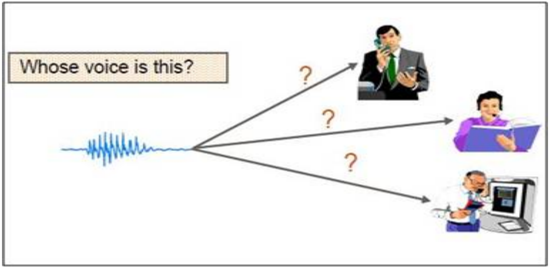
# Speaker Recognition

- Task of recognizing a speaker from his/her voice
  - **Identification:** Whether an unknown speaker matches one in the list?
  - **Verification:** Whether an unknown speaker is who she claims to be?
- Applications of voice biometric
  - Remotely recognizable & contactless
  - Authenticating mobile devices, voice mails, telebanking etc.
  - Forensic applications
- Not an easy task: Humans make 23% errors on unfamiliar voices
- Speaker-specific information in the speech signal
  - **Anatomical differences:** VT size & shape, vocal-fold thickness, pitch..
  - **Learned speaking habits:** dialect, prosody, speaking rate, disfluencies

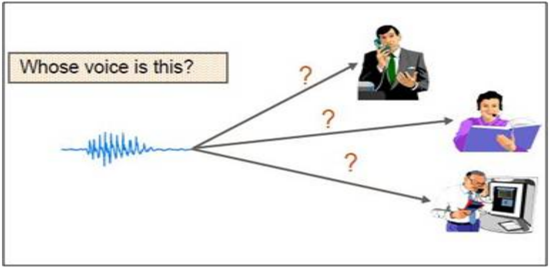
# Types of Speaker Recognition Systems

- **Text-independent speaker recognition**
  - Text content of enrollment and test utterances need not match
  - Recognition system does not know text spoken at test time
  - Offers a flexible system, but difficult to realize
  - Useful in forensic applications
  - Statistical pattern matching techniques are used
- **Text-dependent speaker recognition**
  - Enrollment and test utterances should be of same text
  - Recognition system knows the text spoken at test time
  - Knowledge of known text improves the performance
  - Applications involving cooperative users (banking, mobile phones)
  - Template matching techniques, like DTW, are used

# Text-Independent Speaker Identification

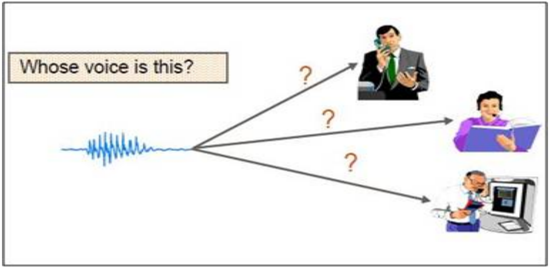


# Text-Independent Speaker Identification



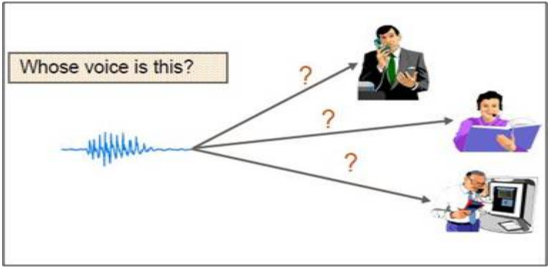

- As text content is different, sequence feature vectors do not match

# Text-Independent Speaker Identification



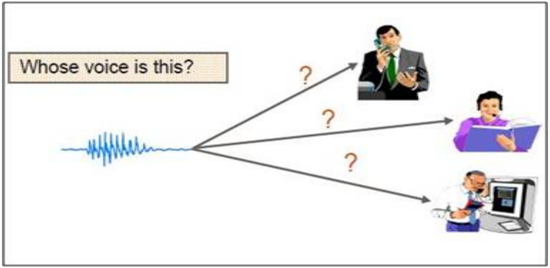

- As text content is different, sequence feature vectors do not match
- The enrollment and test utterances could be of different length

# Text-Independent Speaker Identification



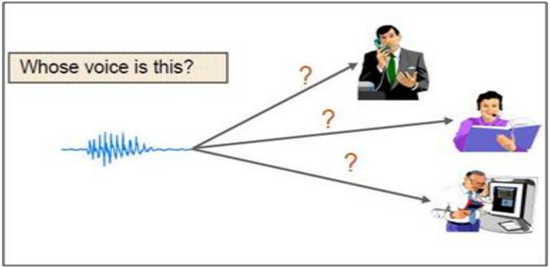

- As text content is different, sequence feature vectors do not match
- The enrollment and test utterances could be of different length
- Estimate the pdf of features from enrollment utterance

# Text-Independent Speaker Identification




- As text content is different, sequence feature vectors do not match
- The enrollment and test utterances could be of different length
- Estimate the pdf of features from enrollment utterance
- Evaluate probability of drawing test data from the estimated pdf

# Text-Independent Speaker Identification




- As text content is different, sequence feature vectors do not match
- The enrollment and test utterances could be of different length
- Estimate the pdf of features from enrollment utterance
- Evaluate probability of drawing test data from the estimated pdf
- Assign speaker-$k$ if $p(\text{test data}/\lambda_k)$ is the highest

# Speaker Identification using GMMs

- Extract features from the speakers data
- Estimate pdf of the features of each speaker using GMM
- Let the speaker models be denoted by $\lambda_k, \quad k = 1, 2, \cdots, K$
- Let $X$ denotes the set of features extracted from a test utterance
- Assign test utterance to the model with maximum likelihood

$$\text{Speaker Id} = \arg\max_k p(X/\lambda_k)$$

- Issues with this approach
  - Likelihoods from different *model estimates* are not comparable
  - All the models might not have got trained to the same extent
  - The variance of the data could be different across speakers
  - Speaker models may yield consistently higher/lower likelihoods

# Speaker Recognition using SkLearn

# Speaker Recognition using SkLearn

- Consider speech data from 10 speakers (at least 1 min each)

# Speaker Recognition using SkLearn

- Consider speech data from 10 speakers (at least 1 min each)
- Extract MFCC feature vectors from the all the speech data

# Speaker Recognition using SkLearn

- Consider speech data from 10 speakers (at least 1 min each)
- Extract MFCC feature vectors from the all the speech data
- Pool all the features (X) and build a 64 mixture GMM

```
ubm = sklearn.mixture.GaussianMixture(
    n_components=64, covariance_type='diag')
ubm.fit(X)
```

## Speaker Recognition using SkLearn

- Consider speech data from 10 speakers (at least 1 min each)
- Extract MFCC feature vectors from the all the speech data
- Pool all the features (X) and build a 64 mixture GMM

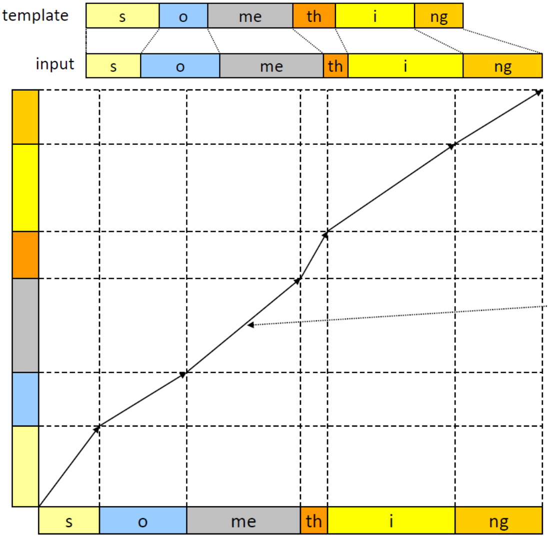
```
ubm = sklearn.mixture.GaussianMixture(
    n_components=64, covariance_type='diag')
ubm.fit(X)
```

- Adapt UBM to each spaker to build speaker-specific model

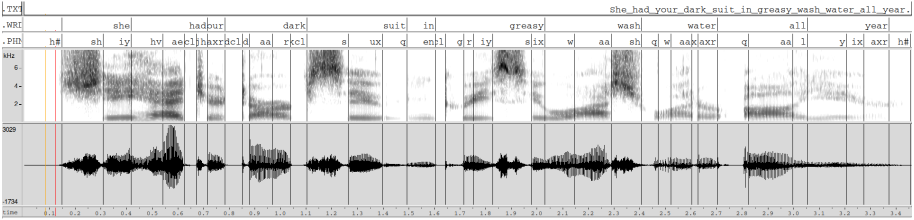
```
spkr1 = sklearn.mixture.GaussianMixture(
        n_components=64, max_iter=1, covariance_type='
        means_init=ubm.means_, weights_init=ubm.weight
        precisions_init=ubm.precisions_)
spkr1.fit(X1)
```

# Isolated Word Recognition

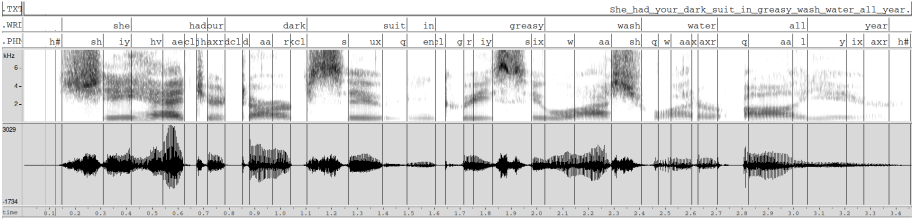
# Isolated Word Recognition Using DTW



# Speech Recognition

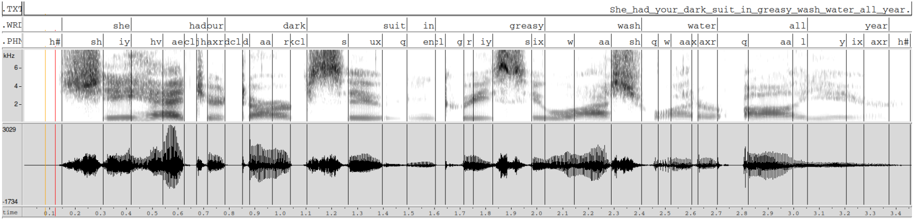


# Speech Recognition



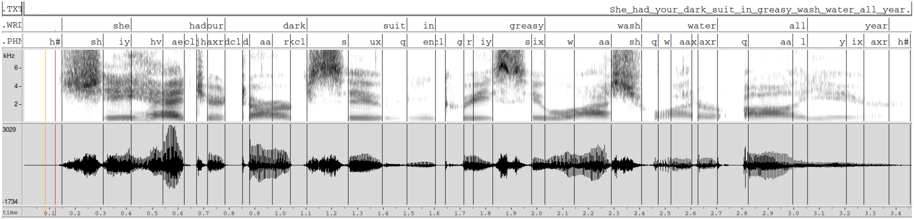

- The task of recognizing the text from the acoustic signal

# Speech Recognition



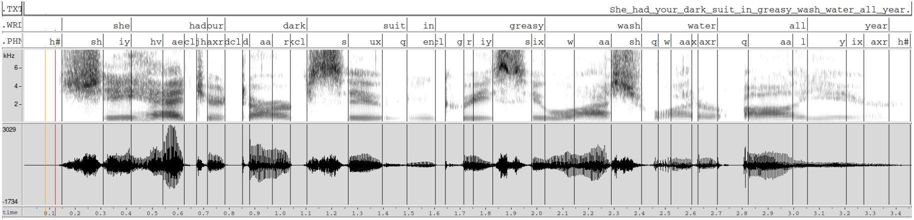

- The task of recognizing the text from the acoustic signal
- Attempted as a supervised learning task

# Speech Recognition




- The task of recognizing the text from the acoustic signal
- Attempted as a supervised learning task
- Determine the best possible word sequence from the observed signal

# Speech Recognition




- The task of recognizing the text from the acoustic signal
- Attempted as a supervised learning task
- Determine the best possible word sequence from the observed signal
- Time-domain samples → Feature representation → Subword units → Words → Sentences

# Mathematical Formulation

# Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_{W} P[W/\mathbf{O}]$$

## Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_{W} P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features

## Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features
- The posterior probability $P(W/\mathbf{O})$ can be evaluated using Bayes as

$$P[W/\mathbf{O}] = \frac{P[W]p(\mathbf{O}/W)}{p(\mathbf{O})}$$

# Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features
- The posterior probability $P(W/\mathbf{O})$ can be evaluated using Bayes as

$$P[W/\mathbf{O}] = \frac{P[W]p(\mathbf{O}/W)}{p(\mathbf{O})}$$

  - $P[W] = P[w_1, w_2, \ldots w_k]$ is the probability of word sequence

# Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features
- The posterior probability $P(W/\mathbf{O})$ can be evaluated using Bayes as

$$P[W/\mathbf{O}] = \frac{P[W]p(\mathbf{O}/W)}{p(\mathbf{O})}$$

  - $P[W] = P[w_1, w_2, \dots w_k]$ is the probability of word sequence
  - $p(\mathbf{O}/W)$ is the probability of observing feature $\mathbf{O}$ given $W$ is uttered

# Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_{W} P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features
- The posterior probability $P(W/\mathbf{O})$ can be evaluated using Bayes as

$$P[W/\mathbf{O}] = \frac{P[W]p(\mathbf{O}/W)}{p(\mathbf{O})}$$

  - $P[W] = P[w_1, w_2, \ldots w_k]$ is the probability of word sequence
  - $p(\mathbf{O}/W)$ is the probability of observing feature $\mathbf{O}$ given $W$ is uttered
  - $p(\mathbf{O})$ is the partition function that normalizes the posterior estimates.

## Mathematical Formulation

- Determine the most likely word sequence given the observation seq.

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- $\mathbf{O}$ denotes the acoustic evidence as captured by the features
- The posterior probability $P(W/\mathbf{O})$ can be evaluated using Bayes as

$$P[W/\mathbf{O}] = \frac{P[W]p(\mathbf{O}/W)}{p(\mathbf{O})}$$

  - $P[W] = P[w_1, w_2, \ldots w_k]$ is the probability of word sequence
  - $p(\mathbf{O}/W)$ is the probability of observing feature $\mathbf{O}$ given $W$ is uttered
  - $p(\mathbf{O})$ is the partition function that normalizes the posterior estimates.
- Most-likely word sequence can be determined by maximizing

$$W^* = \arg\max_W \underbrace{P[W]}_{LM} \underbrace{p(\mathbf{O}/W)}_{AM}$$

# Acoustic Modeling

# Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and $W$

## Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and $W$
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.

## Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and $W$
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,

## Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and $W$
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,
- Front-end signal processing is performed to minimize the effect of unwanted distortions

# Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and $W$
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,
- Front-end signal processing is performed to minimize the effect of unwanted distortions
- Since speech is a nonstationary signal, $p(\mathbf{O}/W)$ varies with time

## Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and $W$
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,
- Front-end signal processing is performed to minimize the effect of unwanted distortions
- Since speech is a nonstationary signal, $p(\mathbf{O}/W)$ varies with time
- $p(\mathbf{O}/W)$ is estimated using hidden Markov models (HMM)

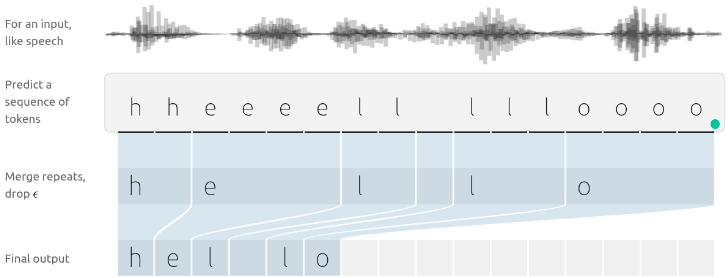
# Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and $W$
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,
- Front-end signal processing is performed to minimize the effect of unwanted distortions
- Since speech is a nonstationary signal, $p(\mathbf{O}/W)$ varies with time
- $p(\mathbf{O}/W)$ is estimated using hidden Markov models (HMM)
- $P[W]$ is estimated using Markov models

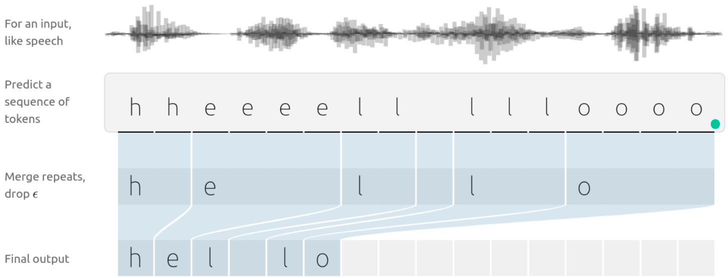
# Acoustic Modeling

- Need to estimate $p(\mathbf{O}/W)$ for all possible pairings of $\mathbf{O}$ and $W$
- Statistical models are employed to compute $p(\mathbf{O}/W)$ on the fly.
- $p(\mathbf{O}/W)$ models the way speaker pronounces the words, the microphone characteristics, channel, ambient noise etc.,
- Front-end signal processing is performed to minimize the effect of unwanted distortions
- Since speech is a nonstationary signal, $p(\mathbf{O}/W)$ varies with time
- $p(\mathbf{O}/W)$ is estimated using hidden Markov models (HMM)
- $P[W]$ is estimated using Markov models
- End-to-end neural network models directly estimate $P[W/\mathbf{O}]$

# Towards End-to-End Speech Recognition



# Towards End-to-End Speech Recognition




- Map acoustic observation sequence $O = (\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T)$ to alphabet sequence $W = (w_1, w_2, \cdots w_U)$, where $W_k \in \{S_1, S_2, \cdots S_{26}\}$
  - The sequences $O$ and $W$ are of different length
  - The ratio of lengths of $O$ and $W$ can vary
  - Do not have access to accurate alignment between $O$ and $W$

## Sequence Model

- Train a model to infer word sequence $W$ from observation sequence $O$
- That is, the model should maximize $P[W/O]$
- During testing, the most likely word sequence can be inferred as

$$W^* = \arg \max_{\text{all } W} P[W/O]$$

# Sequence Model

- Train a model to infer word sequence $W$ from observation sequence $O$
- That is, the model should maximize $P[W/O]$
- During testing, the most likely word sequence can be inferred as

$$W^* = \arg\max_{\text{all } W} P[W/O]$$

- How to proceed with assigning posteriors to word sequences?

## Sequence Model

- Train a model to infer word sequence $W$ from observation sequence $O$
- That is, the model should maximize $P[W/O]$
- During testing, the most likely word sequence can be inferred as

$$W^* = \arg\max_{\text{all } W} P[W/O]$$

- How to proceed with assigning posteriors to word sequences?
  - For every $o_t$, assign a posterior distribution over all possible W

$$P[w_t = S_j/o_t] \qquad j = 1, 2, \cdots 26 \text{ and } t = 1, 2, \cdots T$$

  - Sequence of posteriors can be used to evaluate P[W/O]

## Sequence Model

- Train a model to infer word sequence $W$ from observation sequence $O$
- That is, the model should maximize $P[W/O]$
- During testing, the most likely word sequence can be inferred as

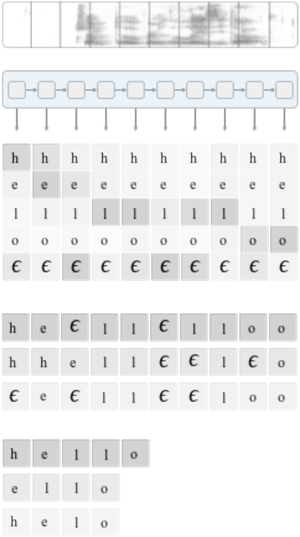
$$W^* = \arg\max_{\text{all } W} P[W/O]$$

- How to proceed with assigning posteriors to word sequences?
  - For every $o_t$, assign a posterior distribution over all possible W

$$P[w_t = S_j/o_t] \qquad j = 1, 2, \cdots 26 \text{ and } t = 1, 2, \cdots T$$

  - Sequence of posteriors can be used to evaluate P[W/O]
- RNNs/CNNs are used to map the observations to word posteriors
  - Cross-entropy loss cannot be used as it requires ground-truth alignment

# Connectionist Temporal Classification (CTC))




We start with an input sequence, like a spectrogram of audio.

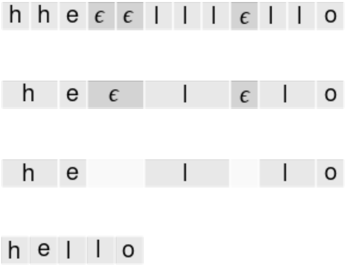
The input is fed into an RNN, for example.

The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, $\epsilon$ } for each input step.

With the per time-step output distribution, we compute the probability of different sequences

By marginalizing over alignments, we get a distribution over outputs.

# CTC Alignment Steps

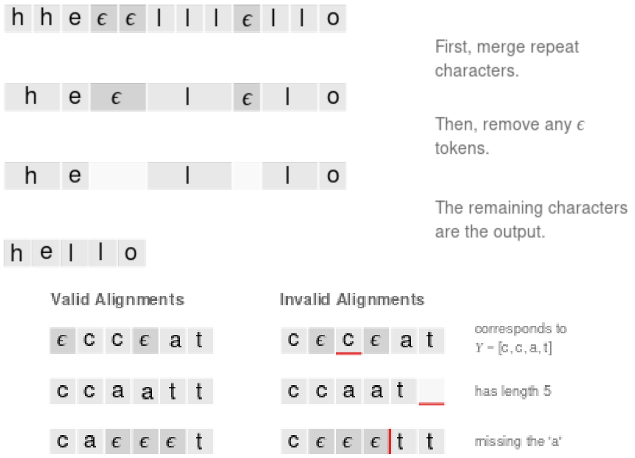



First, merge repeat characters.

Then, remove any $\epsilon$ tokens.

The remaining characters are the output.

# CTC Alignment Steps



## CTC Loss

- Probability of a word sequence W given the observation sequence O

$$P[W/O] = \sum_{\text{all valid paths}} \prod_{t=1}^{T} P[w_t/\mathbf{o}_1, \mathbf{o}_2, \cdots \mathbf{o}_T]$$

- During training, manual transcription of words/sentences is known
  - Restrict output posterior computation to the alphabet in those words
  - Form the trellis by arranging posteriors in the order of alphabet
  - Evaluate the probabilities along all the paths resuting in the given word
  - Compute the gradients, and backpropagate to maximize the probability
- Negative logarithm of the $P[W/O]$ is referred to as CTC loss

$$\mathcal{L}(\theta) = -\frac{1}{\mathcal{B}} \sum_{(O,W)\in\mathcal{B}} P[W/O]$$

# Inference

- Given $\lambda$, evaluate posterior probabilities of alphabet at every time-step

$$P[a_k/o_t] \qquad k = 1, 2, \cdots 26, \forall t$$

# Inference

- Given $\lambda$, evaluate posterior probabilities of alphabet at every time-step

$$P[a_k/o_t] \qquad k = 1, 2, \cdots 26, \forall t$$

- Infer the likely word sequence for the given observation sequence

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

# Inference

- Given $\lambda$, evaluate posterior probabilities of alphabet at every time-step

$$P[a_k/o_t] \qquad k = 1, 2, \cdots 26, \forall t$$

- Infer the likely word sequence for the given observation sequence

$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- Consider most-likely output at each time-step

$$A* = \arg\max_A \prod_{t=1}^{T} p[a_t/\mathbf{O}]$$

# Inference

- Given $\lambda$, evaluate posterior probabilities of alphabet at every time-step

$$P[a_k/o_t] \qquad k = 1, 2, \cdots 26, \forall t$$

- Infer the likely word sequence for the given observation sequence

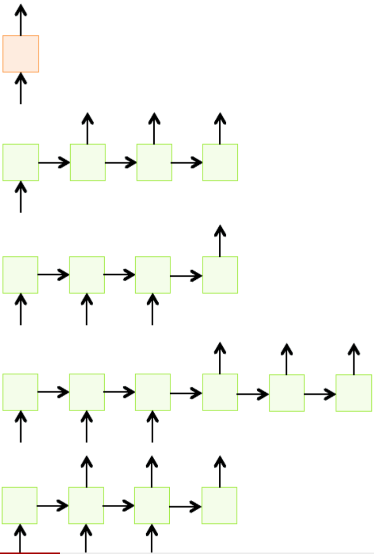
$$W^* = \arg\max_W P[W/\mathbf{O}]$$

- Consider most-likely output at each time-step

$$A* = \arg\max_A \prod_{t=1}^{T} p[a_t/\mathbf{O}]$$

  - It results in alignment with highest probability
  - Collapse the repeats and remove $\epsilon$ to get $W$
  - Works well when most probability mass is allotted to a single alignment

# RNN Configurations



| | |
|---|---|
| Single - Single | Feed-forward Network |
| Single - Multiple | Image Captioning |
| Multiple - Single | Sentiment Classification |
| Multiple - Multiple | Translation |
| | Image Captioning |

# Thank You!