Algorithms for Speech and Natural Language Processing

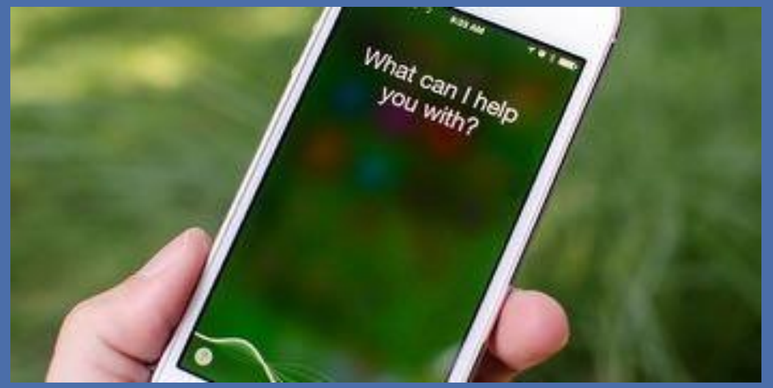
Speech features and Acoustic Models

Emmanuel Dupoux, Robin Algayres (and slides by Neil Zeghidour)

Recommended book:

Speech and Language Processing, volume 2 and 3
by Jurafsky and Martin (free online)

ASR: Automatic Speech Recognition



Personal assistants

Personal assistants

These devices can:

- Identify your voice
- Recognize what you say
- Understand what you mean
- Answer with a natural voice

Personal assistants

These devices can:

- **Identify your voice**
- Recognize what you say
- Understand what you mean
- Answer with a natural voice

Personal assistants

These devices can:

- Identify your voice
- **Recognize what you say**
- Understand what you mean
- Answer with a natural voice

Personal assistants

These devices can:

- Identify your voice
- Recognize what you say
- **Understand what you mean**
- Answer with a natural voice

Personal assistants

These devices can:

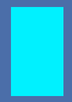
- Identify your voice
- Recognize what you say
- Understand what you mean
- **Answer with a natural voice**

Personal assistants

These devices can:

- Identify your voice: Speaker identification
- Recognize what you say: Speech recognition
- Understand what you mean: Natural Language Processing
- Answer with a natural voice: Speech Synthesis

Personal assistants



These devices can:

- Identify your voice: Speaker identification
- **Recognize what you say: Speech recognition**
- Understand what you mean: Natural Language Processing (chatbot)
- Answer with a natural voice: Speech Synthesis

State of the art in ASR

2016: WER= 6%, human level speech recognition has been reached! (Switchboard corpus)

Microsoft | The AI Blog Our Company News and Stories Press Tools

Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition

October 18, 2016 | Allison Linn

Researchers from the Speech & Dialog research group

Microsoft researchers from the Speech & Dialogue research group include, from back left, Wayne Xiong, Geoffrey Zweig, Xuedong Huang, Dong Yu, Frank Seide, Mike Seltzer, Jasha Droppo and Andreas Stolcke. (Photo by Dan DeLong)

Microsoft has made a major breakthrough in speech recognition, creating a technology that recognizes the words in a conversation as well as a person does.

In a paper published Monday, a team of researchers and engineers in Microsoft Artificial Intelligence and Research reported a speech recognition system that makes the same or fewer errors than professional transcriptionists. The researchers reported a word error rate (WER) of 5.9 percent, down from the 6.3 percent WER the team reported just last month.

The 5.9 percent error rate is about equal to that of people who were asked to transcribe the same conversation, and it's the lowest ever recorded against the industry standard Switchboard speech recognition task.

State of the art in ASR

2016: WER= 6%, human level
speech recognition has been reached!
(Switchboard corpus)

2023: Can you use your voice
assistant in a noisy street/bar/train station?
can you ask a question while somebody
else is talking?
can you speak slang or speak with a
strong accent?

ASR today is still a gadget for most people

Microsoft | The AI Blog Our Company News and Stories Press Tools

Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition

October 18, 2016 | Allison Linn

Researchers from the Speech & Dialog research group

Microsoft researchers from the Speech & Dialogue research group include, from back left, Wayne Xiong, Geoffrey Zweig, Xuedong Huang, Dong Yu, Frank Seide, Mike Seltzer, Jasha Droppo and Andreas Stolcke. (Photo by Dan DeLong)

Microsoft has made a major breakthrough in speech recognition, creating a technology that recognizes the words in a conversation as well as a person does.

In a paper published Monday, a team of researchers and engineers in Microsoft Artificial Intelligence and Research reported a speech recognition system that makes the same or fewer errors than professional transcriptionists. The researchers reported a word error rate (WER) of 5.9 percent, down from the 6.3 percent WER the team reported just last month.

The 5.9 percent error rate is about equal to that of people who were asked to transcribe the same conversation, and it's the lowest ever recorded against the industry standard Switchboard speech recognition task.

What is speech recognition?

RECORDING

TRANSCRIBING

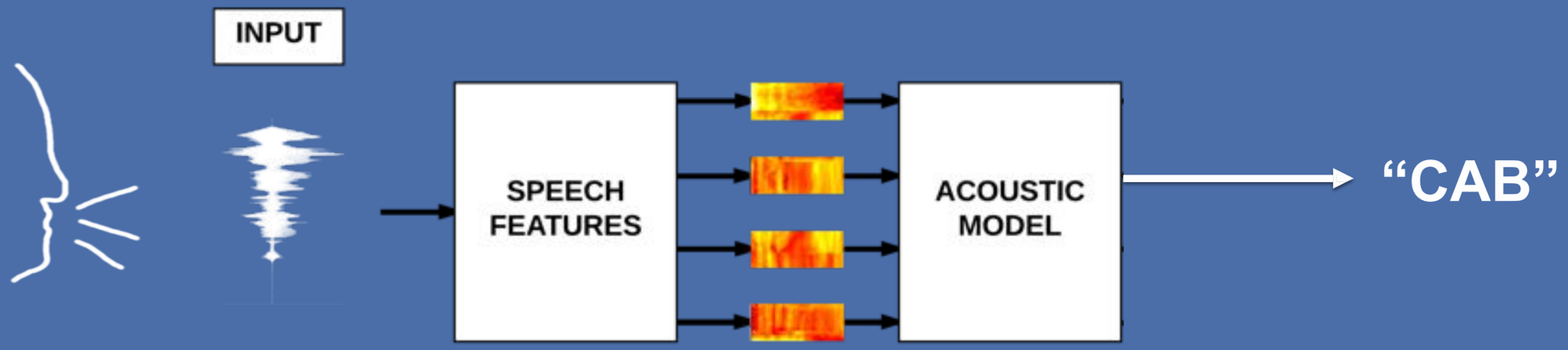
“FIND ME AN ALGERIAN RESTAURANT NEARBY”

Our task: Speech to Text

Outline

I. Anatomy of a speech recognition system
   1. Speech and speech features
   2. HMM model
   3. Single word speech recognition with HMM-GMM model

II. Handling variability in speech
   1. Gender
   2. Speaker identity
   3. Noise



Speech and speech features

- The speech waveform
- Spectrogram
- Speech production
- Matching human perception

Some reminders on sound waves:

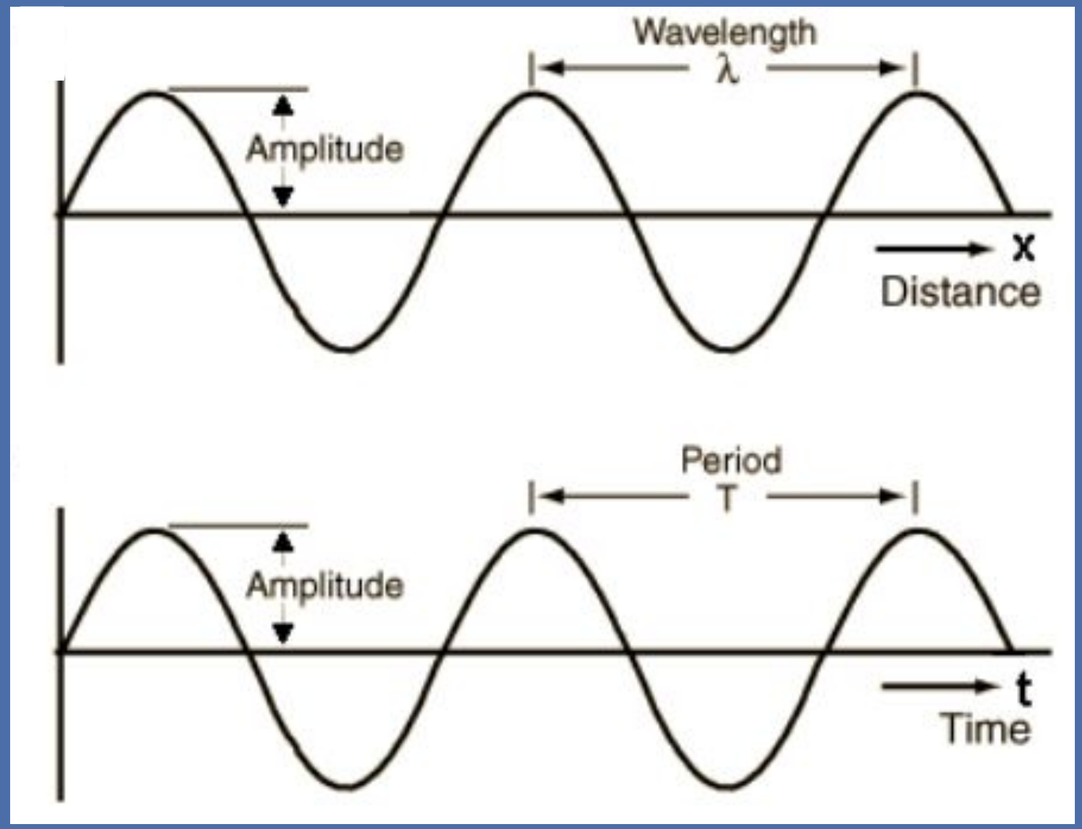
3D wave

the wave length is a distance

the period is a duration

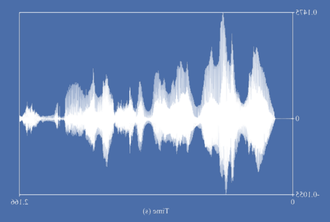
Hertz = nb of vibration per seconds

frequency = 1 / T



What does a microphone record ?

- A microphone measures amplitude of the variation in air pressure (in Pascal, coded in 16bits)
- Sounds is a continuous phenomenon discretized at a given sample rate
- Example with sampling rate=16kHz:



[-0.10, 0.01, -0.02,…, 0.05]

3 seconds * 16 000 Hz = vector of length 48000

- What sampling rate should we choose for ASR ?

- **Nyquist frequency**: highest detectable frequency = half of the sampling rate
- Human hearing frequencies: between 20Hz and 20kHz
- Speech production frequencies: mostly < 5kHz
- Therefore:
  music: sampling rate of 48kHz
  speech: sampling rate of 16kHz (and mobile phones: 8kHz)

Pascal and dB:

Sound intensity is measured not in Pascal (Pa) but in decibels (dB)

1 dB = $10*\log_{10}(P/P_0)$ (P_0= threshold of hearing)

Why decibel:

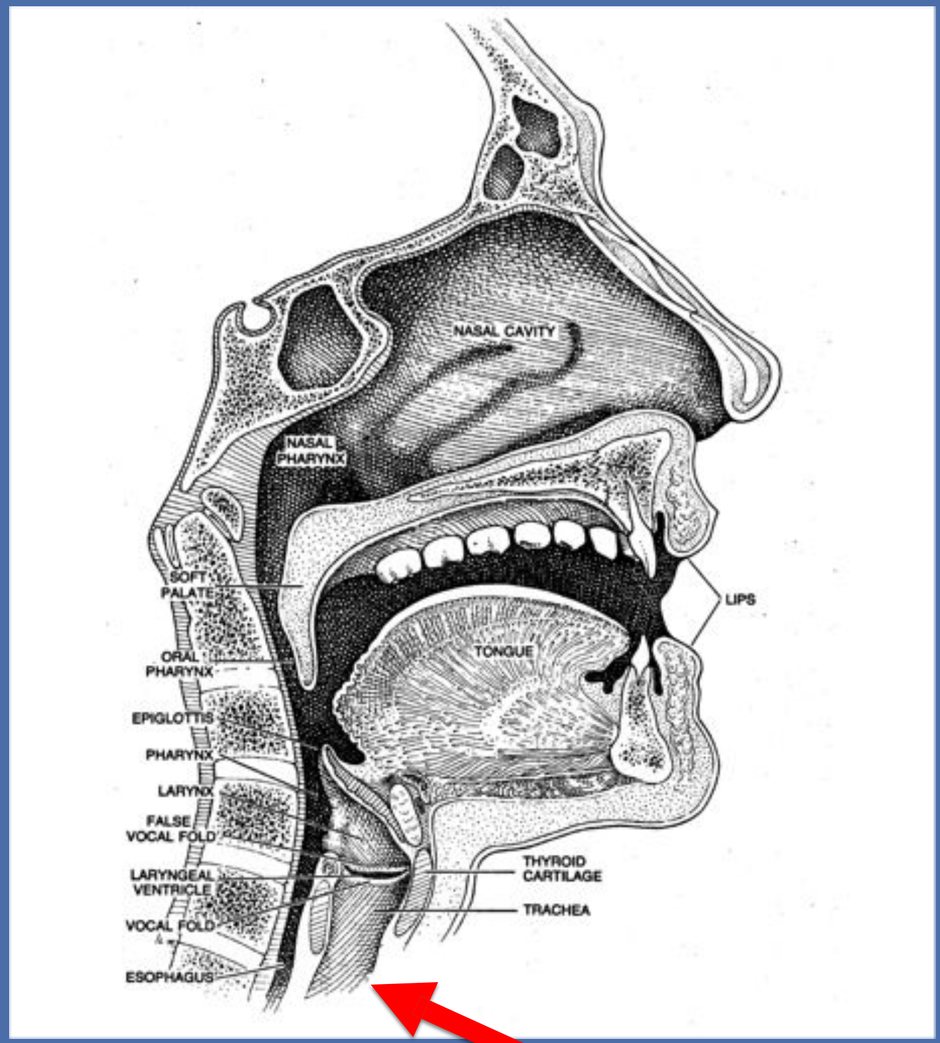
decibels are correlated (i.e linear increase) with human perception of loudness

"Double the decibels sounds twice as loud"

A mosquito flying 3 m away	0 dB	Somebody shouting	100 dB
Softest audible 1000 Hz sound	6 dB	Pneumatic drill	100 dB
Quiet living room	20 dB	Helicopter	110 dB
Soft whispering	25 dB	Loud rock concert	110 dB
Refrigerator	40 dB	Threshold of pain	120 dB
Soft talking	50 dB	Air raid siren	130 dB
Normal conversation	60 dB	Gunshot	140 dB
Busy city street noise	70 dB	Instant perforation of eardrum	160 dB
Passing motorcycle	90 dB	Rocket launch	180 dB

Speech production

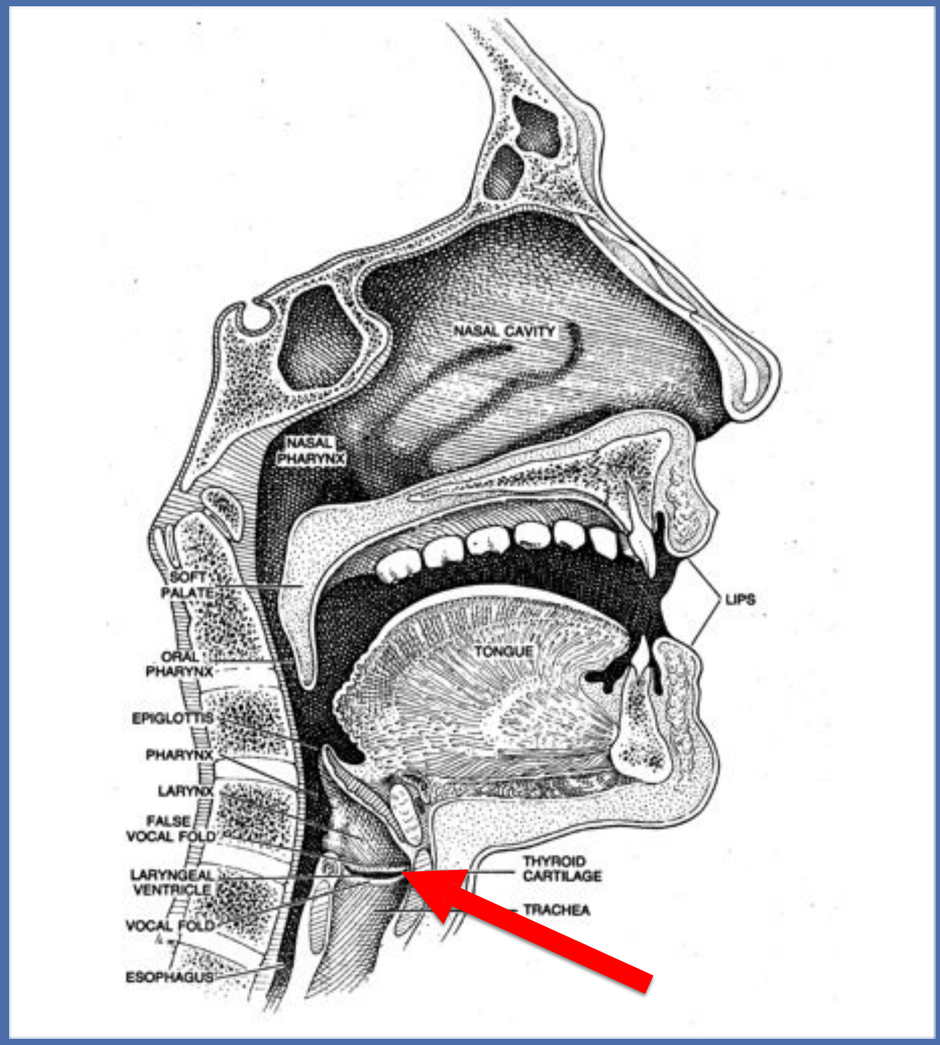
- Air is expelled from the lungs through the trachea



From Sundberg (1977)

Speech production

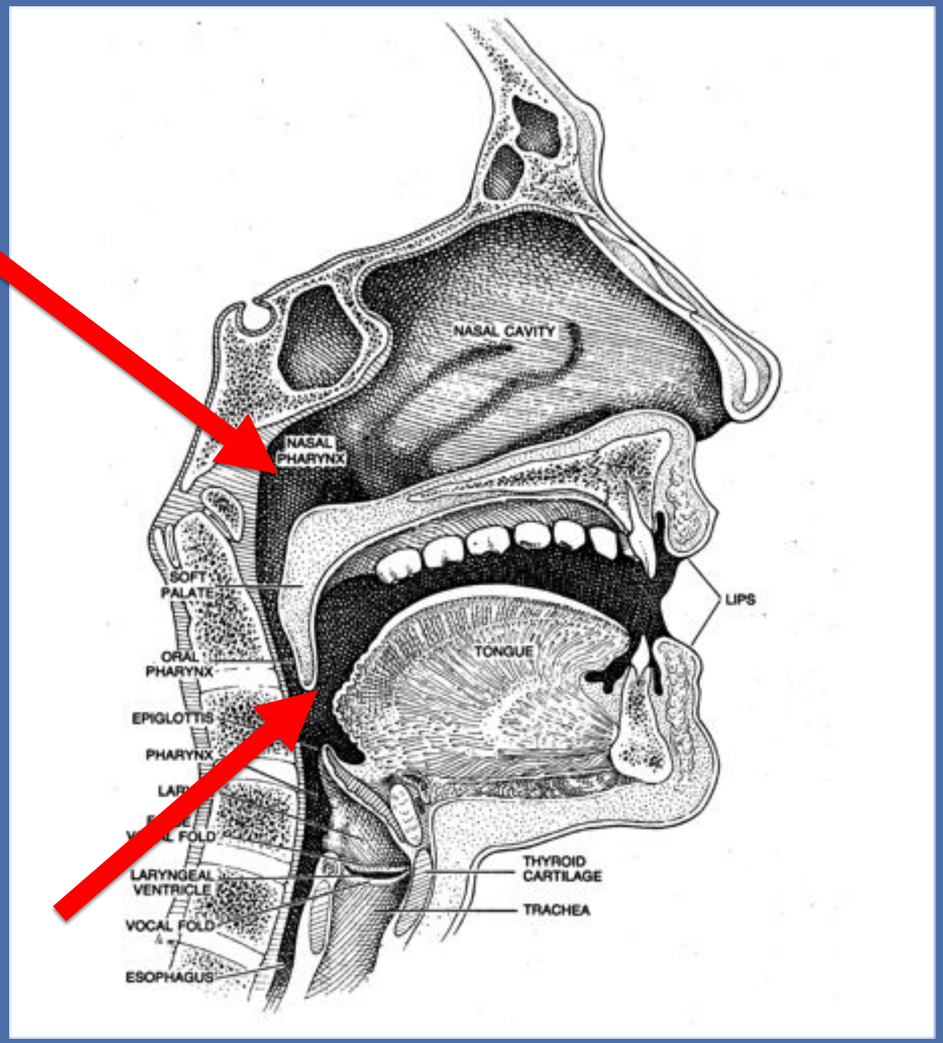
- Air is expelled from the lungs through the trachea and resonate in the larynx
- Passes through vocal cords



From Sundberg (1977)

Speech production

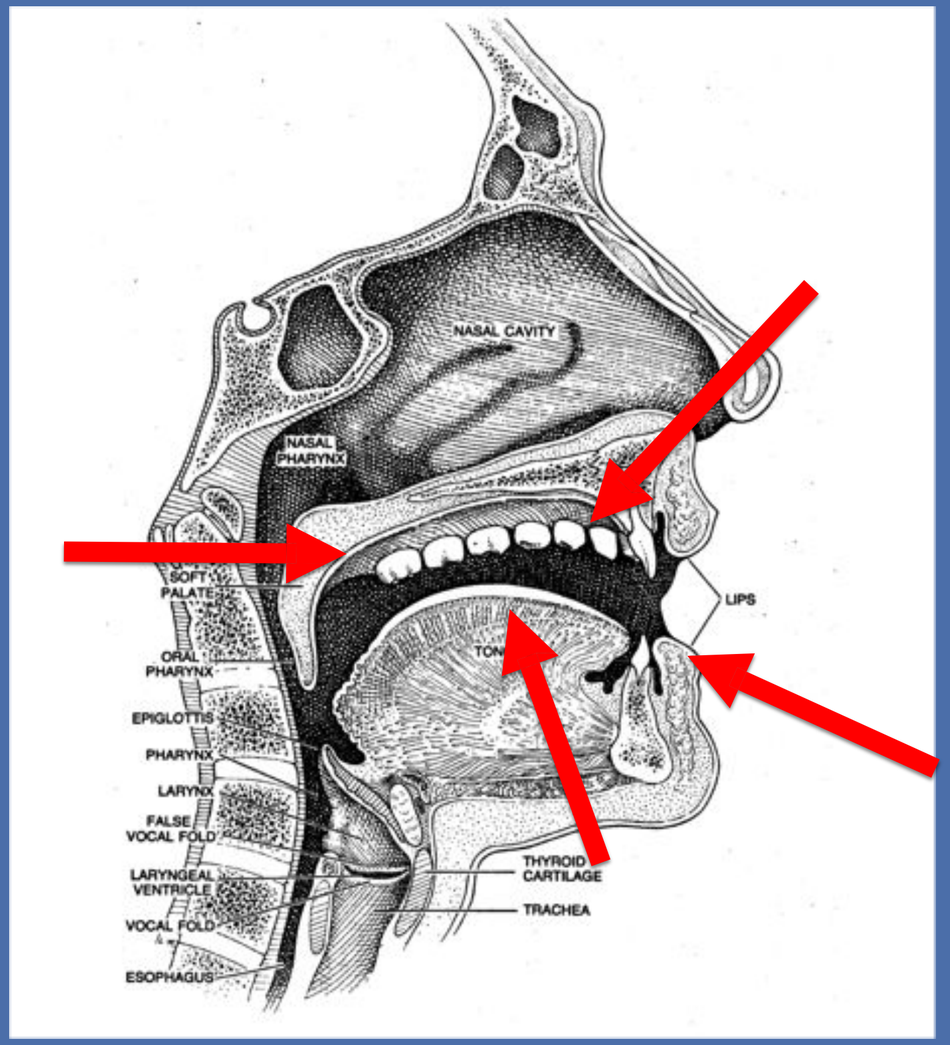
- Air is expelled from the lungs through the trachea
- Passes through vocal cords and resonate in the larynx
- Then through the nose and/or the mouth



From Sundberg (1977)

Speech production

- Air is expelled from the lungs through the trachea
- Passes through vocal cords and resonate in the larynx
- Then through the nose and/or the mouth
- Vocal tract: the « articulators »: lips, tongue, teeth, palate, etc. The way we control the articulators defines the sound we produce.

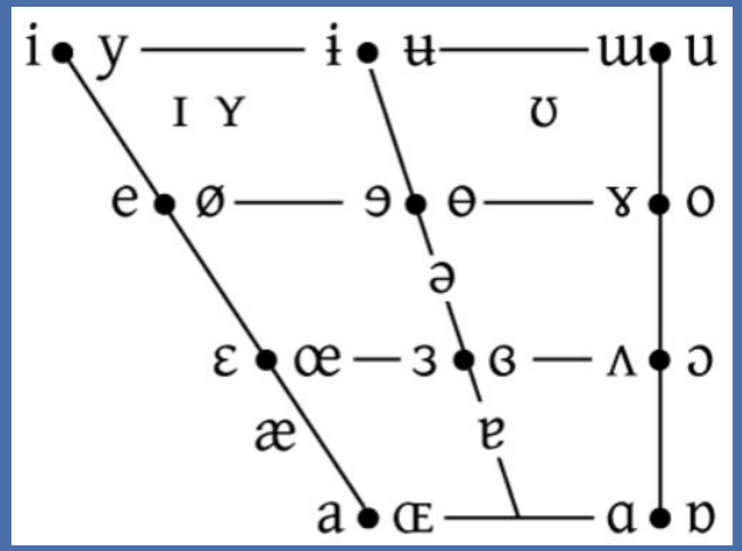


From Sundberg (1977)

Phonemes, phones, vowels and consonants

Phones: any speech sounds (independant of a language)
Phonemes: speech sounds that can discriminate words of a specific language
ex: ‘r’ and ‘l’ are phonemes in English and French but not in Japanese

Phones are usually vowels or consonants



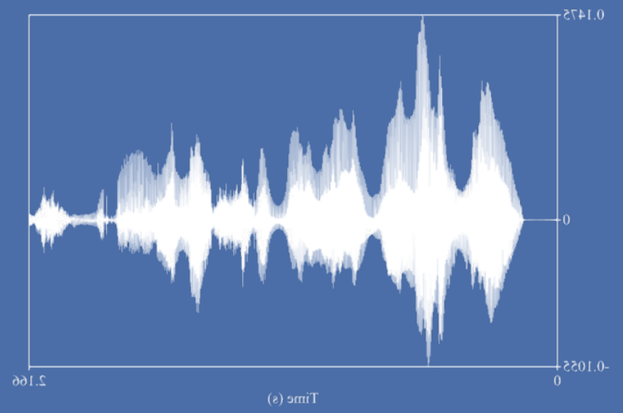
CONSONANTS (PULMONIC)

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap				ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Where symbols appear in pairs, the one to the right represents a voiced consonant. Shaded areas denote articulations judged impossible.

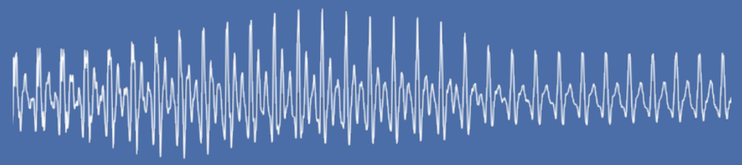
Speech features

“Better” representation of speech than raw signal
Make the job of subsequent ASR model easier



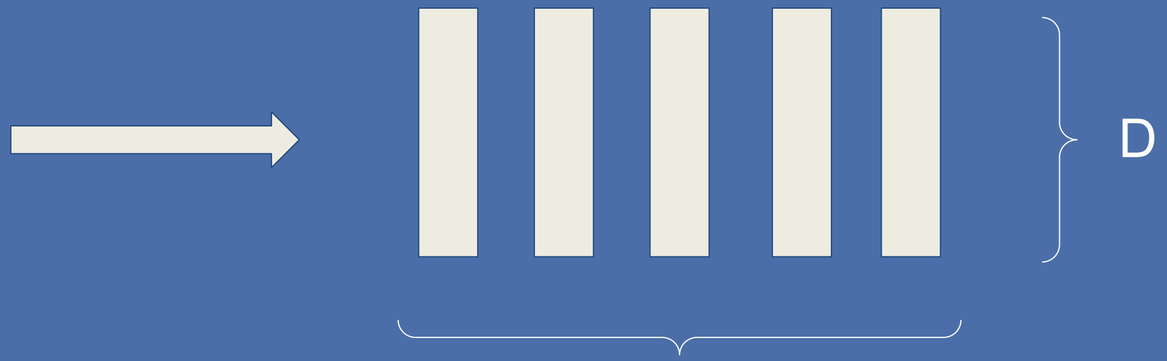
Speech features: definition and properties

Definition: Speech features are the output of a function from $\mathbb{R}^{T}$ into $\mathbb{R}^{T'xD}$



[-0.10, 0.01, -0.02,…, 0.05]

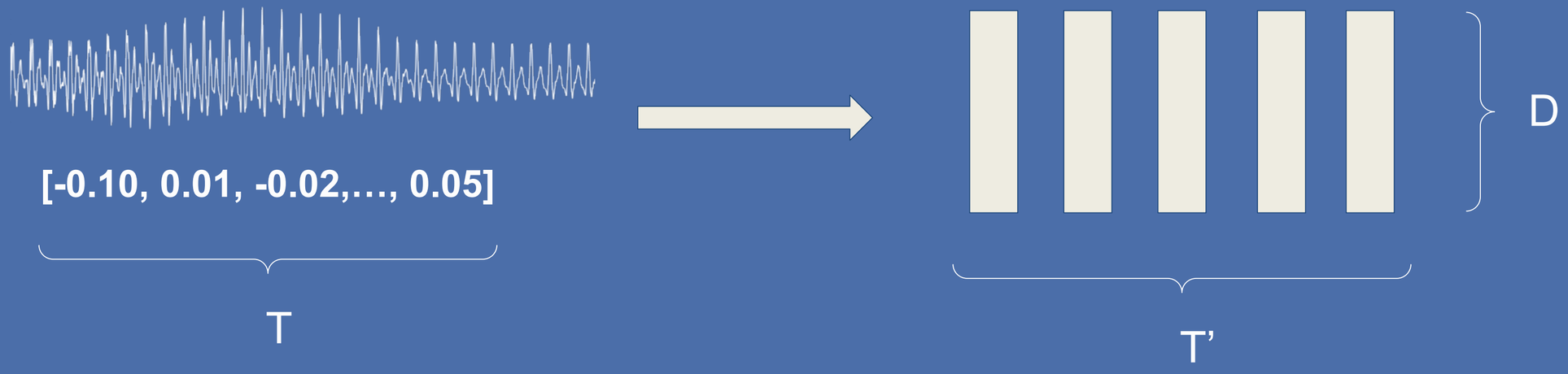
T



Each vector is called a frame

Speech features: definition and properties

Definition: Speech features are the output of a function from $\mathbb{R}^T$ into $\mathbb{R}^{T'xD}$



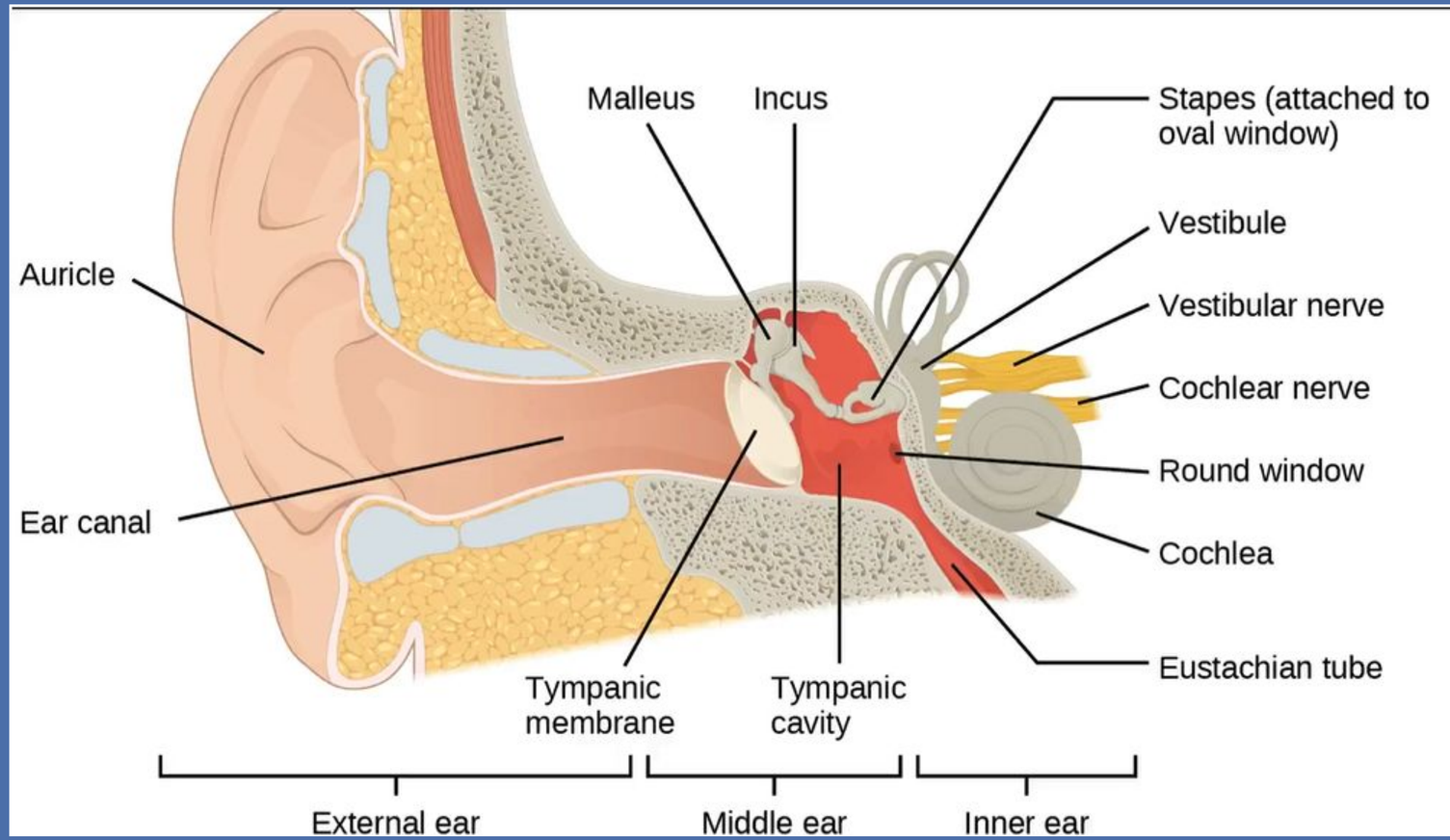
Each vector is called a frame

Properties:

1) T'<<T (necessary for any causal model)
2) locality: each frame represent a small portion of the speech signal
3) invariance: 'similar' speech sounds have similar vector representations

How to encode speech?

The cochlea has a range of sensors that reacts at the beginning to high frequencies (20kHz) and at the end to low frequencies (20Hz)

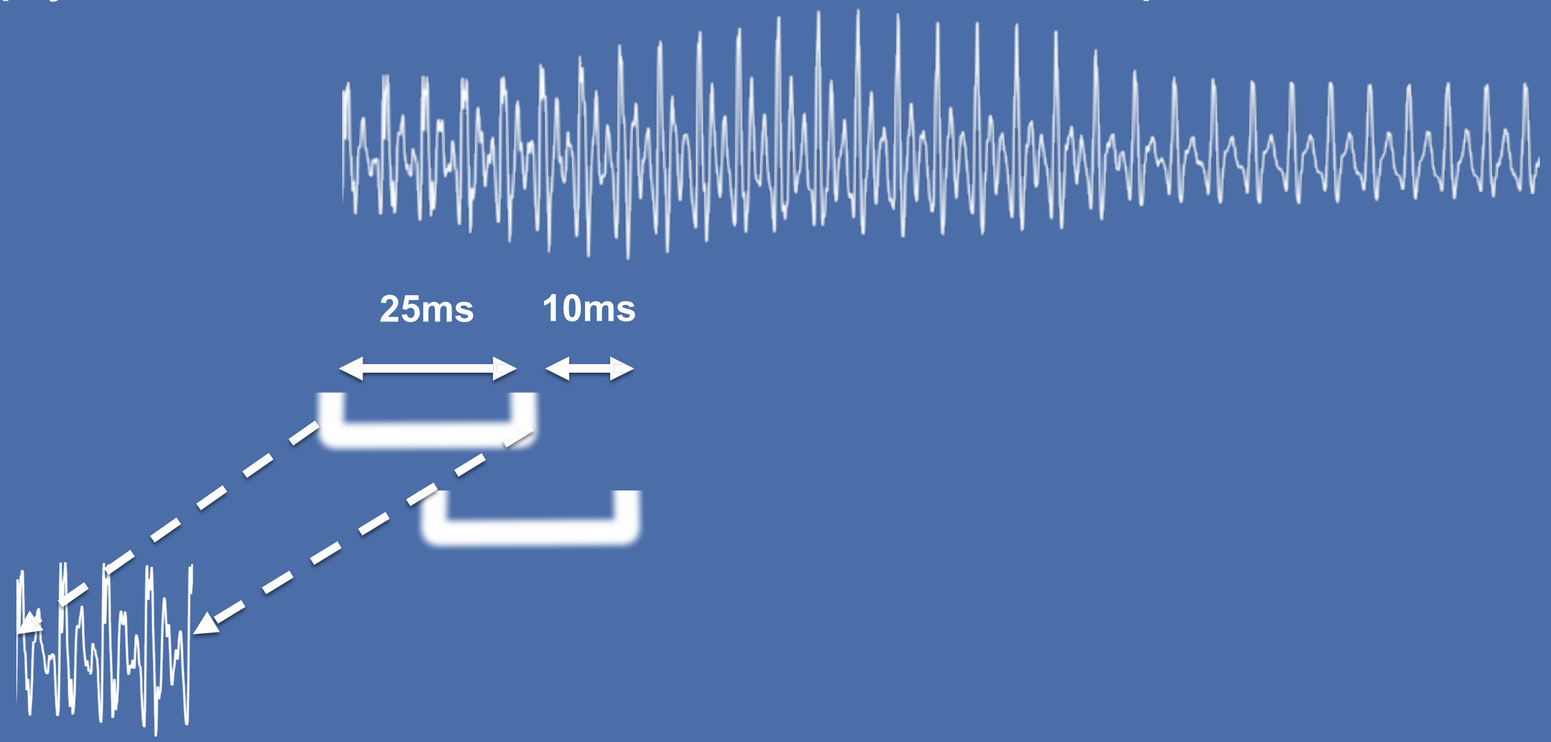


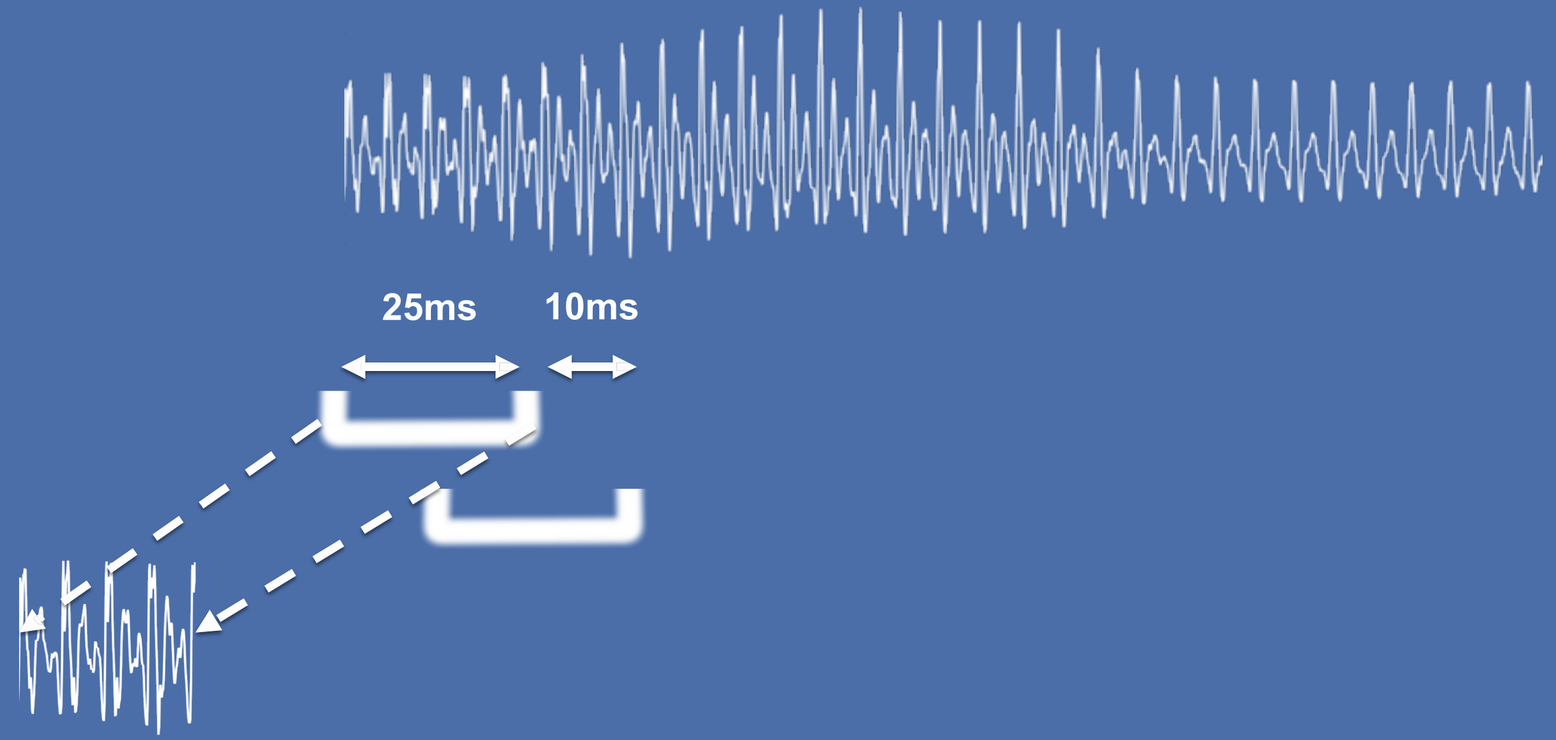
Main idea for speech features: spectral analysis

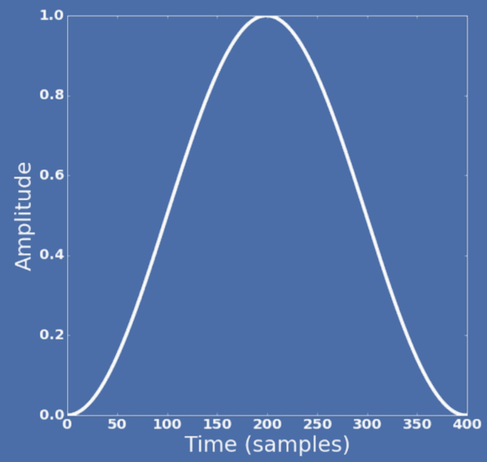
Speech is a vibration of local chords: locally periodic
Can be decomposed as a sum of frequencies !
Apply Fourier series on each small chunk of speech

Main idea for speech features: spectral analysis

Speech is a vibration of local chords: locally periodic
Can be decomposed as a sum of frequencies !
Apply Fourier series on each small chunk of speech





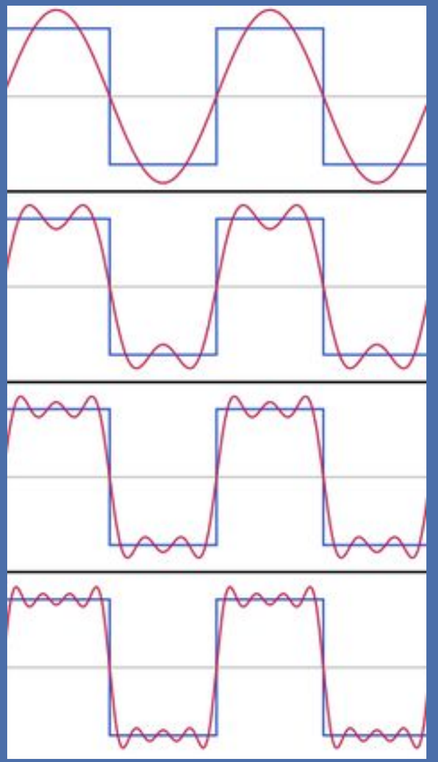


Speech chunks are not really periodic,
apply Hanning window to smooth both ends

$$w[n] = 0.54 - 0.46cos(\frac{2\pi n}{N-1})$$

Fourier Series

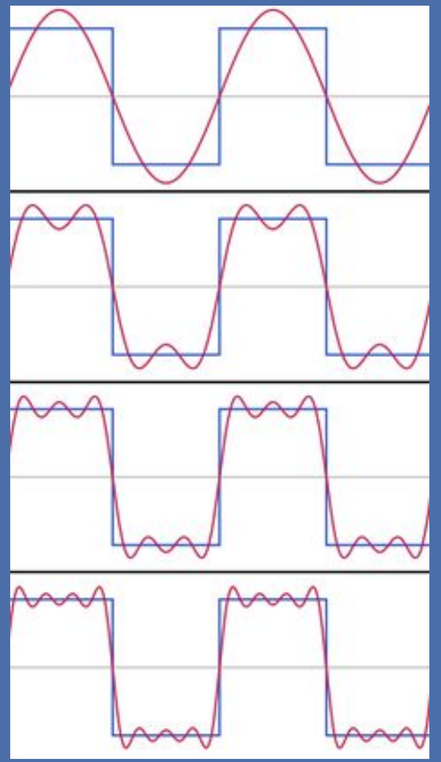
Periodic function with frequency f, can be expressed by the sum of sine and cosine waves whose frequencies are multiple of f. These waves are called the harmonics



Fourier Series

Periodic function with frequency f, can be expressed by the sum of sine and cosine waves whose frequencies are multiple of f. These waves are called the harmonics

Let us take all sine waves in the Human voice range,
and compute how present they are in a speech chunk



The Discrete Fourier Transform (DFT)

- How present is a frequency in a speech chunk? dot product!

$$X(\omega) = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi\omega\frac{n}{N}}$$

← a sine wave

↑ Frequency

↑ Waveform value over the 25ms chunk

The Discrete Fourier Transform (DFT)

- How present is a frequency in a speech chunk? dot product!

$$X(\omega) = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi\omega\frac{n}{N}}$$

← a sine wave

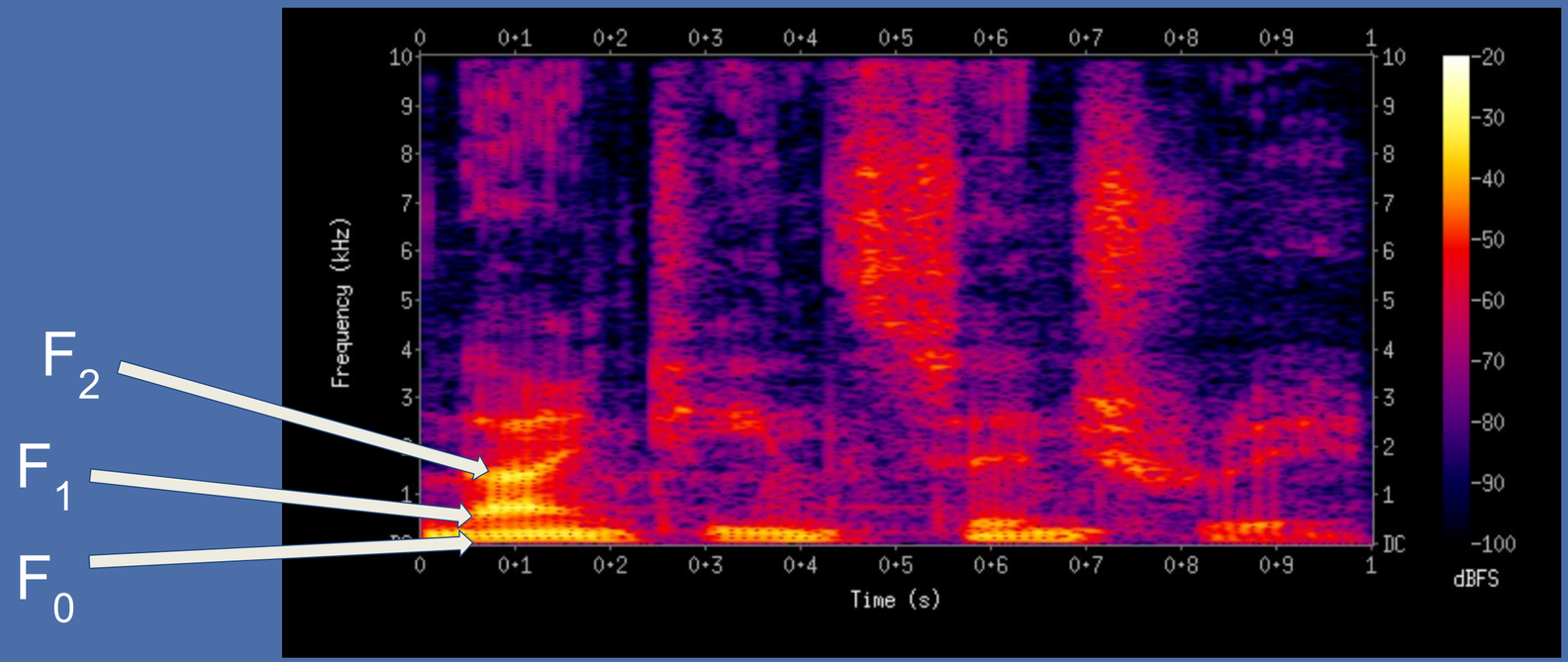
Frequency

Waveform value over the 25ms chunk

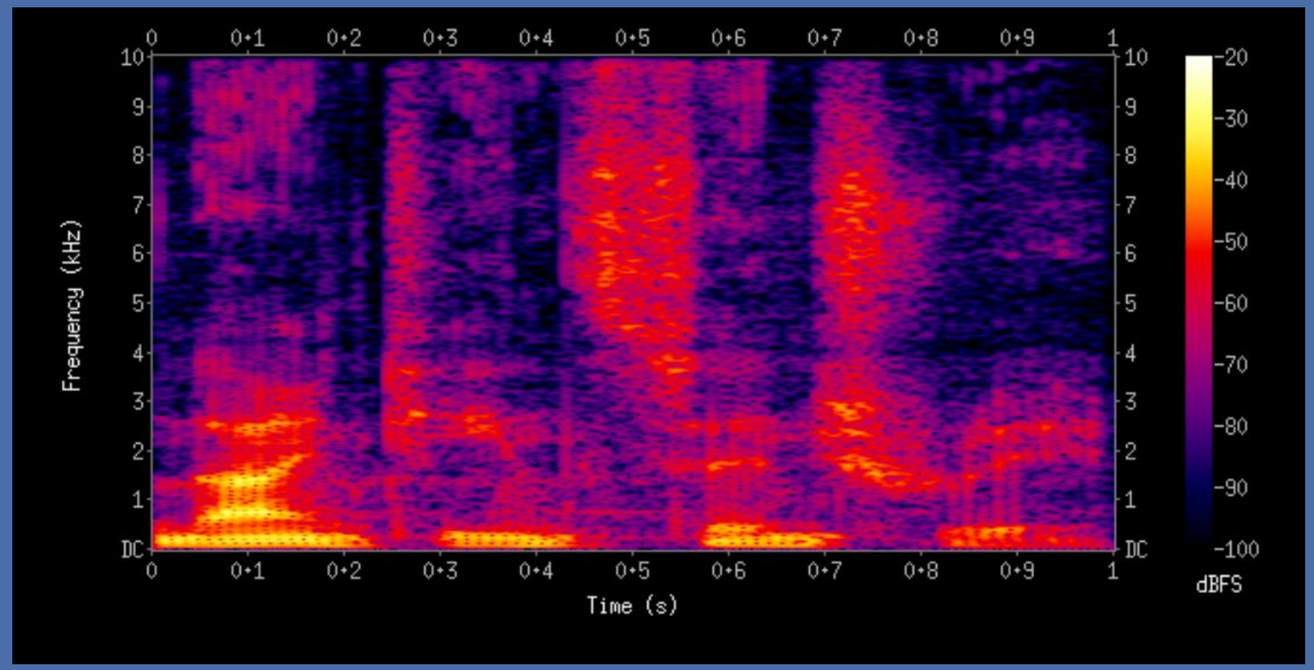
- Complex value, so we take the modulus of it (i.e the power of a frequency)
- For each chunk: do this for all frequencies in human voice range
- In practice, 256 frequencies evenly space in [0, samplerate/2]
- The phase is ignored !

DFT on a speech utterance: spectrogram

The spectrogram is the power of all frequencies in all speech chunks
The harmonics are the most powerful frequencies
F0 is the fundamental (or the pitch) and F1,F2 are the formants

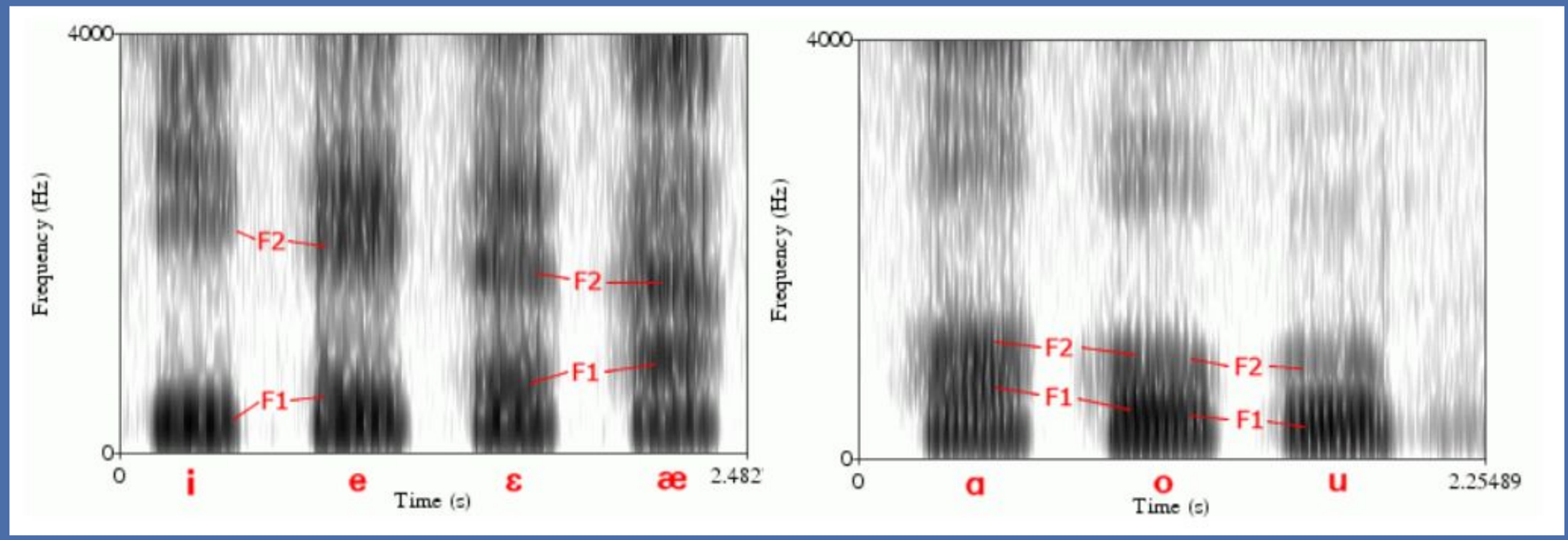


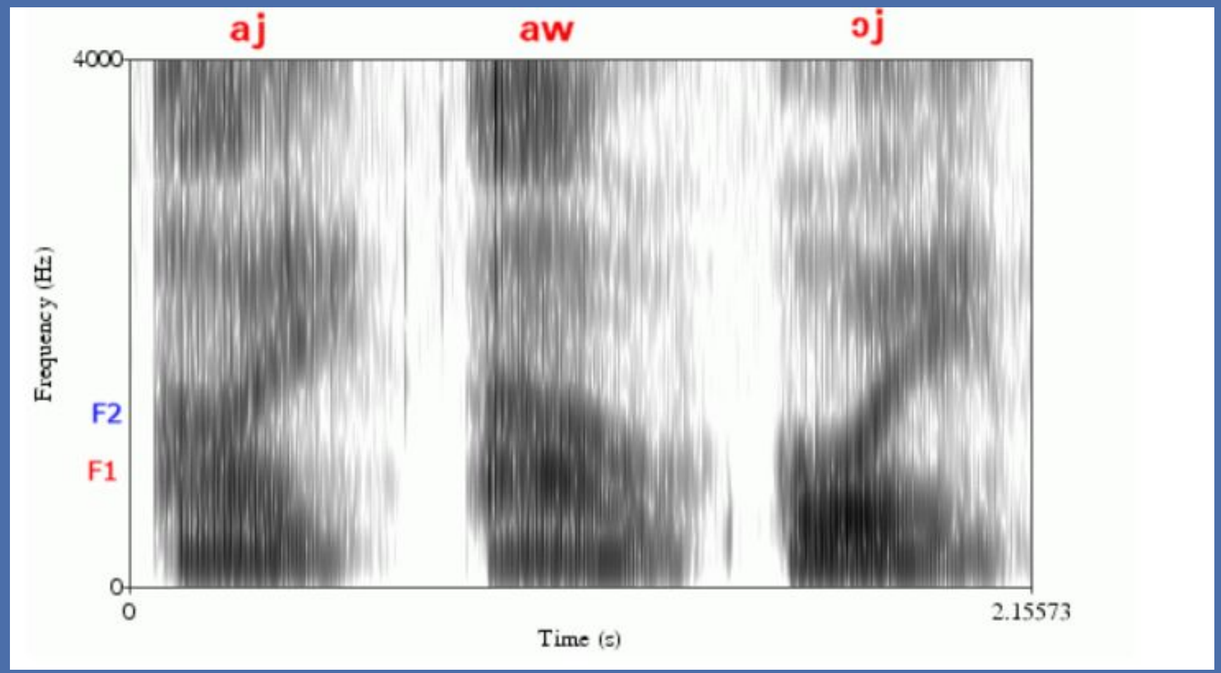
Some sounds have an harmonic ‘signature’
You can learn to ‘read’ a spectrogram



Example of harmonic vowels and diphthongs

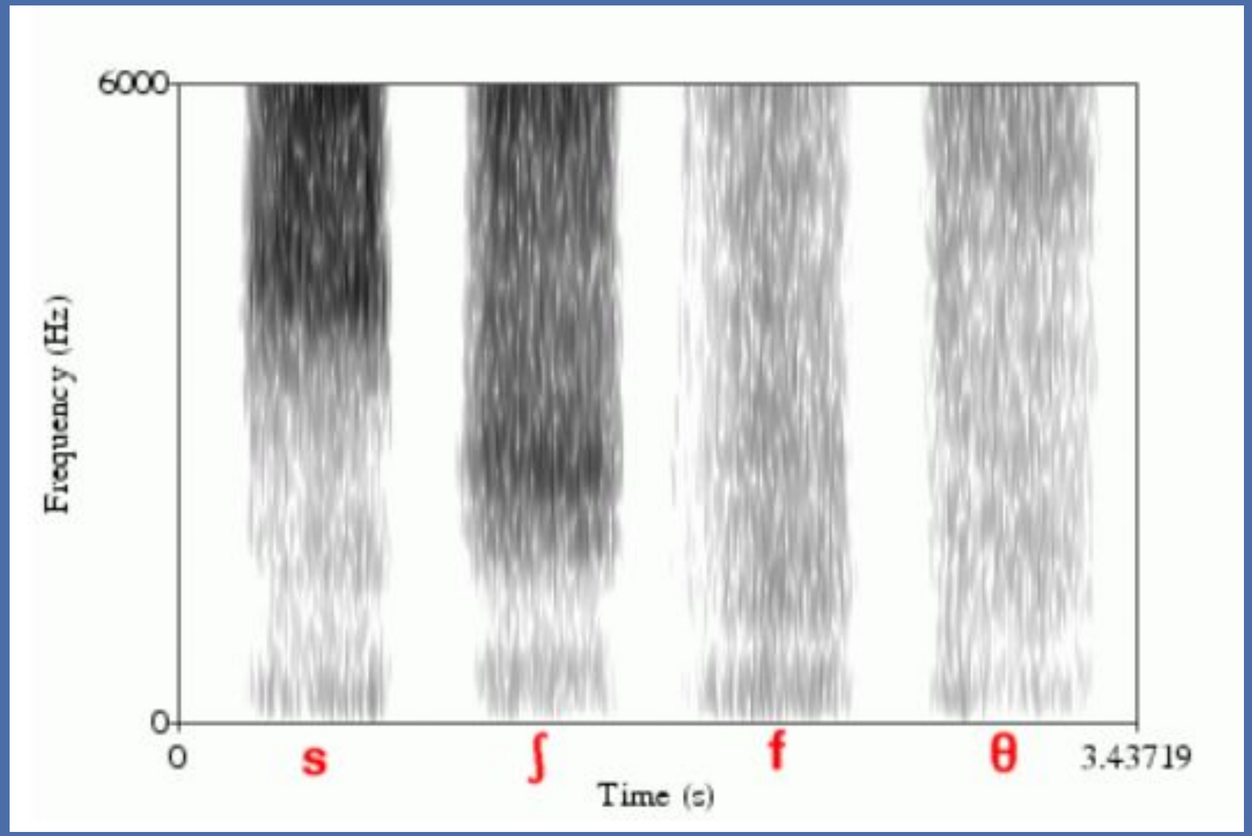
The harmonics are created by the vocal cords and filtered but the vocal tract





Unvoiced phones

Some phones are not voiced: vocal cords vibrate weakly , harmonics are harder to detect



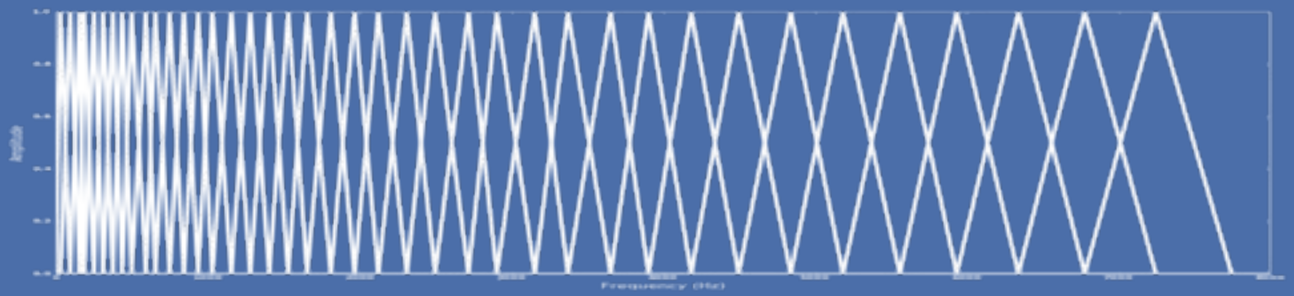
Can we make better features than a spectrogram ?

Main ideas:
- make spectrogram closer to human perception of sounds
- we want less dimensions (currently 256)

Matching human perception: the mel-filterbanks

- Human ear more sensitive to variations in low frequencies than in high frequencies

  ⇒ We warp the spectrogram to a new scale that gives more importance to low frequencies



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

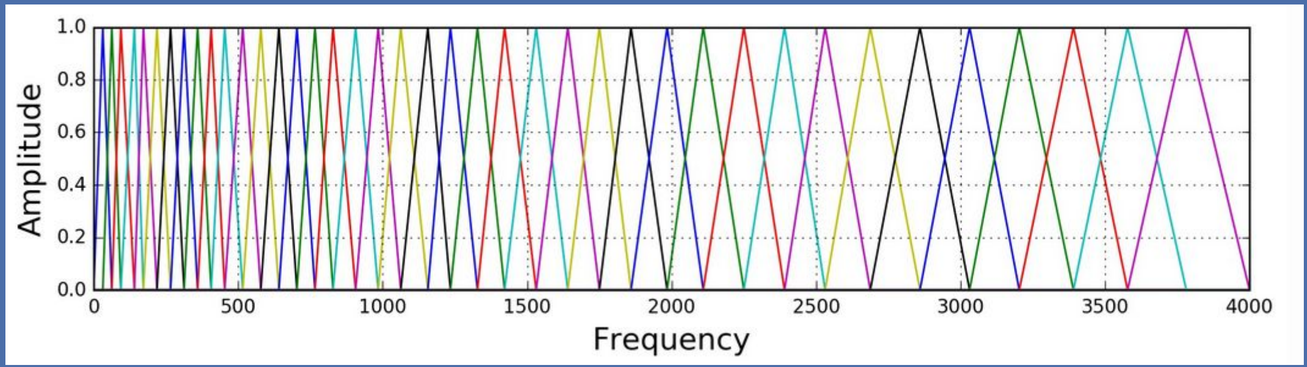
Mel-scale

Linear scale

Matching human perception: the mel-filterbanks

- Human ear more sensitive to variations in low frequencies than in high frequencies

  ⇒ We warp the spectrogram to a new scale that gives more importance to low frequencies



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

Mel-scale

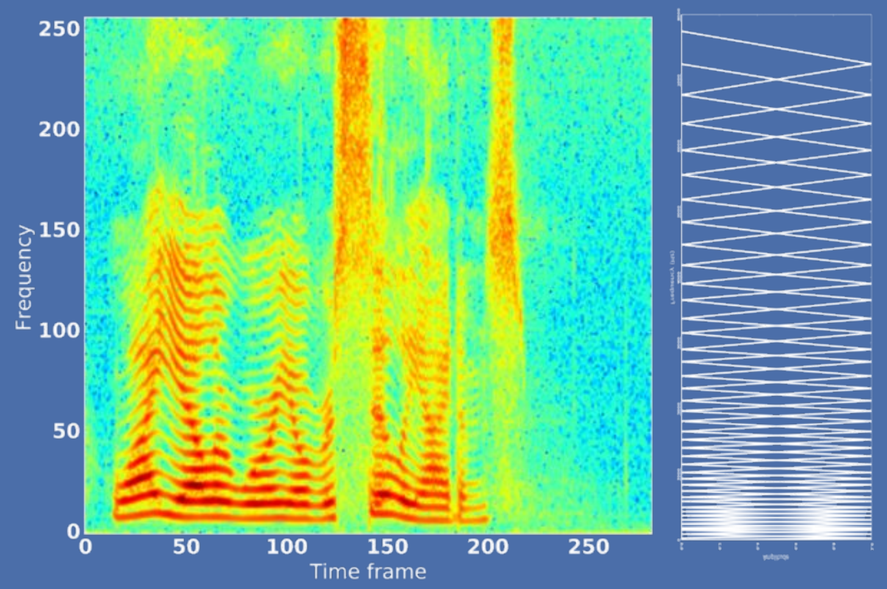
Linear scale

- Filters are linearly spaced in the mel scale
- We compute the product of our spectrum with each filter, then sum the results -> our features are now a vector of length n = number of filters

Matching human perception: the mel-filterbanks

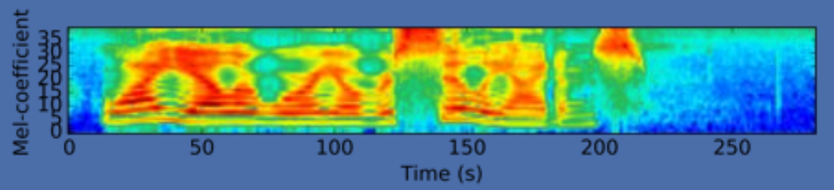
- ~6x fewer dimensions than spectrograms

$$Melfbank_j(k) = \sum_{\omega=0}^{256} Spectrogram(k,\omega) Melfilter_j(\omega)$$



Each frame has 256 dimensions

Mel-averaging

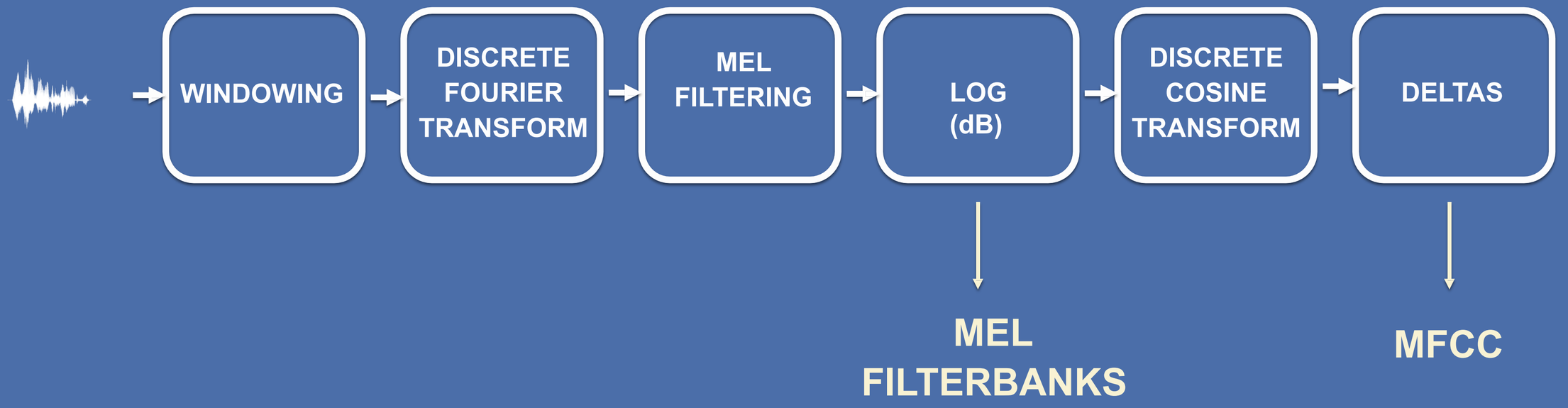


Each frame has 40 dimensions

Matching human perception: logarithmic compression

- Huge range of variations in energy (loudness), but the ear has a logarithmic sensitivity to loudness

From mel-filterbanks to MFCC



From mel-filterbanks to MFCC: the source-filter model

- Speech is the vibration of the glottal source (vocal cords that resonate in the larynx,F_0) that is **filtered** by the vocal tract

From mel-filterbanks to MFCC: the source-filter model

- Speech is the vibration of the glottal source (vocal cords that resonate in the larynx,F_0) that is **filtered** by the vocal tract
- Filtering is mathematically modelled by a convolution:

$$x[n] = s[n] * v[n]$$

- s: vocal cords , v: vocal tract

From mel-filterbanks to MFCC: the source-filter model

- Speech is the vibration of the glottal source (vocal cords that resonate in the larynx,F_0) that is **filtered** by the vocal tract
- Filtering is mathematically modelled by a convolution:

$$x[n] = s[n] * v[n]$$

- s: vocal cords , v: vocal tract
- but the glottal source do not bring much information on the phones being said
- Can we deconvolve the signal and remove s ?

From mel-filterbanks to MFCC: the source-filter model

- Convolution theorem: the fourier transform of two convolved functions is the point wise product of their fourier transform

$$X[\omega] = S[\omega]V[\omega]$$

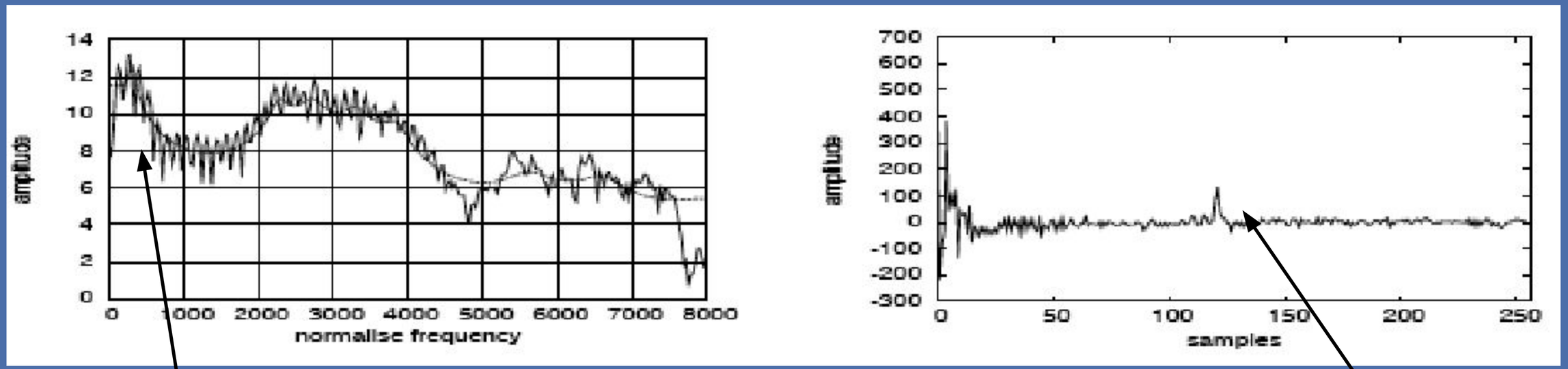
$$|X(\omega)|^2 = |S(\omega)|^2|V(\omega)|^2$$

$$\log(|X(\omega)|^2) = \log(|S(\omega)|^2) + \log(|V(\omega)|^2)$$

- The log spectrum is a sum of two signal, how do we separate two summed signals? … FFT

Cepstrum: FFT of log(FFT)

Log-spectrum=log(FFT(signal)) —FFT→ Cepstrum=FFT(log(FFT(signal)))

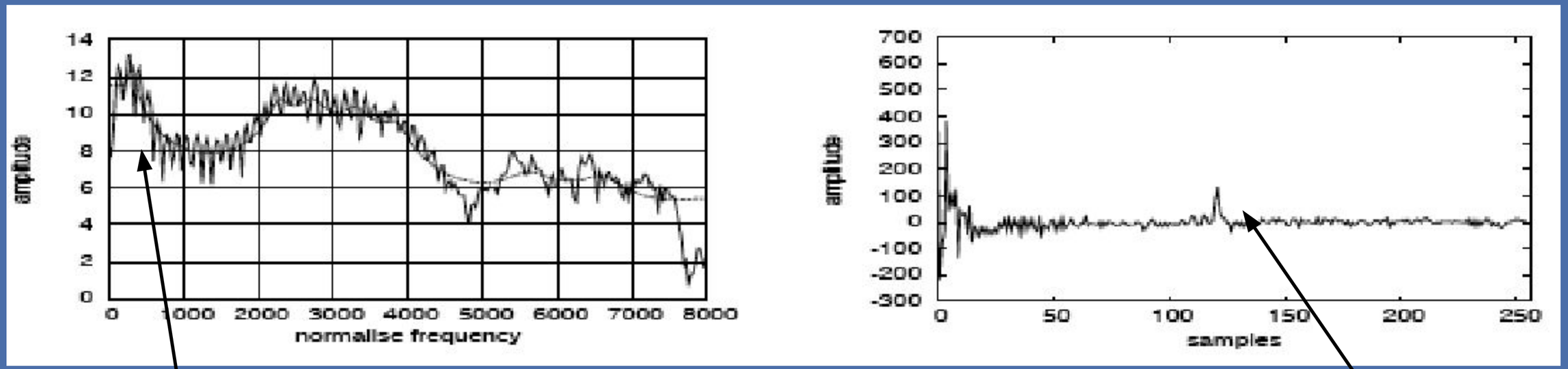


F_0 (pitch)

F_0 (pitch)

Cepstrum: FFT of log(FFT)

Log-spectrum=log(FFT(signal)) —FFT→ Cepstrum=FFT(log(FFT(signal)))

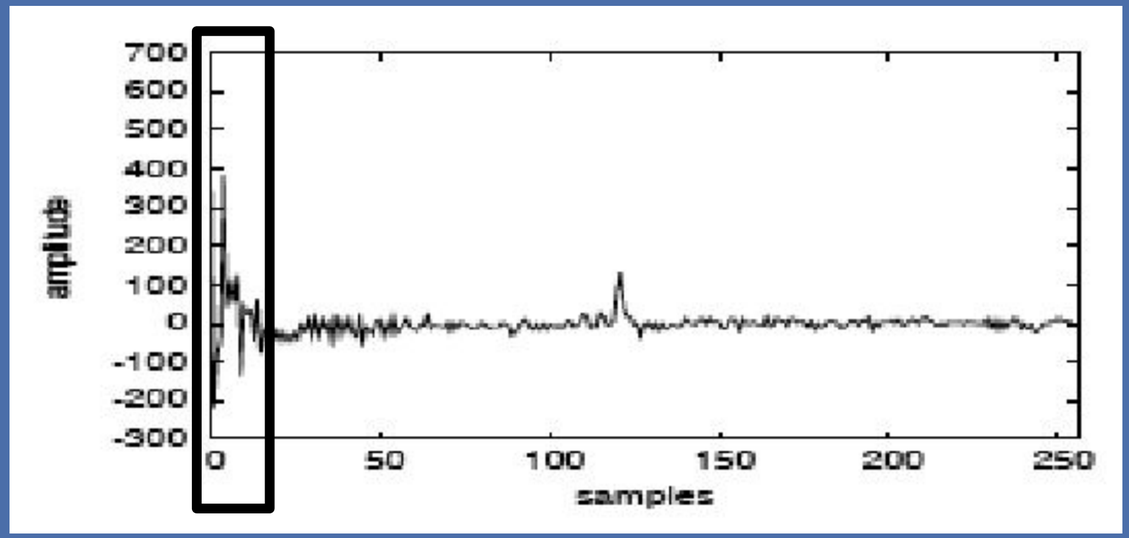


F_0 (pitch)

F_0 (pitch)

- Projects the signal from the spectral to the **cepstral** domain, where vocal cords and vocal tract are well separated
- Important: outputs uncorrelated coefficients (FFT projects in orthonormal base)
- In practice we use Discrete Cosine Transform

The Mel-Frequency Cepstral Coefficients



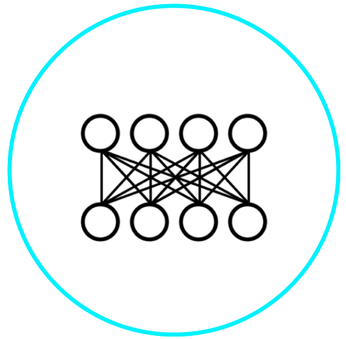
- Take the first 12 cepstral coefficients (glottal source)
- Concatenate the log energy (13 coefficients)
- Mean normalisation along time
- Concatenate first and second order derivatives (along time)
- 39-dimensional coefficients

$$\Delta(n) = \frac{c(n+1) - c(n-1)}{2}$$

$$\Delta\Delta(n) = \frac{\Delta(n+1) - \Delta(n-1)}{2}$$

Recap: Mel filterbanks and MFCC

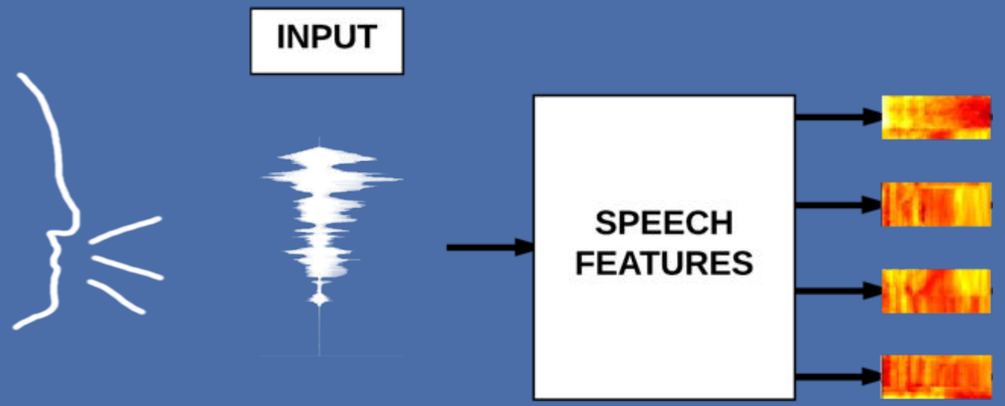
Given a speech utterance
1) Frame: divide into 25ms chunks + hanning window
For each frame:
    2) compute spectral content with DFT (256 coefs)
    3) Filter banks: mel scale (40 coefs)
    4) log to amplitude
    if MFCC:
        5) DCT (13 coefs)
    6) Mean normalisation along time
    if MFCC:
        7) Concatenate with delta and delta-delta (39 coefs)



Acoustic modelling

- The three problems of Hidden Markov Models
- Gaussian Mixture Models

Speech recognition as a statistical problem



“MY NAME IS NEIL”

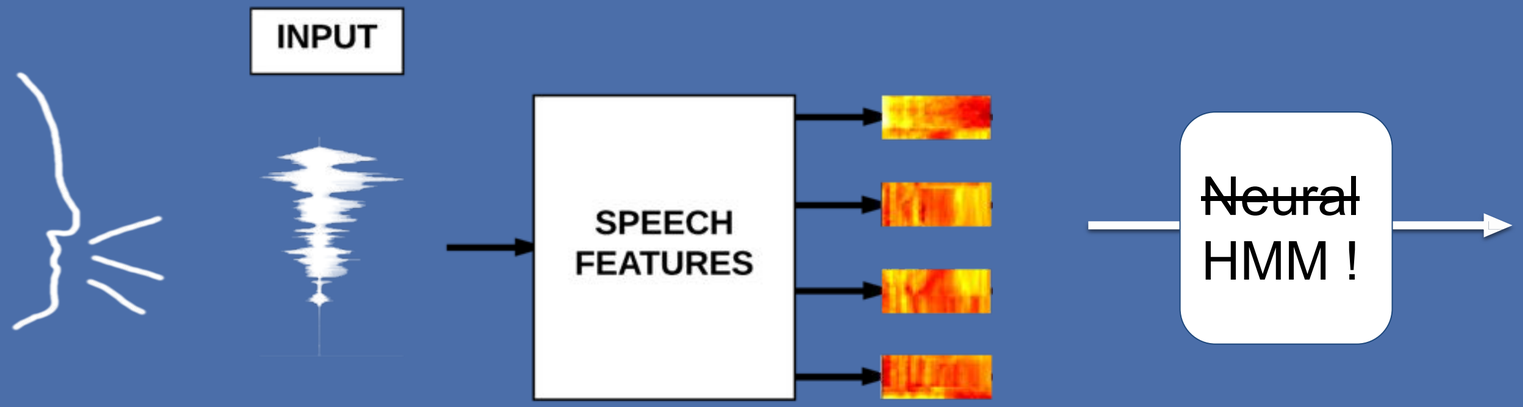
$X_T = x_1, ..., x_T$

A sequence of features frames

$W_M = w_1, ..., w_M$

A sequence of words

Speech recognition as a statistical problem



$$X_T = x_1, ..., x_T$$

A sequence of features frames

$$W_M = w_1, ..., w_M$$

A sequence of words

Speech recognition as a statistical problem

$$X_T = x_1, ..., x_T$$

A sequence of features frames

$$W_M = w_1, ..., w_M$$

A sequence of words

- Goal of ASR:

$$\hat{W} = \arg\max_W P(W|X)$$

Dividing the problem

$$\hat{W} = \arg\max_{W} P(W|X)$$

- By Bayes formula:

$$\hat{W} = \arg\max_{W} P(X|W)P(W)$$

ACOUSTIC MODEL **LANGUAGE MODEL**

We omit P(X) as we maximize over W.

ASR database

Speech database: Speech sequences and transcription

Not fully supervised: no one-to-one mapping between speech and labels

seq2seq RNN ? need big database and cannot exploit human knowledge.

We have a lot of prior knowledge: phonemes, different pronunciations of words, typical formant structures, biological difference between gender,...

ASR database

Hidden Markov Model (HMM) !
Can be trained on small databases by injected **a lot of** human prior knowledge

Widely used until 2016

Remark on Hidden Markov Models (HMMs)

- HMMs models are complex (lots of moving parts), not neural, not end to end, full of human expert knowledge and are disappearing from conference papers, why should I study this?
  - some parts are still used in production
  - much lighter than DNN, can be used on device

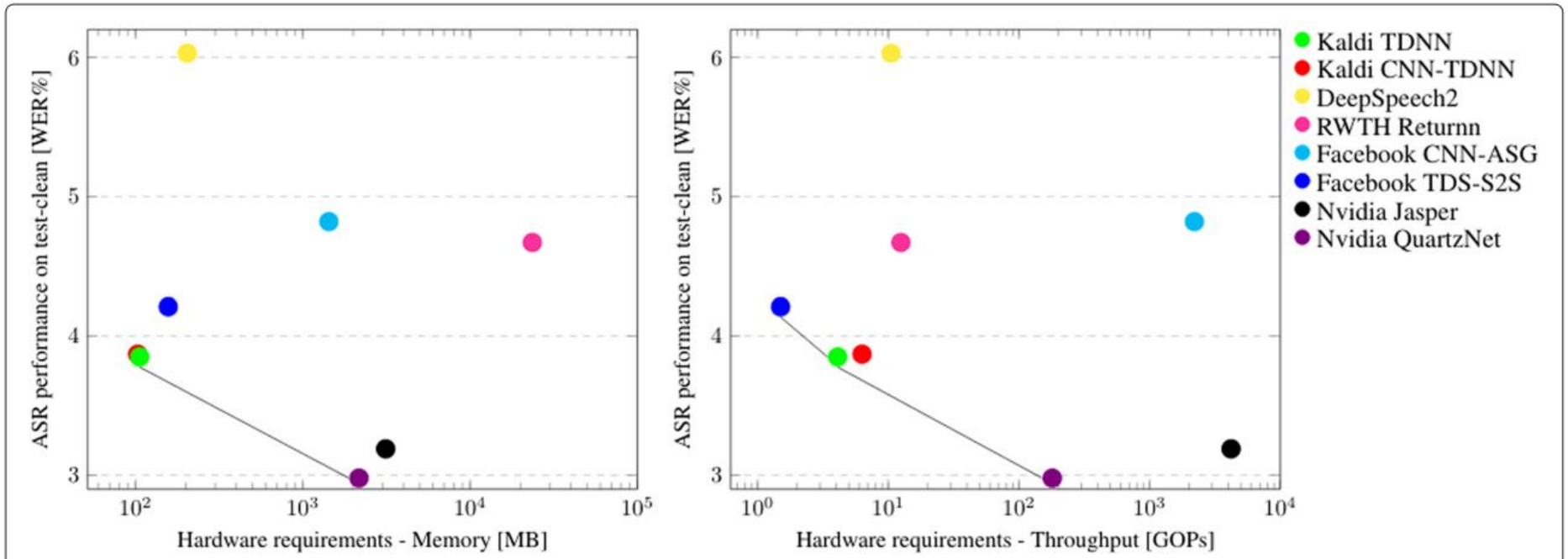


Fig. 7 Trade-off between ASR performance and memory requirements (left) and between ASR performance and throughput requirements (right). Memory load is expressed in Mega bytes (MB). Throughput is expressed in Gigaoperations per second (GOPS)

lower WER is better !

Georgescu et al: Performance vs. hardware requirements in state-of-the-art automatic speech recognition (2021!)

Observable Markov Model and Hidden Markov Model

Let us keep ASR aside for a moment

Observed Markov Model (also Markov chain)

A probabilistic graphical model
It is used to compute the probability of a chain of events

Observed Markov Model (also Markov chain)

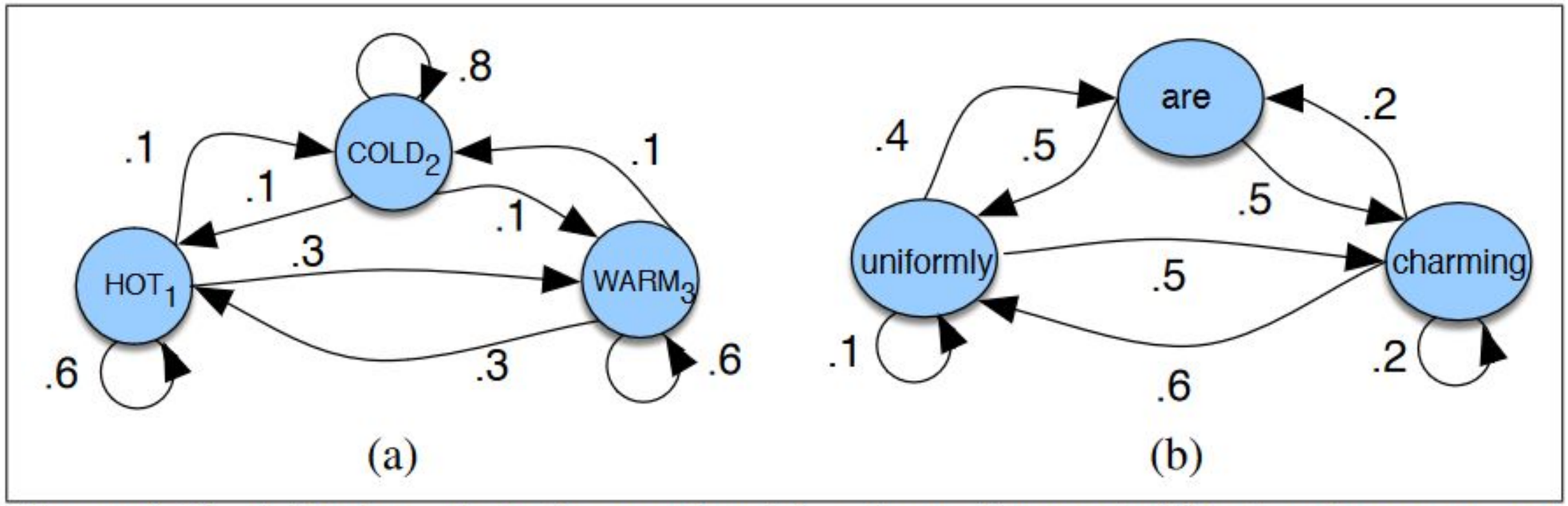


Figure A.1 A Markov chain for weather (a) and one for words (b), showing states and transitions. A start distribution π is required; setting $\pi = [0.1, 0.7, 0.2]$ for (a) would mean a probability 0.7 of starting in state 2 (cold), probability 0.1 of starting in state 1 (hot), etc.

P(COLD,HOT,HOT) ? product of the probabilities of starting in COLD and transitioning from each event to the next

Markov Model:

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{N} \pi_i = 1$

Markov Assumption: $P(q_i = a | q_1 \ldots q_{i-1}) = P(q_i = a | q_{i-1})$

Hidden Markov Model:

Markov chain: probability of an observable sequence of events

What if a sequence of events is not directly observable (*hidden*) and that we want to infer it from another *observed* sequence of events?

Hidden Markov Model:

Markov chain: probability of an observable sequence of events

What if a sequence of events is not directly observable (*hidden*) and that we want to infer it from another *observed* sequence of events ?

Example throughout this section on HMM:

Your friend went on holidays. He does not tell you what the weather was like but he tells you the number of ice-cream he had everyday.
Can you infer the weather?

Hidden Markov Model:

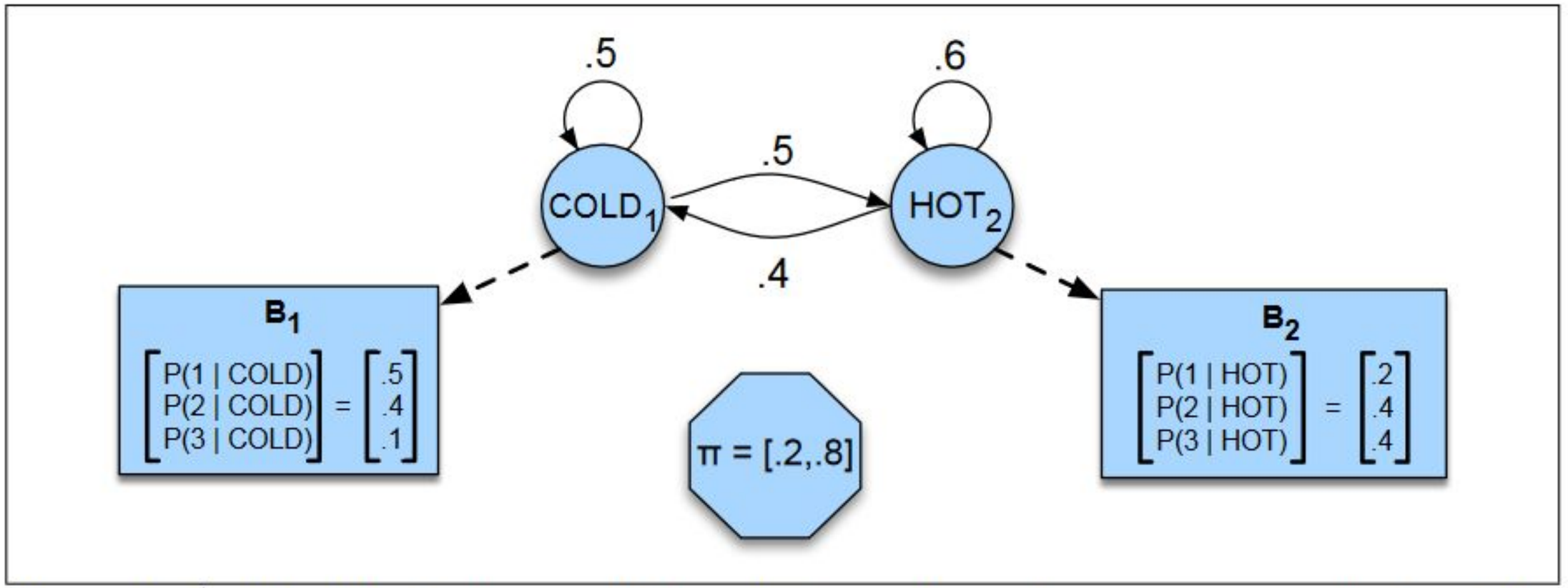


Figure A.2 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

Hidden Markov Model:

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

Markov Assumption: $P(q_i|q_1 \ldots q_{i-1}) = P(q_i|q_{i-1})$

Output Independence: $P(o_i|q_1 \ldots q_i, \ldots, q_T, o_1, \ldots, o_i, \ldots, o_T) = P(o_i|q_i)$

The three problems of HMMs

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

P1: forward algorithm
P2: viterbi algorithm
P3: forward-backward algorithm

Basically read that chapter: https://web.stanford.edu/~jurafsky/slp3/A.pdf

P1 Likelihood and forward algorithm

What is the probability of an observation O (of length T) under an HMM=(A,B) with N states ?

P1 Likelihood and forward algorithm

What is the probability of an observation O (of length T) under an HMM=(A,B) with N states ?

$$P(O) = \sum_{Q} P(O,Q) = \sum_{Q} P(O|Q)P(Q)$$

Using the *output independence assumption and the Markov assumption:*

$$P(O,Q) = P(O|Q) \times P(Q) = \prod_{i=1}^{T} P(o_i|q_i) \times \prod_{i=1}^{T} P(q_i|q_{i-1})$$

B (emission) A(transition)

P1 Likelihood and forward algorithm

Introduce the forward probability:

$$\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$$

Can be written in recursive forms.
Let us create a lattice

Forward recursive formulation:

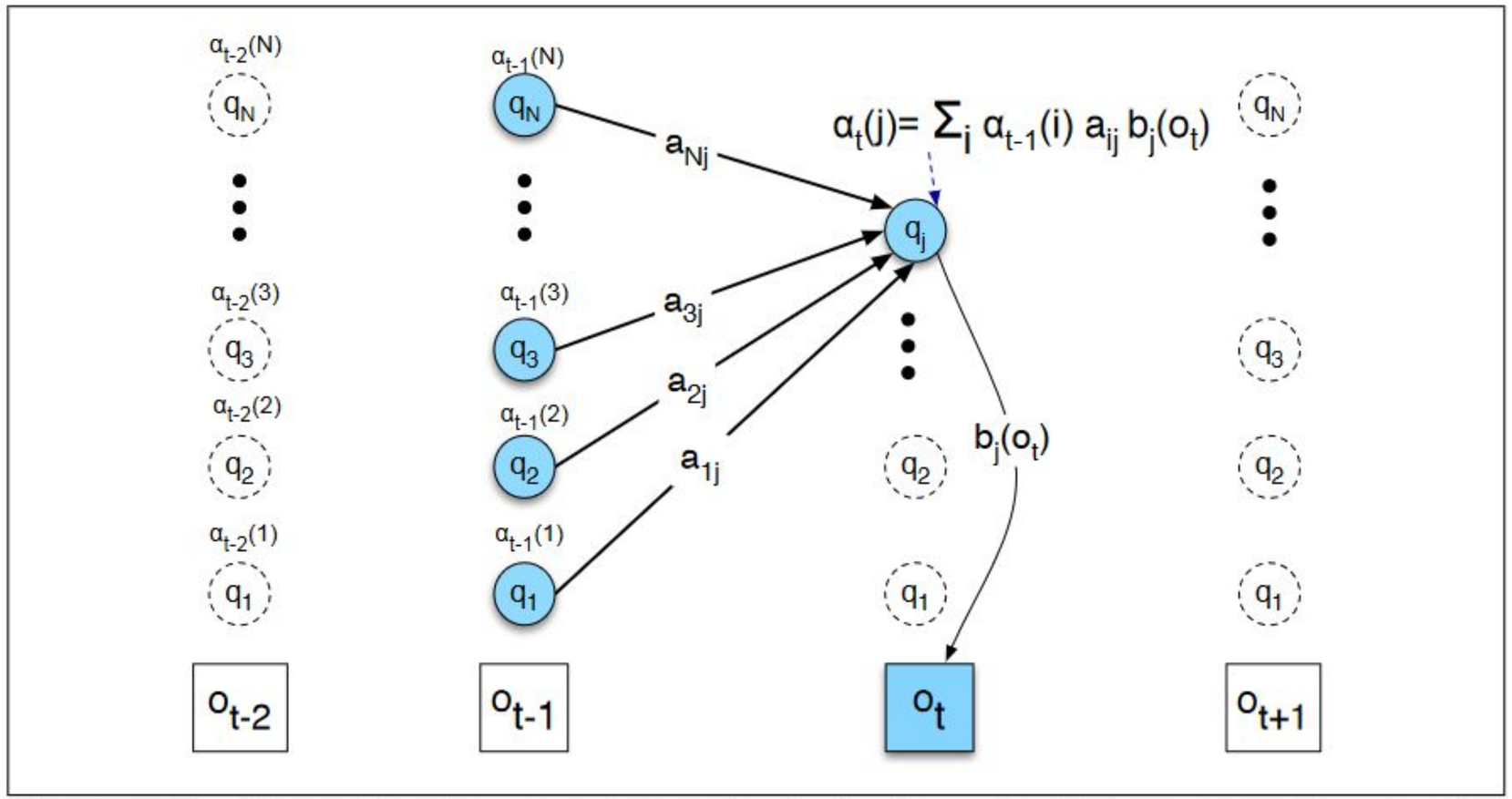


Figure A.6 Visualizing the computation of a single element $\alpha_t(i)$ in the trellis by summing all the previous values α_{t-1}, weighted by their transition probabilities a, and multiplying by the observation probability $b_i(o_t)$. For many applications of HMMs, many of the transition probabilities are 0, so not all previous states will contribute to the forward probability of the current state. Hidden states are in circles, observations in squares. Shaded nodes are included in the probability computation for $\alpha_t(i)$.

The forward algorithm

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \le j \le N, 1 < t \le T$$

Recursive formulation, number of Q sequences: N^T !! Need Dynamic Programming

how do you compute fibo(n) ?

The forward algorithm

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \le j \le N, 1 < t \le T$$

Recursive formulation, number of Q sequences: N^T !! Need Dynamic Programming

Dynamic Programming: store already computed values for future use

Complexity => $O(N^2T)$

The forward algorithm

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \le j \le N, 1 < t \le T$$

Recursive formulation, number of Q sequences: N^T !! Need Dynamic Programming

Dynamic Programming: store already computed values for future use
Complexity => $O(N^2T)$

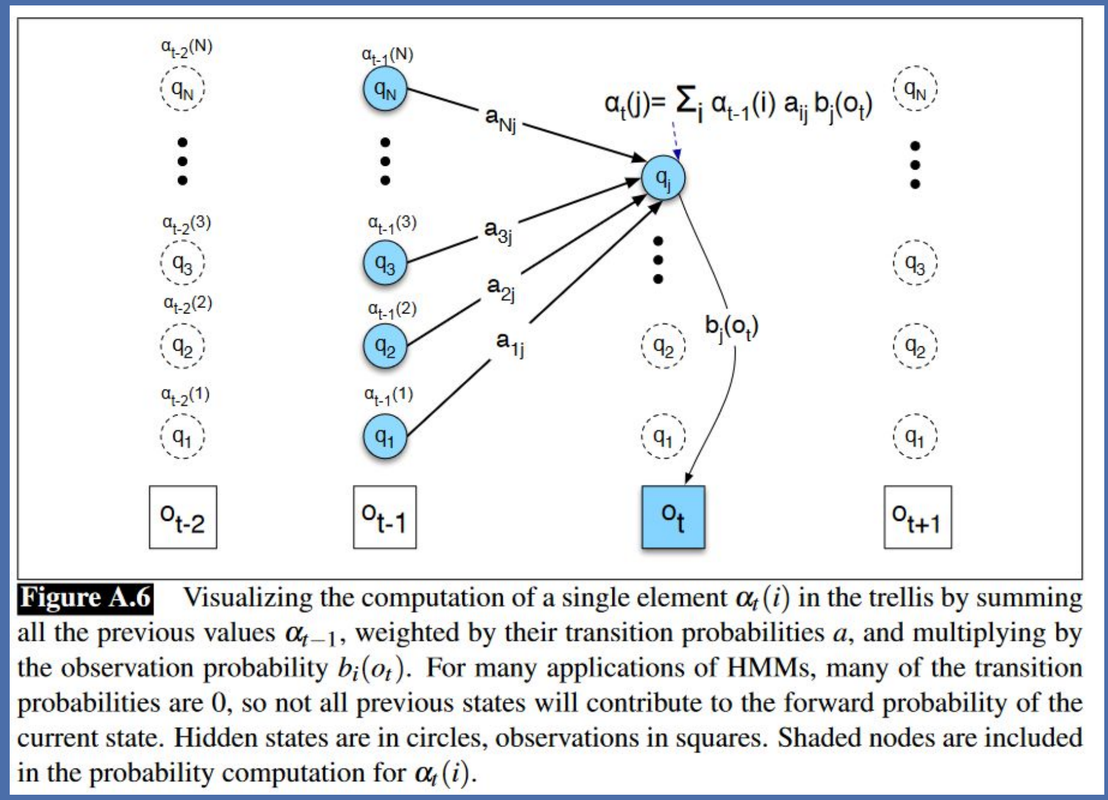


Figure A.6 Visualizing the computation of a single element $\alpha_t(i)$ in the trellis by summing all the previous values α_{t-1}, weighted by their transition probabilities a, and multiplying by the observation probability $b_i(o_t)$. For many applications of HMMs, many of the transition probabilities are 0, so not all previous states will contribute to the forward probability of the current state. Hidden states are in circles, observations in squares. Shaded nodes are included in the probability computation for $\alpha_t(i)$.

The three problems of HMMs, illustrated with the Ice-cream/Weather example

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

P1: forward algorithm
P2: viterbi algorithm
P3: forward-backward algorithm

Basically read that chapter: https://web.stanford.edu/~jurafsky/slp3/A.pdf

P2: Decoding with Viterbi

The lattice is the same, only the cells are are not exactly the same:

Forward:

$$\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$$

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

Viterbi:

$$v_t(j) = \max_{q_1,\cdots,q_{t-1}} P(q_1 \ldots q_{t-1}, o_1, o_2 \ldots o_t, q_t = j|\lambda)$$

Recursive formulation:

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$$

P2: Viterbi, change the sum by a max and keep a backpointer

Recursion

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T$$

$$bt_t(j) = \operatorname*{argmax}_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T$$

Termination:

$$\text{The best score:} \quad P* = \max_{i=1}^{N} v_T(i)$$

$$\text{The start of backtrace:} \quad q_T* = \operatorname*{argmax}_{i=1}^{N} v_T(i)$$

The three problems of HMMs, illustrated with the Ice-cream/Weather example

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

P1: forward algorithm
P2: viterbi algorithm
P3: forward-backward algorithm

Basically read that chapter: https://web.stanford.edu/~jurafsky/slp3/A.pdf

P3: training HMM with forward backward (also Baum-Welch)

Supposed we have a **labeled** database with ice-cream and weather

```
3    3   2        1    1    2        1    2  3
hot  hot cold     cold cold cold     cold hot hot
```

P3: training HMM with forward backward (also Baum-Welch)

Supposed we have a **labeled** database with ice-cream and weather

```
3    3   2        1    1    2        1    2  3
hot  hot cold     cold cold cold    cold hot hot
```

We can estimate our parameters for matrix A:

$$p(hot|hot) = 2/3 \quad p(cold|hot) = 1/3$$
$$p(cold|cold) = 2/3 \quad p(hot|cold) = 1/3$$

And for matrix B:

$$P(1|hot) = 0/4 = 0 \quad p(1|cold) = 3/5 = .6$$
$$P(2|hot) = 1/4 = .25 \quad p(2|cold = 2/5 = .4$$
$$P(3|hot) = 3/4 = .75 \quad p(3|cold) = 0$$

P3: forward backward

What if we do not have a the weather information ?

Expectation Maximisation (EM) an **unsupervised** algorithm to estimate A and B:

P3: forward backward

What if we do not have a the weather information ?

Expectation Maximisation (EM) an **unsupervised** algorithm to estimate A and B:

Start with random model and data (ice-cream sequences)
While not likelihood increases:
Compute the probability of your data (likelihood) under your model
Re-estimate the parameters of your model based on your data’s probability

P3: forward backward

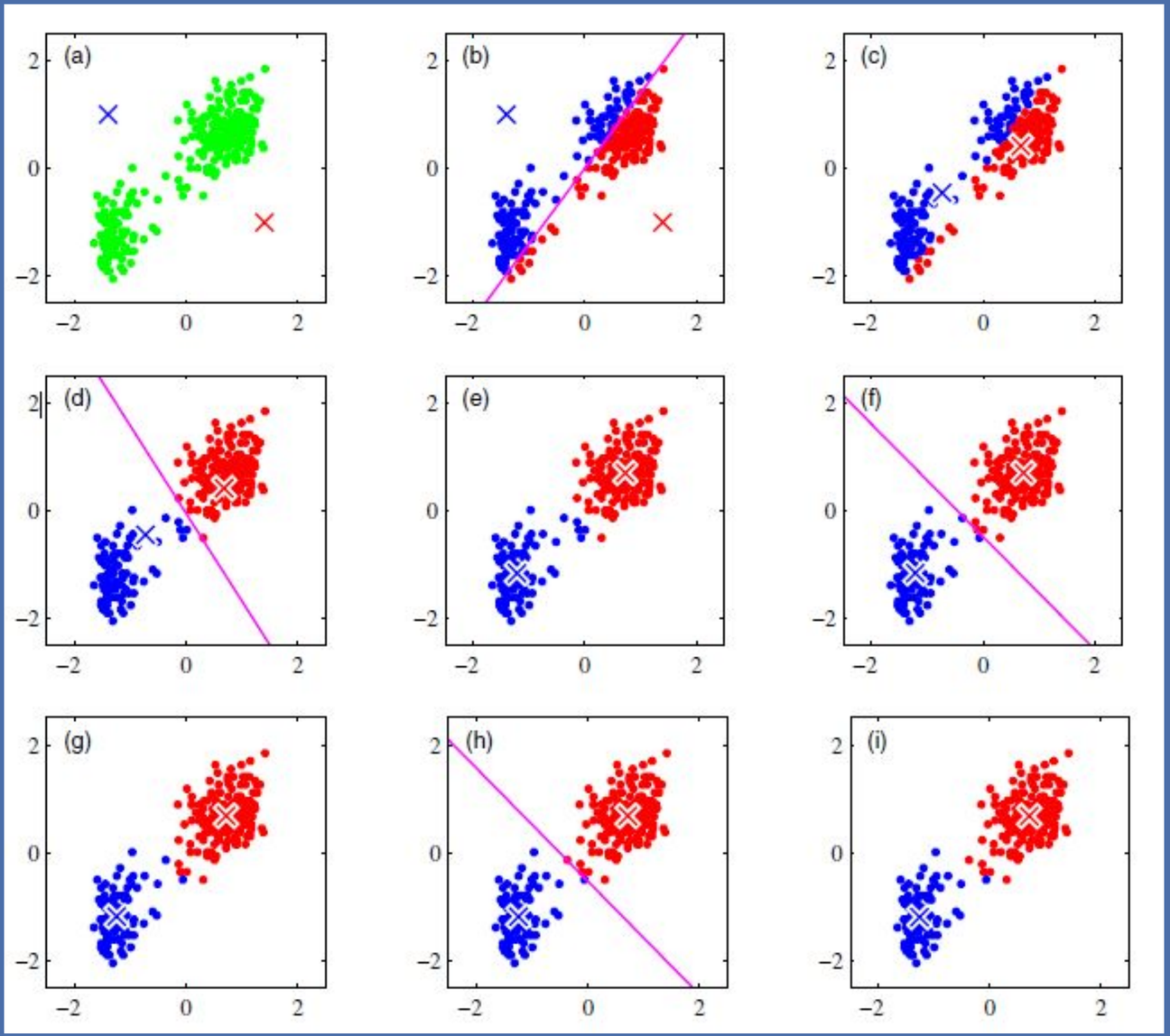
What if we do not have a the weather information ?

Expectation Maximisation (EM) an **unsupervised** algorithm to estimate A and B:

Start with random model and data (ice-cream sequences)
While not likelihood increases:
Compute the probability of your data (likelihood) under your model
Re-estimate the parameters of your model based on your data's probability

EM is based on Maximum Likelihood Estimation
Guarantee to increase the likelihood
No guarantee to converge to the global optimum

Informal understanding of EM with k-means



Bishop p.426

Informal understanding of EM for Gaussian Models

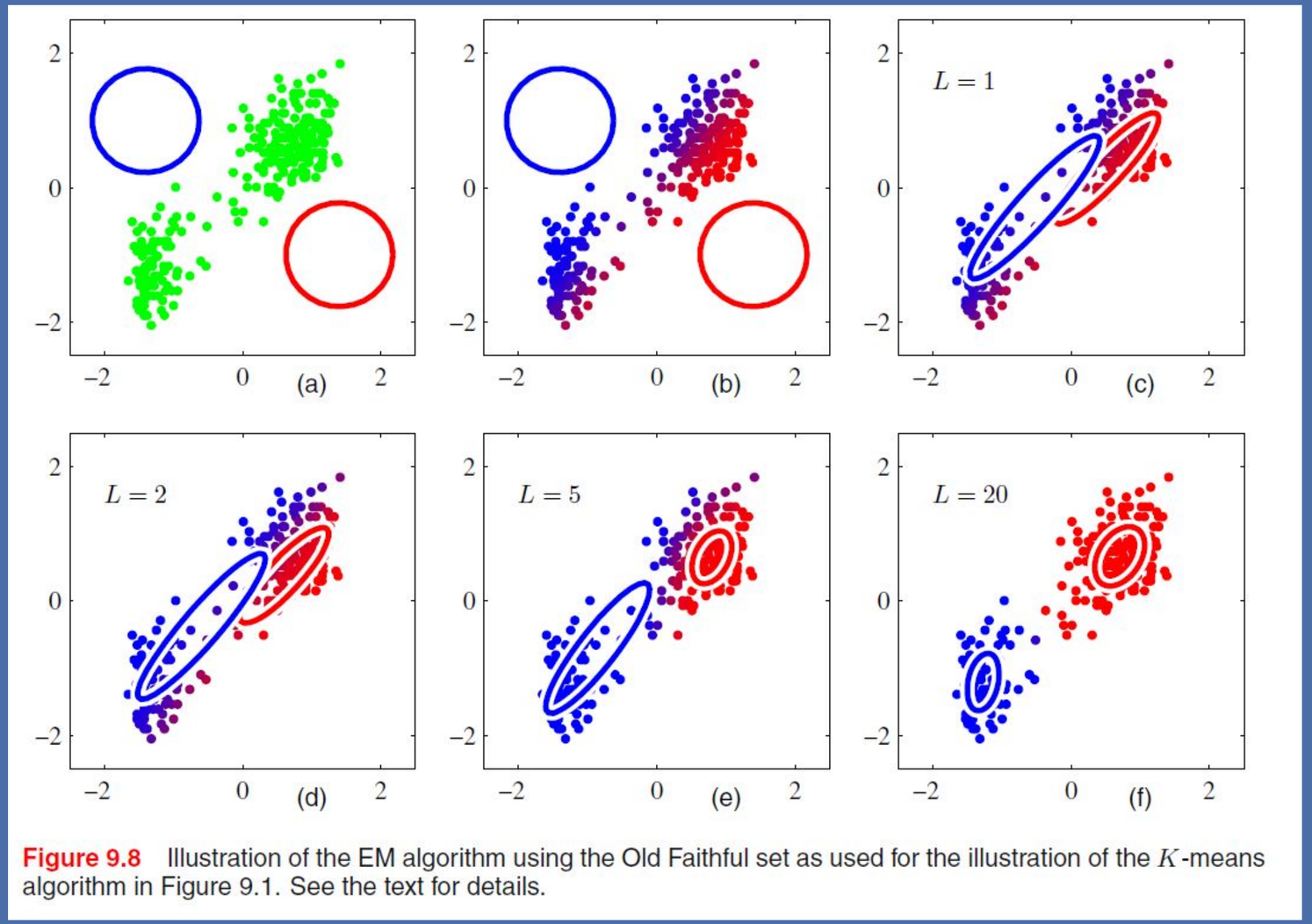


Figure 9.8 Illustration of the EM algorithm using the Old Faithful set as used for the illustration of the K-means algorithm in Figure 9.1. See the text for details.

Bishop p.437

Forward-backward: compute a forward and a backward pass

Forward-backward: compute a forward and a backward pass

$$\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$$

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

$$\beta_t(i) = P(o_{t+1}, o_{t+2} \ldots o_T | q_t = i, \lambda)$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j)$$

Forward-backward: compute a forward and a backward pass

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$

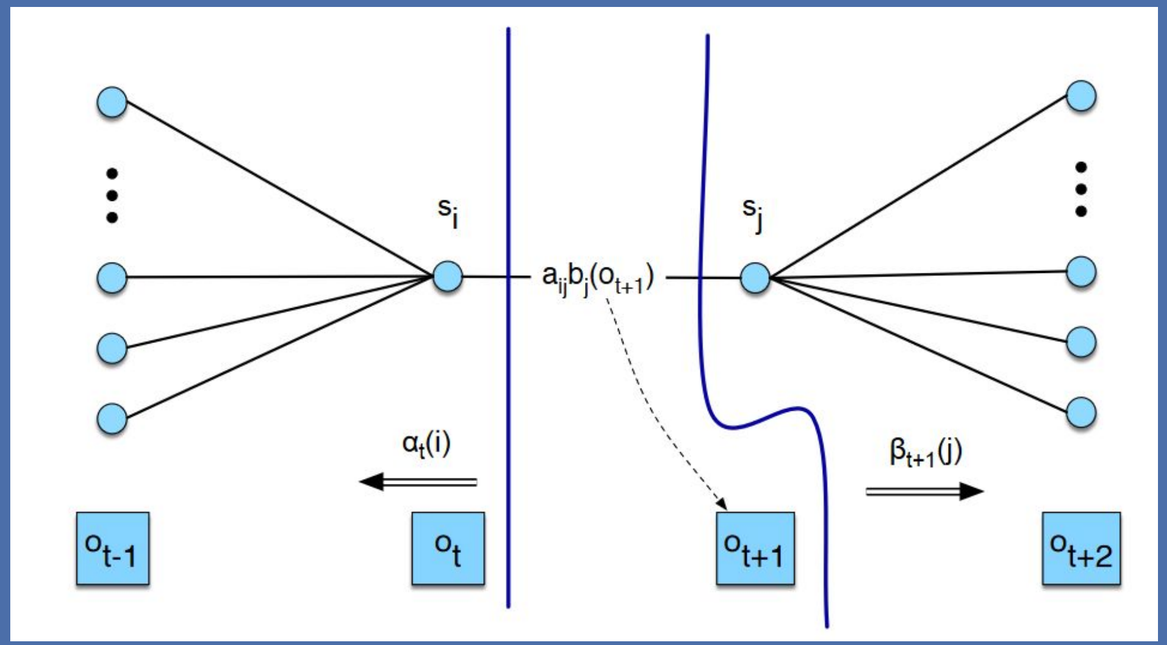
$$\hat{b}_j(v_k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

(each observation o_t is drawn for a vocabulary V)

Forward-backward:

expected number transition from state i to j ?

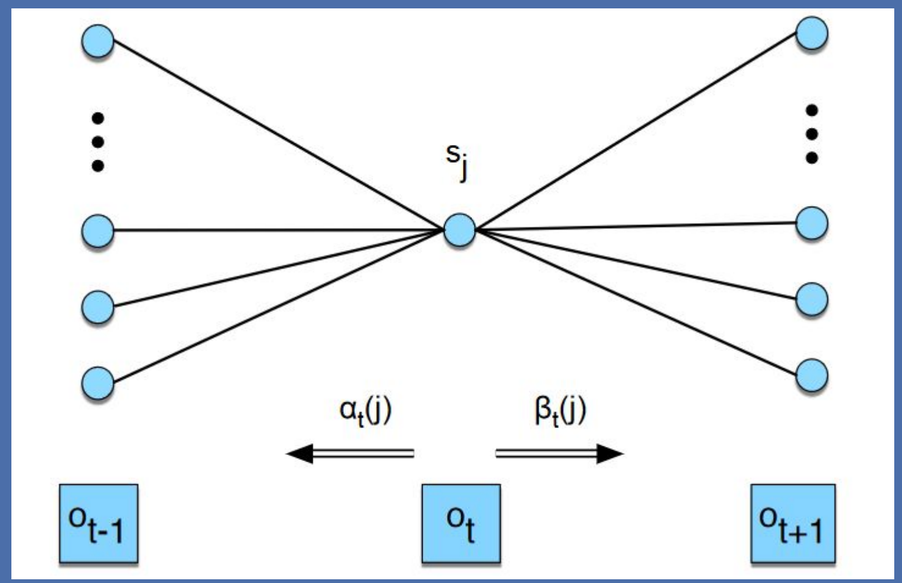
$$P(q_t = i, q_{t+1} = j | O, \lambda)$$



$$\alpha_t(i)\, a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

expected number of times in state j and observing the symbol v_k ?

$$P(q_t = j | O, \lambda)$$



$$\alpha_t(j)\beta_t(j)$$

Forward backward:

E step:

$$\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$$

$$\beta_t(i) = P(o_{t+1}, o_{t+2} \ldots o_T|q_t = i, \lambda)$$

$$P(q_t = i, q_{t+1} = j|O, \lambda)$$

$$P(q_t = j|O, \lambda)$$

M step:

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$

$$\hat{b}_j(v_k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

HMM for ASR

Let’s get back to ASR now

Using HMM for ASR

Why unsupervised learning (forward-backward) for ASR?

Semi-supervised case: we have the transcription of the speech data
but we do not have the state-level alignment of the recordings

Using HMM for ASR

Why unsupervised learning (forward-backward) for ASR?

Semi-supervised case: we have the transcription of speech data
but we do not have the state-level alignment of the recordings

What are the states in for ASR ?

Using HMM for ASR

Why unsupervised learning (forward-backward) for ASR?

Semi-supervised case: we have the transcription of speech data
but we do not have the state-level alignment of the recordings

What are the states in for ASR ?

one state per words ? too coarse
need to be defined for each words (how to define them? next class)

What are the emission probabilities ?

Using HMM for ASR

Why unsupervised learning (forward-backward) for ASR?

Semi-supervised case: we have the transcription of speech data
but we do not have the state-level alignment of the recordings

What are the states in for ASR ?

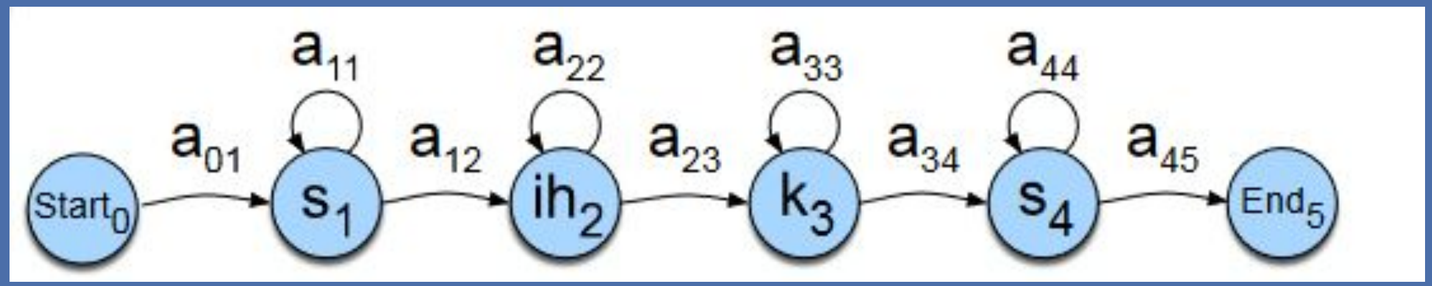
one state per words ? too coarse
need to be defined for each words (how to define them? next class)

What are the emission probabilities ?

Gaussian mixture model (GMM)

Using HMM for ASR

Let us consider the case of single-word HMM for the word ‘six’
Database: speech recordings of the word ‘six’
Goal: does a speech utterance says ‘six’ ?



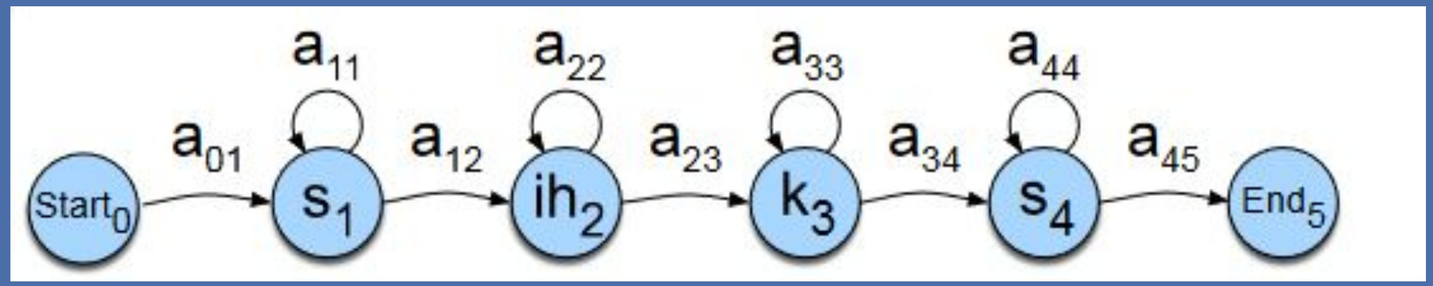
Single-word HMM-GMM

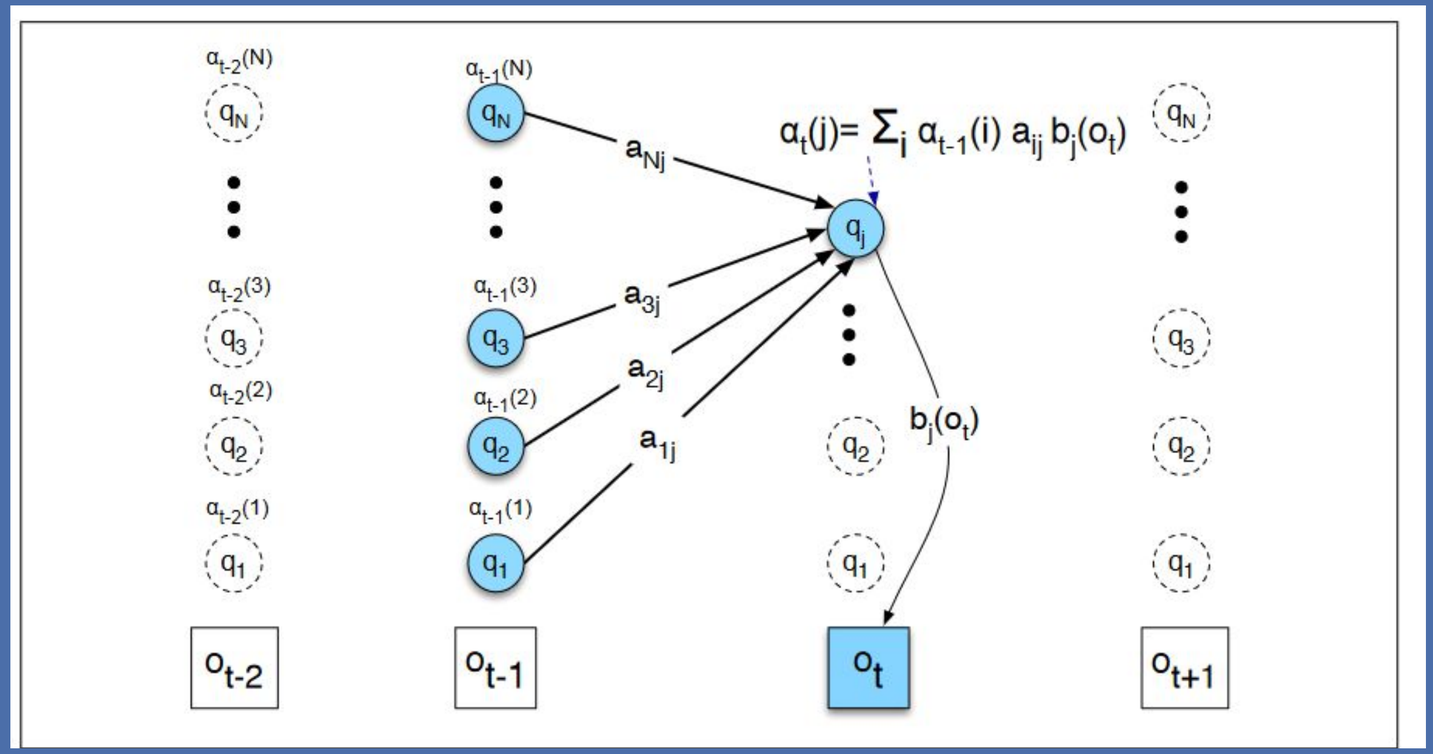
States are phonemes (very bad in practice)
Each state is modelled with a GMM

$t \in [1,T]$, o_t is a speech frame (39 MFCC)

$j \in [1,N]$, q_j is a phoneme (s,ih,k,s)

$j \in [1,N]$, b_j is a GMM for MFCC vectors



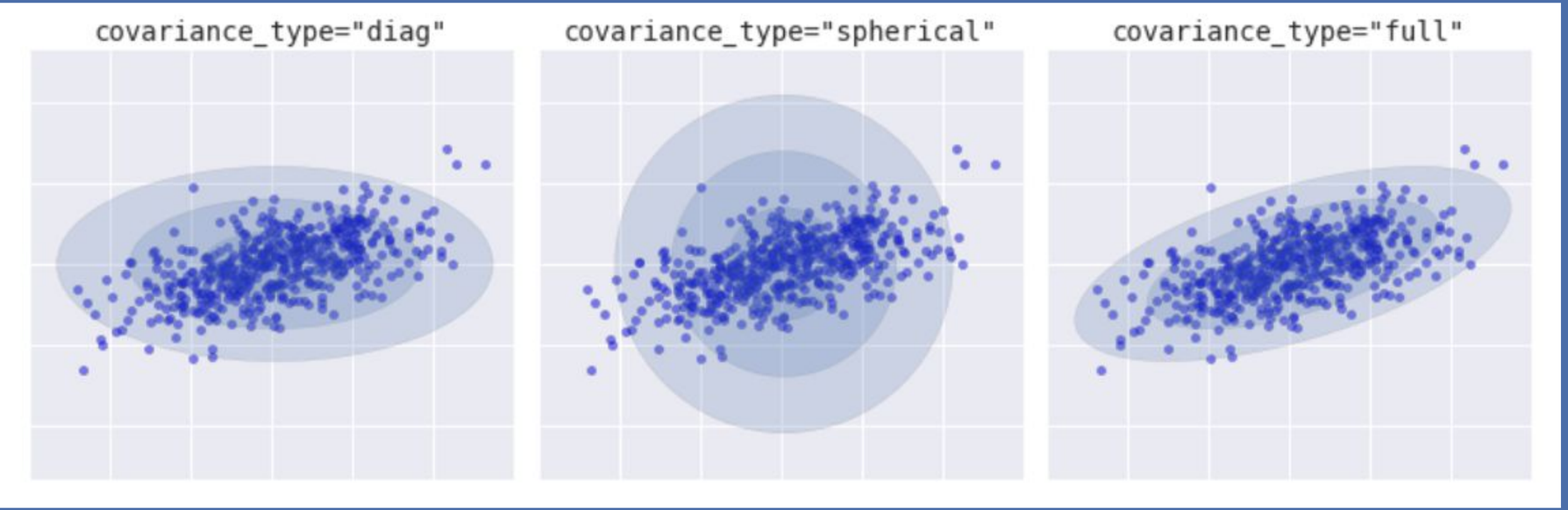


Multivariate Gaussian Model

Each state is modelled with a multivariate Gaussian that takes in MFCC frames (39 dimension)

$$b_j(o_t) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(o_t - \mu_j)^T \Sigma_j^{-1}(o_t - \mu_j)\right)$$

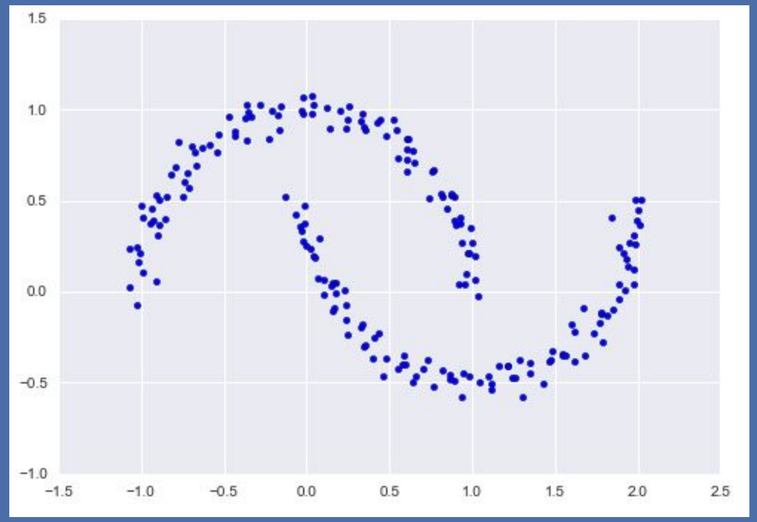
Reminder: MFCCs are decorrelated: use a diagonal covariance



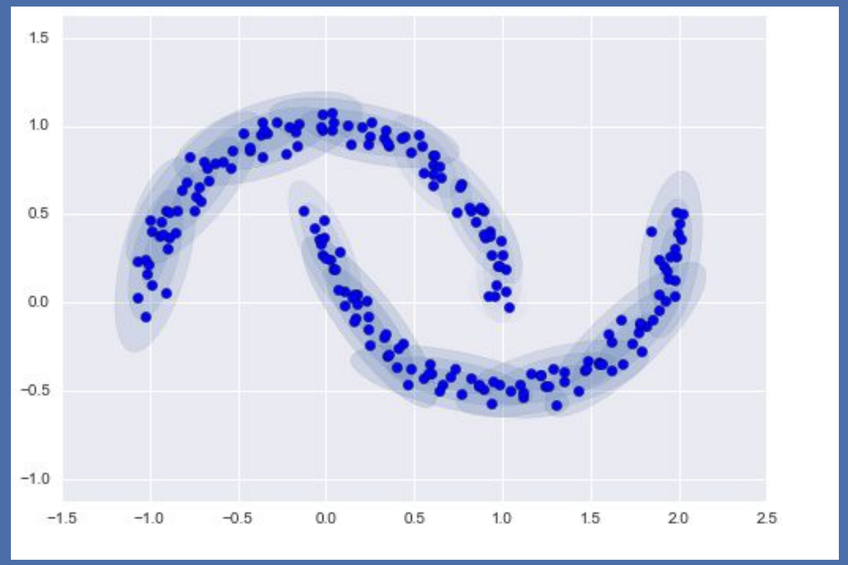
GMM: Mixture of Gaussians

You may use several multivariate gaussians with c_k that sums to 1

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} \frac{1}{\sqrt{2\pi|\Sigma_{jm}|}} exp[(x - \mu_{jm})^T \Sigma_{jm}^{-1} (o_t - \mu_{jm})]$$



mixture of 16 multivariate gaussians



Single-word HMM recap

Given an database of several recordings of the word ‘six’
Let us train a HMM-GMM for the word ‘six’ with N=4 states

1) Encode your database into MFCC frames

2) Train A and B with forward-backward

At inference:

1) decode a new utterance: encode in MFCC and use Viterbi
2) You get the probability of decoding the word six
(You need to set a threshold for accepting or rejecting the decoding)

What if we have a large vocabulary ?

One HMM model per word to decode!
Can we do better than forward-backward?

Conditional MLE (or Maximum Mutual Information Estimation, MMIE)

MLE maximize the probability of observing a sequence knowing we are decoding a word (M_k)

$$\mathcal{F}_{\text{MLE}}(\lambda) = P_\lambda(O|M_k)$$

We want to probability of the correct word (M_k) to be higher than other word

$$\mathcal{F}_{\text{CMLE}}(\lambda) = P_\lambda(M_k|O) = \frac{P_\lambda(O|M_k)P(M_k)}{\sum_{M \in \mathcal{L}} P_\lambda(O|M)P(M)}$$

During forward-backward:

run multiple forward-backward pass with the HMMs of other words
and subtract the counts during the estimation step

Next class

What about several possible pronunciation of a word?

How to choose the number of mixture in GMM ?

How to decode with a large vocabulary ?

Out of vocabulary words ?

How to use a language model to improve ASR ?

How to use neural models with HMM ?

Can we get rid of HMMs ?

Handling variability in speech

- Word Error Rate
- Noisy environment
- Gender
- Speaker
- Accents

Word Error Rate (WER)

Reference metric in ASR

Smallest amount of Insertion,Deletion,Substitution applied on the prediction to match the true transcription

Also called Levenshtein Distance, Edit Distance

$$WER = \frac{S + D + I}{N}$$

N = number of words in the référence
WER is in percentage
WER can be > 100%

Example: WER = ¾ = 75%

Reference:	SHOW ME THE INTERFACE
Hypothesis:	I SHOW ME FACE
Alignment:	I D S

Word Error Rate (WER)

Recursive formulation

$$\mathrm{lev}(a,b) = \begin{cases} \max(|a|,|b|) & \text{si } \min(|a|,|b|) = 0, \\ \mathrm{lev}(a-1,b-1) & \text{si } a[0] = b[0], \\ 1 + \min \begin{cases} \mathrm{lev}(a-1,b) \\ \mathrm{lev}(a,b-1) \\ \mathrm{lev}(a-1,b-1) \end{cases} & \text{sinon.} \end{cases}$$

Solved by dynamic programming in O(|a|*|b|)

Domain shift in ASR

Noisy environment, different accentuations, biological differences

Two ways to address them all:

- bigger database: include different recording conditions, genders, ethnicities
- data augmentation: reverb, add gaussian noise, time stretch, pitch shift,...

Can we do better than that?

Difficulty of noisy environments

- Real life VS studio recordings: huge drop in WER
- No real method to address this
  (appart bigger database + data aug)

Dataset	wav2vec 2.0 Large (no LM)	Whisper Large V2	RER (%)
LibriSpeech Clean	**2.7**	**2.7**	0.0
Artie	24.5	**6.2**	74.7
Common Voice	29.9	**9.0**	69.9
Fleurs En	14.6	**4.4**	69.9
Tedlium	10.5	**4.0**	61.9
CHiME6	65.8	**25.5**	61.2
VoxPopuli En	17.9	**7.3**	59.2
CORAAL	35.6	**16.2**	54.5
AMI IHM	37.0	**16.9**	54.3
Switchboard	28.3	**13.8**	51.2
CallHome	34.8	**17.6**	49.4
WSJ	7.7	**3.9**	49.4
AMI SDM1	67.6	**36.4**	46.2
LibriSpeech Other	6.2	**5.2**	16.1
Average	29.3	**12.8**	55.2

from Robust Speech Recognition via Large-Scale Weak Supervision

Speaker variability

- Variability in speakers is one of the main challenges in speech recognition:
  - different vocal tract and larynx
  - some accents are not well recognize

HMM-GMM can be easily tuned to a specific speaker (not the case of neural models, overfits too quickly)

Model	med. avp	P_{90} avp	top accent	worst acccent
SB transformer	22.2	30.6	19.6	28.7
SB crdnn	28.4	40.0	24.5	39.7
W2v base 960-960	23.3	32.5	19.6	30.7
W2v large 960-960	18.2	25.9	14.7	24.0
W2v large 60k-100	18.9	28.6	15.6	26.8
W2v large 60k-960	20.4	28.3	17.9	27.7
RASR small	18.6	29.6	15.7	25.6
RASR small-distill	20.2	28.9	16.9	26.0
RASR big	**15.7**	23.5	**12.3**	21.9
Google API video	16.0	**23.4**	12.9	**19.7**

Table 3. Median WER on accented rehearsed speech On accented Vox Populi (AVP), the global median WER (med.) is compared to the best median WER by accent and the worst one.

from ASR4REAL: AN EXTENDED BENCHMARK FOR SPEECH MODELS

Fitting the GMMs to a new speaker:

Given a small speaker set (<1 min of speech)

MLLR (maximum likelihood linear regression):

Learn a affine transformation of the mean of the gaussian mixture to maximize the likelihood on the speaker set (regression loss)

fMLLR (feature space maximum likelihood linear regression):
same thing but on the MFCCs features

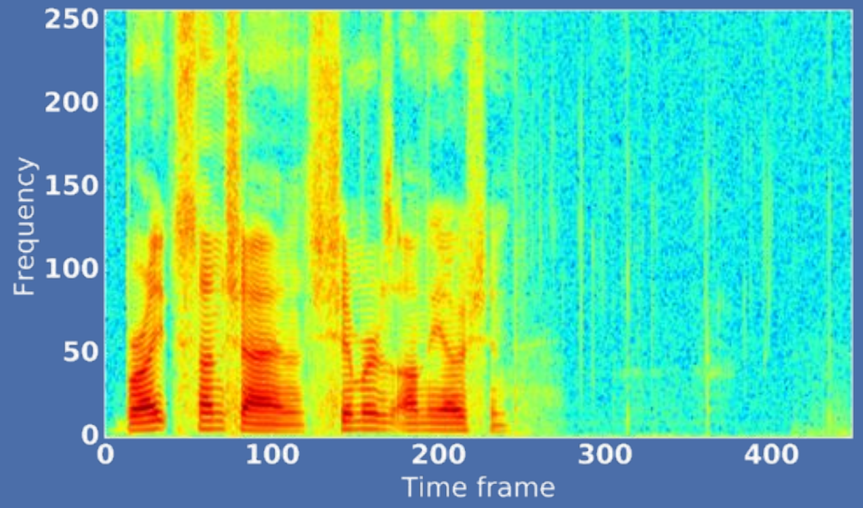
Advantage: big gain in WER
Disadvantage: may overfit the speaker set

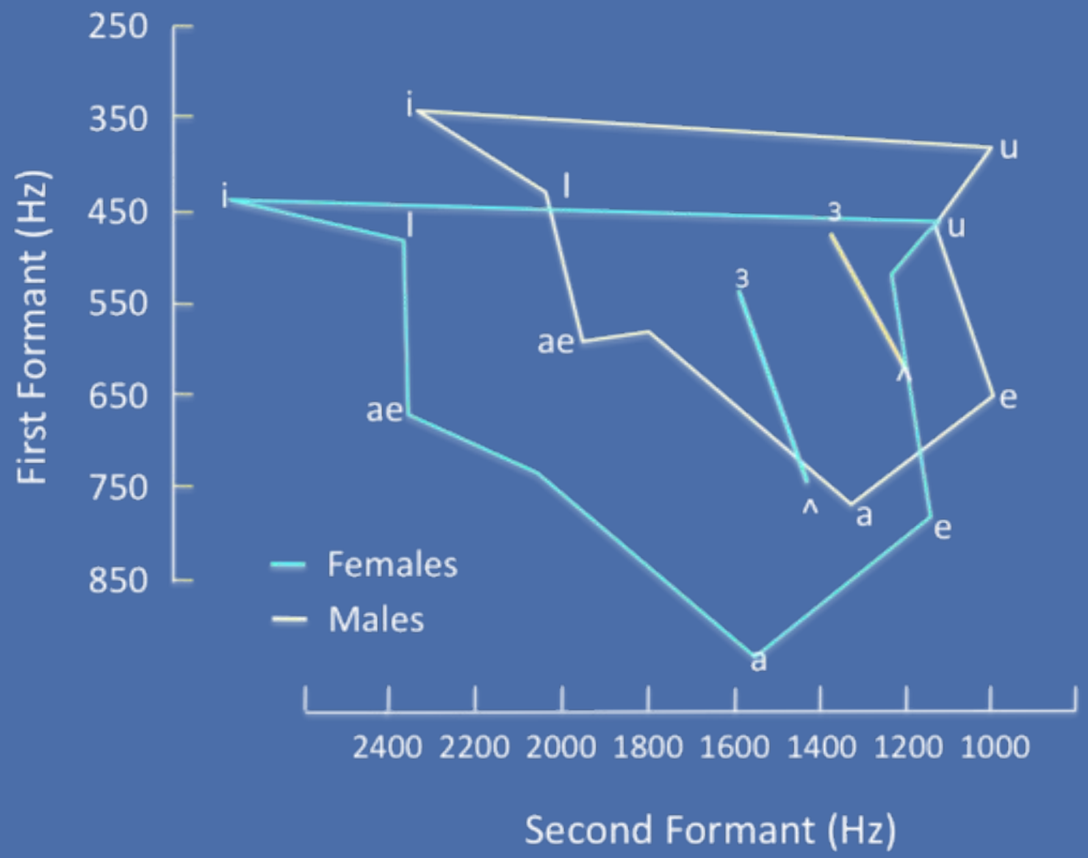
Dealing with variations in gender

- Female voices have a higher pitch than male voices, and children’s voices have a higher pitch than female voices
- There are also deformations in the formants (spectrogram patterns that characterize phonemes)

Dealing with variations in gender

- Formants: main resonances that characterize the phonetic content

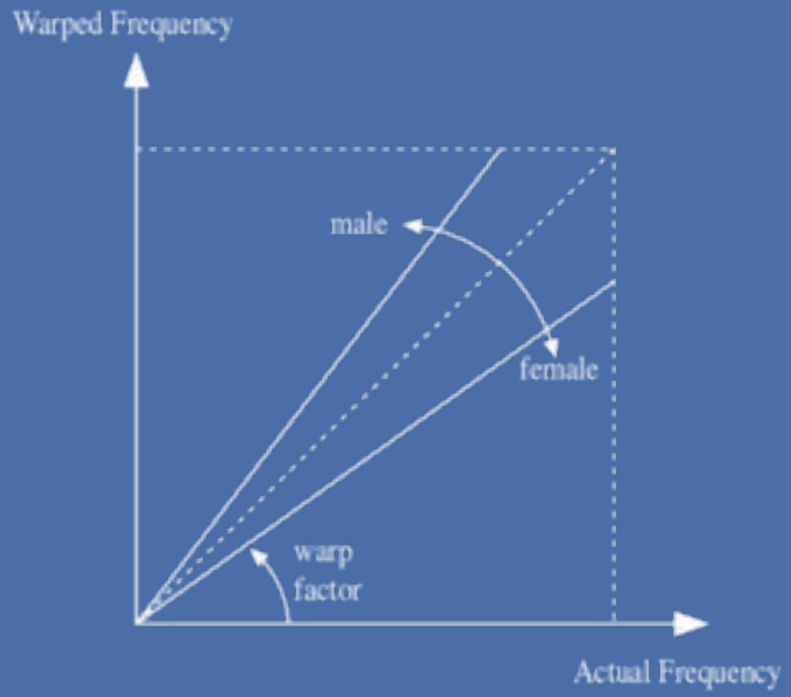




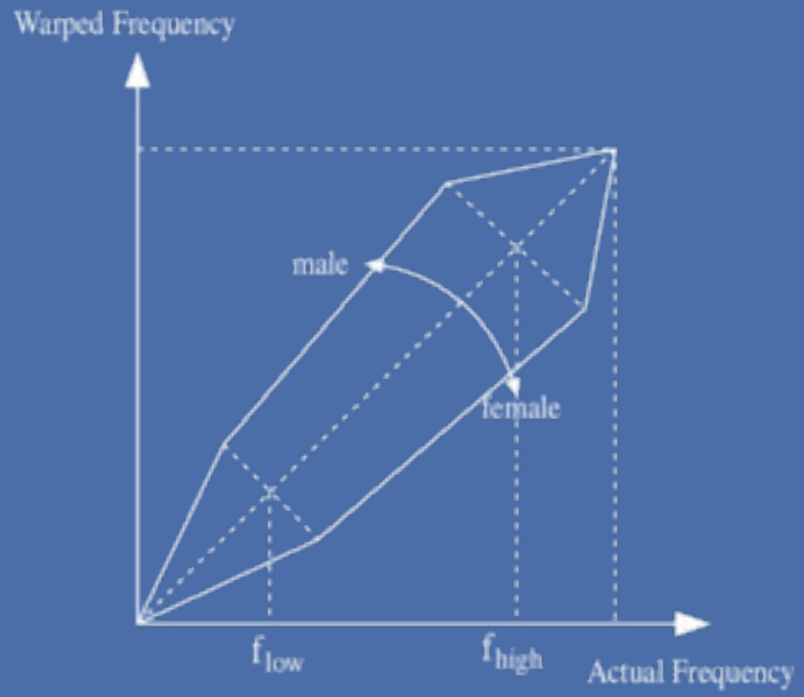
Dealing with variations in gender: Vocal Tract Length Normalization

Shift the spectrum up or down based on the gender

Fit the scaling factor after training to maximize the likelihood (gain a few WER points)



Linear scaling



Piecewise-linear scaling

Dealing with variations in gender

- The main way of dealing with this variability is to have a balanced dataset: Half male, half female
- This way your classifier will not be biased towards males or female voices
- Same solution for the system to work on children’s voices

Conclusion

- Speech and speech features
- Modeling speech with Hidden Markov Models
- Modeling speech features with Gaussian Mixture Models
- Handling variability in gender and speaker

Question and quizz