

VOICE ANALYTICS

Speech processing *crash course*

15/01/2019

Kris Demuynck

What & Why

speech is the most used form of communication between humans
speech is considered a cornerstone of human intelligence

⇒ learn machines to handle speech

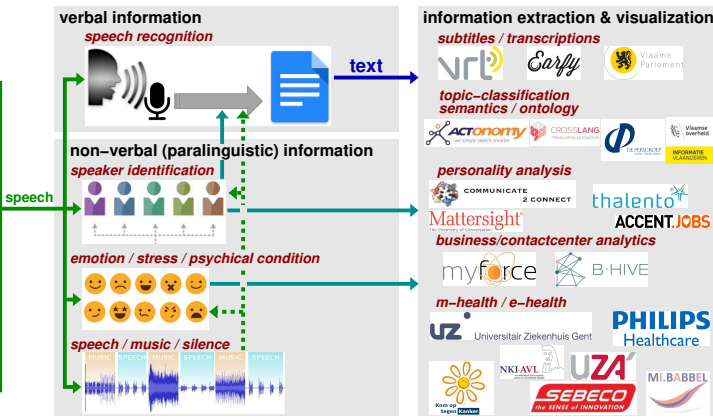
telephone



face-to-face



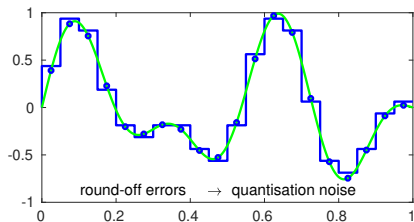
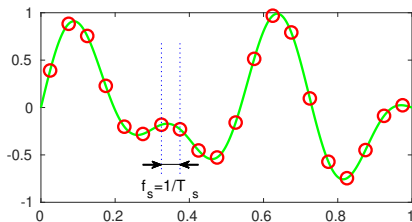
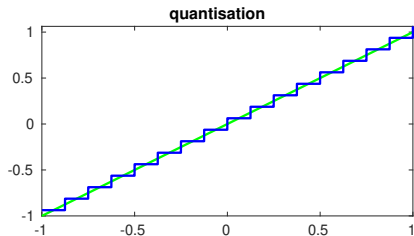
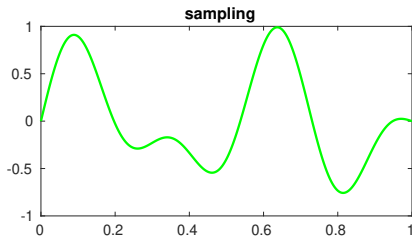
meetings



Speech as acoustic signal

Recording speech:

■ digital (analog is obsolete) \Rightarrow sampling + quantisation



Speech as acoustic signal

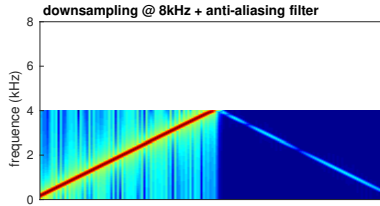
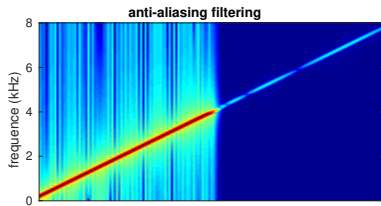
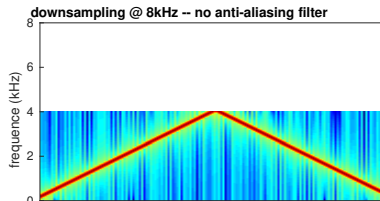
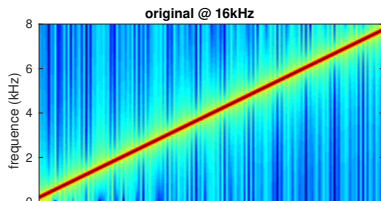
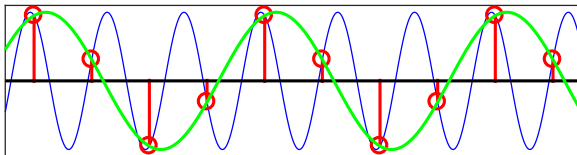
Recording speech:

- digital (analog is obsolete) \Rightarrow sampling + quantisation
- typical sample frequencies
 - ▶ telephone: 8kHz (information loss)
 - ▶ **speech processing**: 16kHz (close to no information loss)
 - ▶ CD-quality: 44.1kHz (sufficient to match the human hearing)
 - ▶ DAT (digital audio tape): 48kHz
 - ▶ high-end audio gear: ~~96kHz or 192kHz~~ (for analysing bat sounds?)
- Nyquist: max. frequency content = $\frac{1}{2}$ of the sample frequency $[1, -1, 1, \dots]$
- (re)sampling: low-pass filter at $f_s/2$ needed to prevent aliasing
- quantisation:
 - ▶ normal: 16bits (quantisation error: -96dB; sufficient for human hearing)
 - ▶ telephone speech: 8bits non-linear
 - ▶ high-end gear: 24 bits
- avoid 'clipping'!



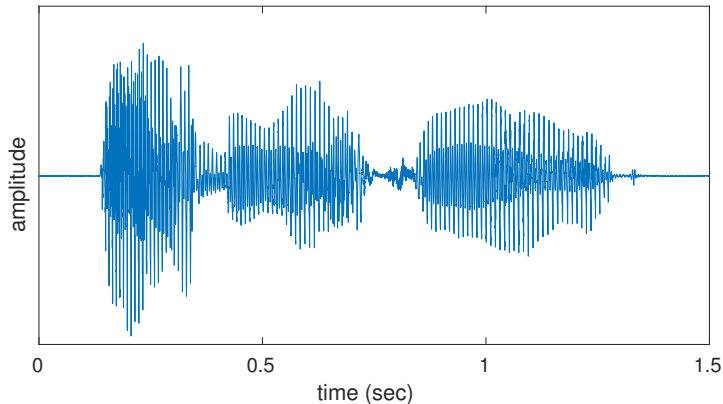
Speech as acoustic signal

Aliasing:



Speech as acoustic signal

Time-amplitude plot (oscillogram)

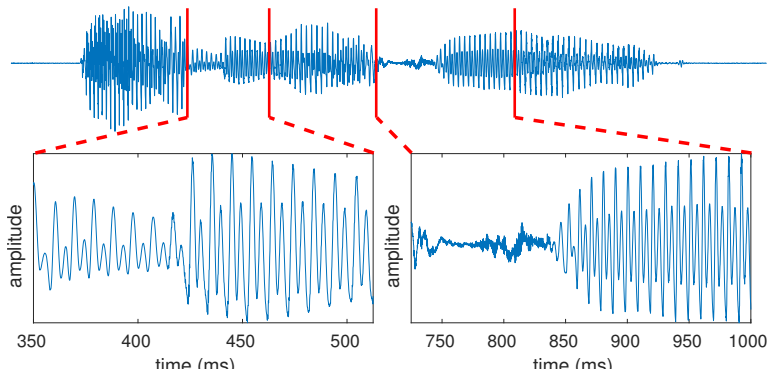


⇒ speech is a non-stationary signal

- a time-amplitude plot is not very informative; we can observe:
 - ▶ energy changes
 - ▶ changes in basic properties of the sounds (pulse train versus noisy)

Speech as acoustic signal

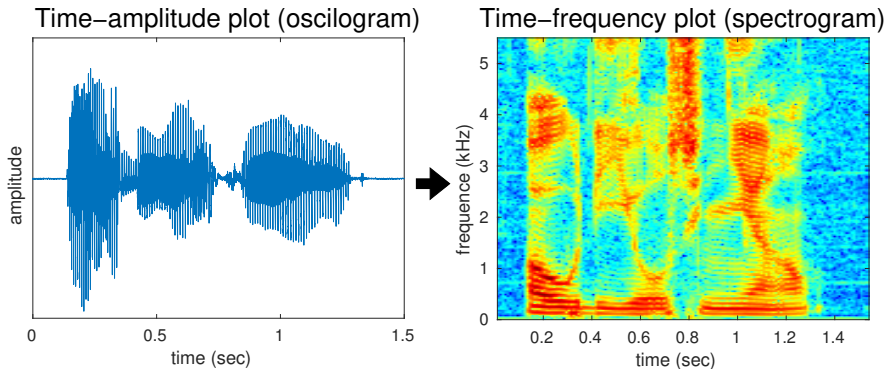
Zoom-in on the waveform:



⇒ more or less isolated acoustic units are detectable

- ▶ either an (almost) periodic signal;
 $F_o = 1/T_o$ (pitch, vibrations per second produced by the vocal cords)
- ▶ either an irregular (noisy) signal

Speech analysis: the spectrogram



spectrogram : same information, different representation

axes : x time
 y frequency
 color (log) energy

usage : render information (much) more visible

Speech analysis: the spectrogram

compact visual representation of spectral information in non-stationary signals

- spectral content \Rightarrow Fourier-transform (see next slide)
- only interested in power distribution (the ear is phase deaf)
- short-term power spectrum: power density along frequency axis

$$S(f; t) = \frac{1}{E_w} \left| \int_{-T/2}^{+T/2} w(u) x(t + u) e^{-j2\pi fu} du \right|^2$$

- Parseval:

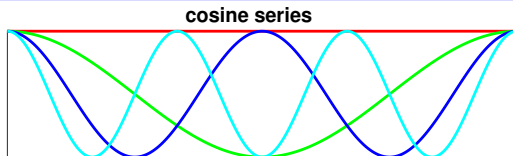
$$V(t) = \int_{-\infty}^{\infty} S(f; t) df$$

- use color to represent the 3D info in 2D
 - ▶ time = x-axis
 - ▶ frequency = y-axis
 - ▶ $S(f; t)$ in dB = gray value, color, ...
- only positive frequencies are drawn (symmetry)

Speech analysis: intermezzo – Fourier-transform

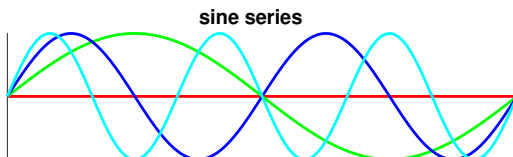
Fourier-transform

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft} dt$$



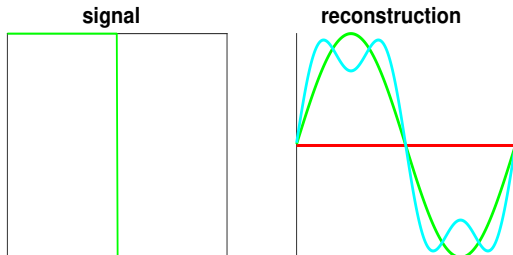
Fourier-series

$$a_n = \int_{-T/2}^{+T/2} x(t) e^{-jn2\pi t/T} dt$$



Z-transform

$$X(e^{j2\pi f}) = \sum_{k=-\infty}^{+\infty} x[k] e^{-j2\pi fk}$$



Discrete Fourier-transform

$$X[n] = \sum_{k=0}^{N-1} x[k] e^{-j2\pi kn/N}$$

Speech analysis: the spectrogram

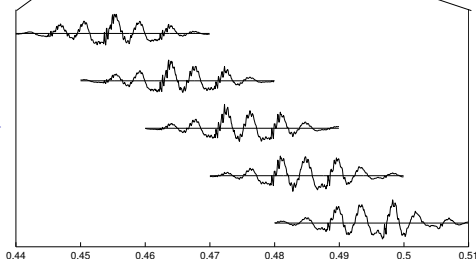
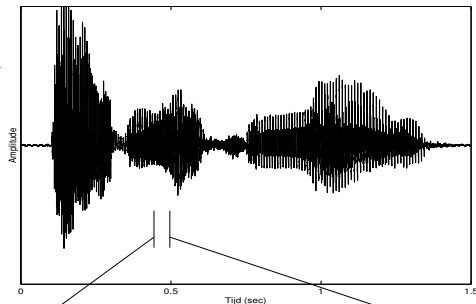
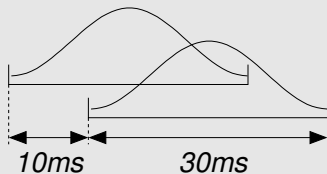
audio signal



pre-emphasis
 $y_i = x_i - 0.95x_{i-1}$

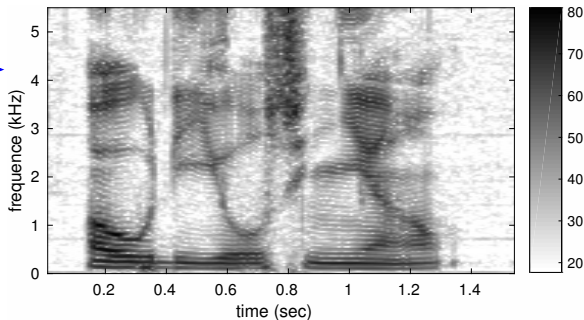


*divide into fragments
+ windowing*



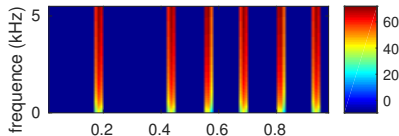
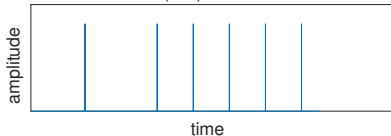
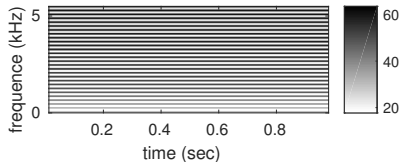
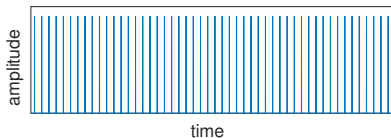
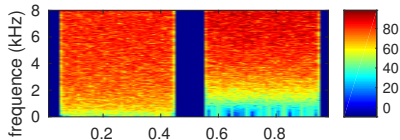
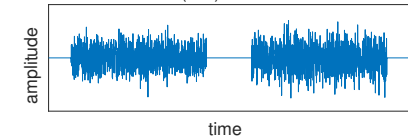
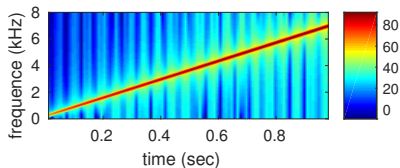
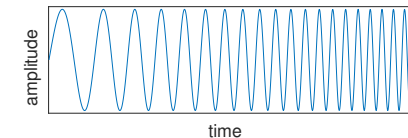
Speech analysis: the spectrogram

Fourier-transformation
+ power in (in dB)
 $10 \log_{10}(\|FFT\|^2)$



- computers \Rightarrow discrete signals \Rightarrow Discrete Fourier Transform
- in practice: Fast Fourier Transform (FFT)
- windowing: DFT/FFT assumes a periodic extension of the signal
 \Rightarrow suppress unpredictable transitions at the edges
- examples on artificial signals – see next slide + [demo1](#)
- visual result: strong dependency on the integration time T in eqn. of $S(f; t)$
speech: small-band versus wide-band spectrogram – [demo2](#)

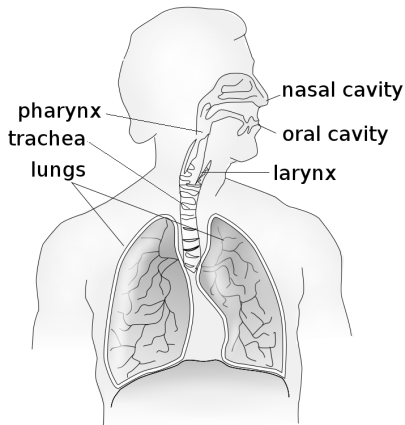
Speech analysis: the spectrogram



Speech production: physiology

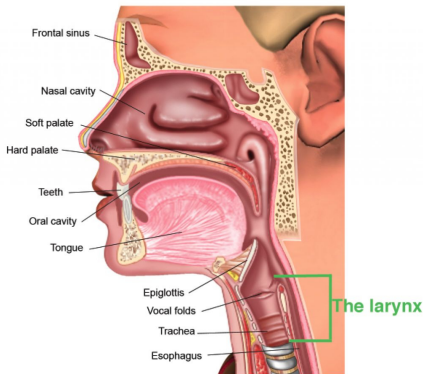
Electro-mechanical equivalent:

- electronics: the brains
- mechanics: lungs, trachea, larynx, pharynx, oral cavity, nasal cavity



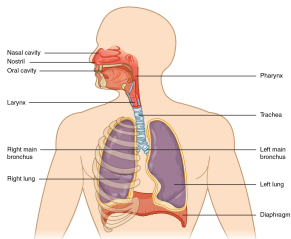
Fast moving parts:

- articulators: lips, tongue, mandible, soft palate (velum), epiglottis
- vocal cords (no articulator since not directly controllable)



Speech production: physiology

Lungs and thorax



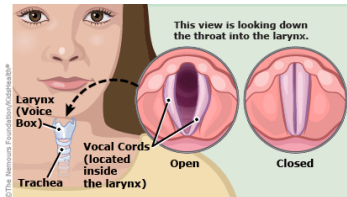
- lungs drive the speech

- ▶ pump air through the larynx
- ▶ air stream makes vocal cords vibrate

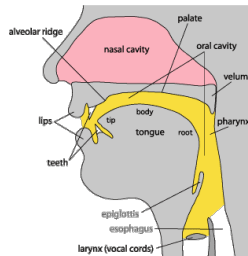
- speech generated only during exhalation

Larynx + vocal cords

- vibration → voiced sounds
- pitch (F_o): vibration freq.
 - ▶ pressure difference
 - ▶ vocal cord length
 - ▶ vocal cord tension
- unvoiced:
stricture in vocal tract
→ turbulent air stream
(white noise)



Vocal tract



- acts as a filter

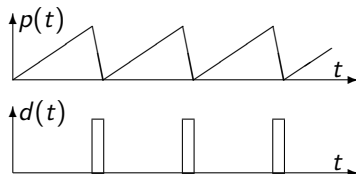


- determines the timbre/color of the sound
- formants (F_1 , F_2): resonance freqs.

Speech production: introducing audible frequencies

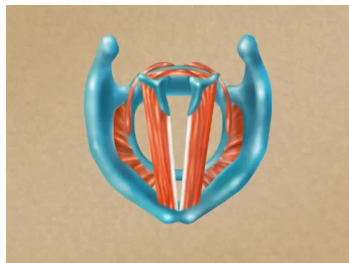
Voiced sounds: chopping-up the air stream

- the vocal cords are closed
 - ⇒ building-up and release of pressure
 - ⇒ pulse train
- results in a periodical pressure and air flow pattern (fundamental frequency; pitch F_0)
- pitch+loudness (intonation) controlled by:
 - ▶ tension in vocal cords
 - ▶ length of vocal cords
 - ▶ air flow rate (pressure difference)
- F_0 of men, women, and kids: [demo3](#)



Voiceless sounds: obstruction of the air stream

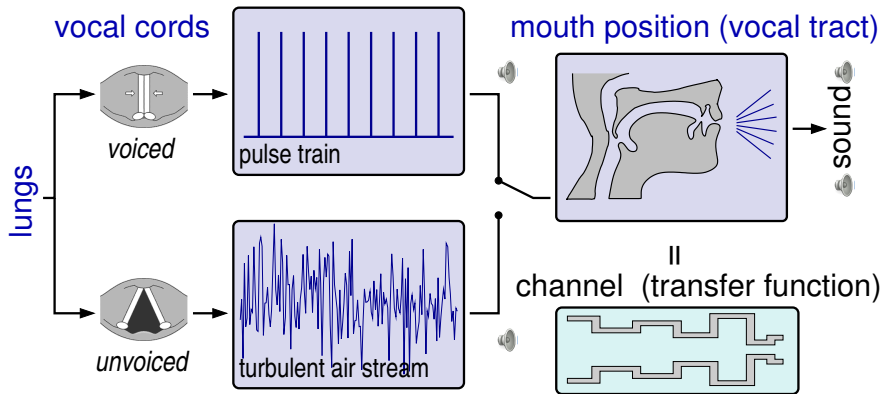
- air flows through a narrow opening
 - ⇒ turbulence
 - ⇒ high frequencies



Obstruents: block air stream + release

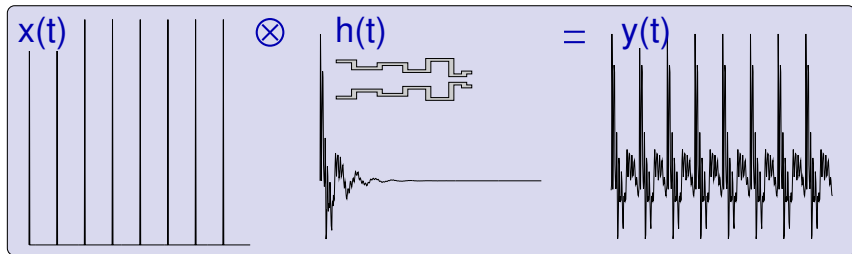
- the vocal cords remain active for voiced obstruents
- plosives (/p/, /b/, /t/, /d/, /k/, /g/)
- fricatives (/f/, /v/, /s/, /z/, /ʃ/, /ʒ/, /x/, /ɣ/, /h/)

Speech production: visualising the process

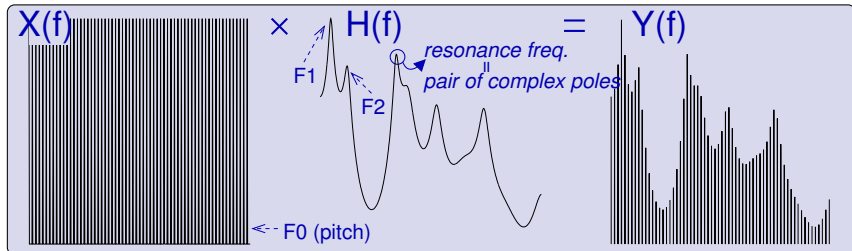


Speech production: visualisation in time & frequency

time domain



frequency domain



Speech production: making the basic sounds (phones)

Manner of articulation:

- how the sound is being made:
 - ▶ degree of obstruction
 - ▶ change in time of the obstruction
 - ▶ which cavities resonate
- determines the phonetic class:
 - ▶ vowels
 - ▶ obstruents: fricatives & plosives
 - ▶ nasals (/m/, /n/, /ɲ/, /ɳ/)
 - ▶ approximants (/r/, /l/, /j/, /w/)

Place of articulation:

- place of the main obstruction in the vocal tract
- further sub-division of the phonetic class
 - ▶ bilabial: (/p/, /b/, /m/)
 - ▶ labiodental: (/f/, /v/, /ɱ/)
 - ▶ velar: (/k/, /g/, /x/, /ɣ/)
 - ▶ ...

Speech production: making the basic sounds (phones)

Consonants:

CONSONANTS (PULMONIC)

place →

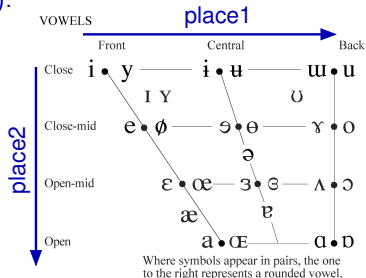
© 2005 IPA

manner ↓

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		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.

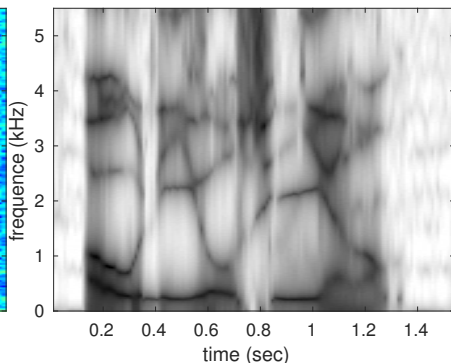
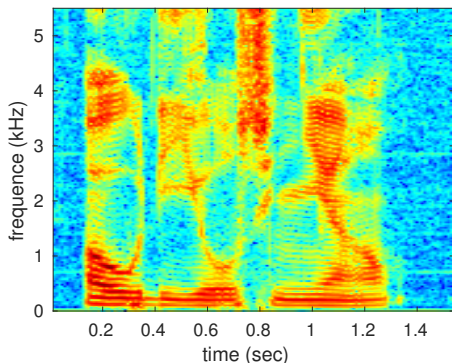
Vowels (manner=vowel):



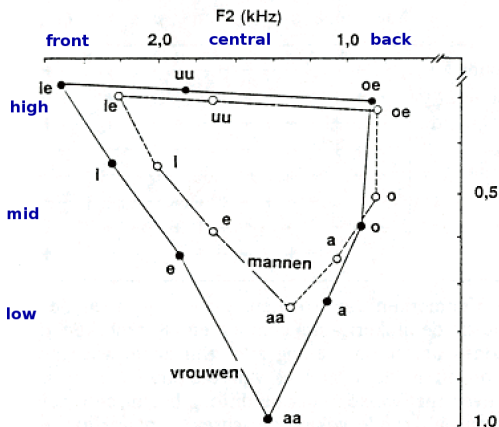
Speech production: making the basic sounds (phones)

The **sound timbre/color** is mainly reflected in the formants (F_1 , F_2 , ...):

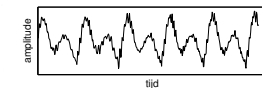
- formant: resonance frequency of a vocal cavity
- the absolute size of the cavity is related to the openness of a sound (F_1)
- the place of articulation determines the relative sizes of the cavities (F_2)
- example: the vowel triangle (see next slides)
- the formants are visible in a spectrogram (left);
linear predictive coding (LPC) (right) shows the formants more clearly



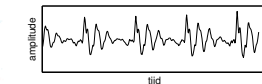
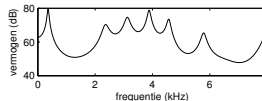
Speech production: the vowel triangle



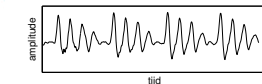
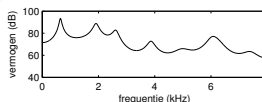
- F_0 : pitch (fundamental freq. of vocal cords)
- F_1 : 1st formant; height of the tongue
- F_2 : 2nd formant; place of the tongue
- approximately one formant per kHz
- F_4, F_5, \dots speaker dependent timbre



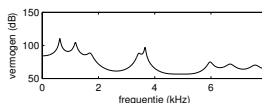
sh/e/



h/a/d



d/a/rk



Speech as information carrier

Verbal information: what is being said

Linguists distinguish different processes:

- syntax : rules concerning sentence structure, e.g. word order;
rules, principles, and processes on how to combine words to sentences
- semantics : the meaning of words, sub-sentences, ...
relations between words (e.g. chair ↔ table)
- pragmatics : influence of context on meaning; examples:
 - speaking style (formal ↔ informal),
 - co-reference (to who/what refers 'he', 'it'),
 - implicit implications, ...
- phonetics : the speech sounds (word pronunciations)
also includes lexical stress
- prosodics : intonation, tone, stress, and rhythm (including pauses)
usage: macro structure (grouping words), emphasis, question, sarcasm, ...

Speech as information carrier

Non-verbal information: how is something being said (and who is speaking)

Also called paralinguistics.

“How something is said” is reflected in:

- pitch, volume (loudness), pauses, speech rate (tempo), modulation, ...
(common denominator: prosodics)

Affected by:

- ▶ mood (angry versus happy),
- ▶ mental state (stress, depression),
- ▶ physical state (gender, age, weight, alcohol, smoking),
- ▶ ...
- fluency (filler words [uhm], vowel lengthening)
- language, accent, dialect

Speech as information carrier

Encoding of the information in the speech signal

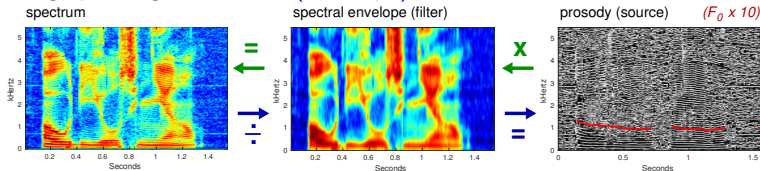
phonetics

- the phone identity is mainly determined by the signal's spectral envelope
- 12 to 13 phones per second, 10.000 words/hour

prosody

- combination of $V(t) + F_o(t) + \text{metrum}$
 - ▶ volume pattern $V(t)$: loudness
 - ▶ intonation pattern $F_o(t)$: melody or intonation
 - ▶ metrum (tempo, temporal structure)
- intonation can encode lexical info as well (tone languages such as Chinese)
- intonation can encode semantic info as well (e.g. Dutch)
(*vóórkomen* (appearance) \leftrightarrow *voorkómen* (prevent))

⇒ splitting spectrogram in filter(envelope) and source



→ analysis: $\text{signal} / \text{filter} = \text{source}$

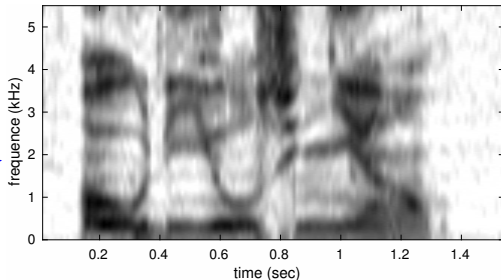
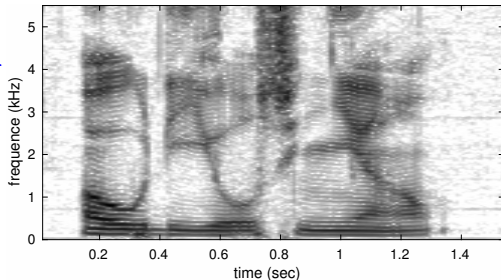
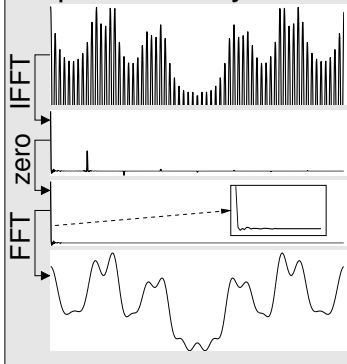
synthesis: $\text{source} \times \text{filter} = \text{signal}$ ←

Speech analysis: envelope (cepstrum)

Fourier-transformatie
 $10 \log_{10}(\|FFT\|^2)$

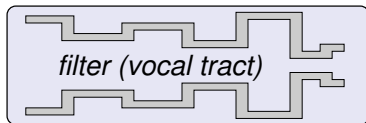
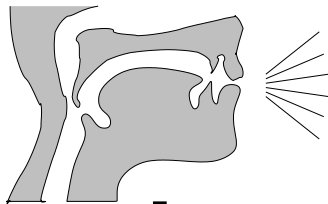


Cepstrale analyse

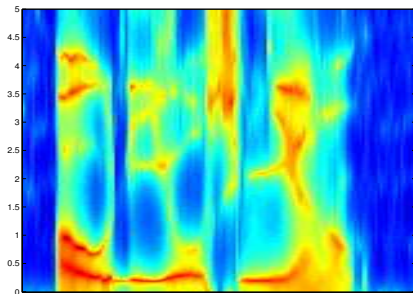
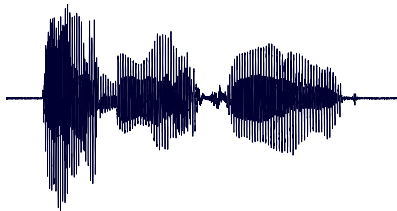


Speech analysis: envelope (LPC)

Linear Predictive Coding



$$Y(z) = \frac{1}{1 - \sum a[j] Y(z^{-j})} E(z)$$



Speech analysis: envelope (LPC)

Linear Predictive Coding

$$y[i] = \sum_{j=1}^n a[j] \times y[i-j] + e[i]$$

speech signal

excitation signal
(white noise)

$$Y(z) = \frac{1}{1 - \sum a[j] z^{-j}} E(z)$$

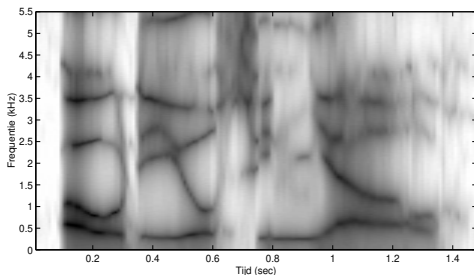


Spectrum

$$E(z)=1 \rightarrow Y(e^{-2\pi f/F})$$

Least Squares Solution

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} y_{-1} & y_{-2} & y_{-3} & \cdots & y_{-n} \\ y_0 & y_{-1} & y_{-2} & & y_{1-n} \\ y_1 & y_0 & y_{-1} & & y_{2-n} \\ \vdots & \vdots & & & \vdots \\ y_{k-1} & y_{k-2} & y_{k-3} & \cdots & y_{k-n} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

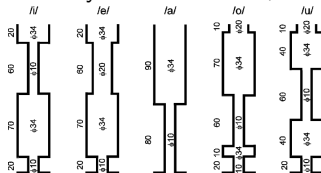


Speech analysis: envelope (LPC)

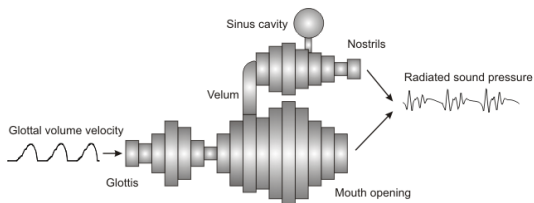
- LPC coefficients can be converted into a tube shape



- works very well for vowels, even when simplified (simple speech synthesis)



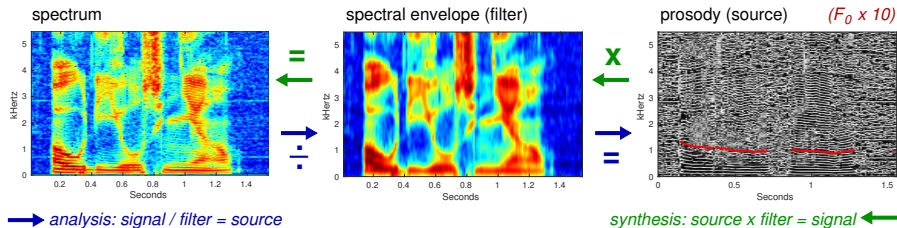
- assumes a simple linear tube, the reality is more complex (e.g. nasals)



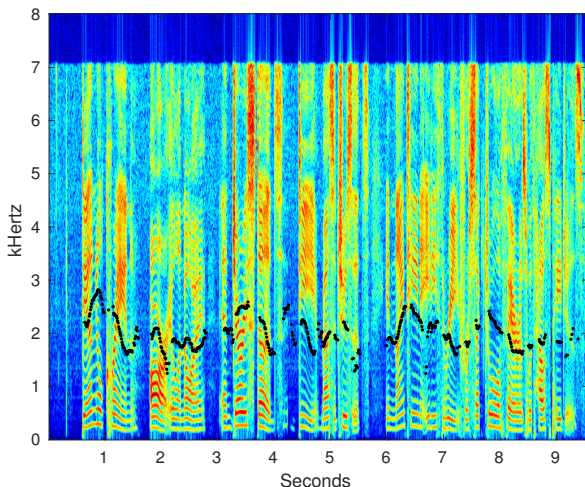
- no closed form solution for complex model (signal \rightarrow filter shape)

Speech analysis: describing the source

splitting spectrogram in filter(envelope) and **source**



Speech analysis: pitch tracking



- a wide variety of algorithms
- clean-up needed to obtain logical pitch tracks
- implemented in toolboxes, e.g. Praat (<http://www.fon.hum.uva.nl/praat>), OpenSMILE (<https://www.audeering.com/technology/opensmile/>), . . .

Speech analysis: jitter, shimmer, voicing quality, ...

Jitter

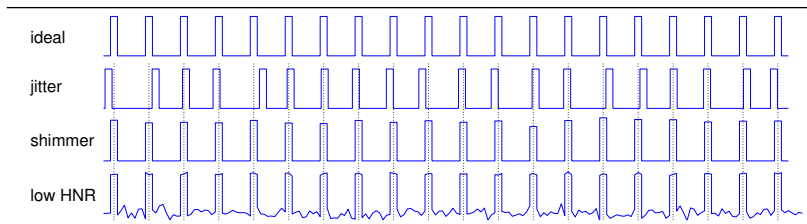
- timing variations on the pitch pulses; perturbation on F_o
- mainly due to lack of control of vocal folds

Shimmer

- amplitude variations on the pitch pulses
- due to reduction of glottic resistance and mass lesions in the vocal folds

Harmonic to noise ratio (HNR)

- percentage of energy that is voiced versus unvoiced



Why + how

- indicates less control over the vocal cords (stress, emotion, ...)
- a wide variety of algorithms; same toolboxes as for pitch tracking

Speech analysis: MFCCs

compact + informative representation of the spectral envelope
filter (timbre) \leftrightarrow source (prosody)

Requirements

- compact + informative
 - ▶ keep all relevant information + suppress/remove non-relevant information
 - ▶ remove non-informative part
 - \Rightarrow model needs less parameters, ergo less data needed to train the model
 - ▶ auditive close together \longleftrightarrow close together in the feature space
- tuned to the modelling techniques being used
 - ▶ requirement for more or less any modelling technique:
same class (phone) \longleftrightarrow close together in the feature space
 - ▶ also: a good feature representation allows simpler models (real-time)
 - ▶ in practice (see later)
 1. decorrelate the feature components of X_n
 2. features that help in describing time evolution
- robust w.r.t. the acoustic environment (noise, reverberation)

MFCCs: the Mel(ody) frequency scale

Principles/ideas:

- speech and hearing are well matched
⇒ look at limitations of hearing to compress the information
- what works for humans, may also work for computers
⇒ **auditive** spectral representation
- envelope ⇒ cepstrum

Spectral analysis in the cochlea:

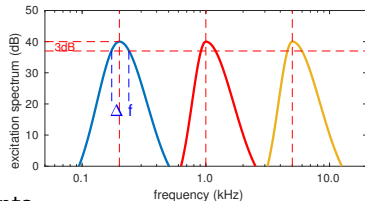
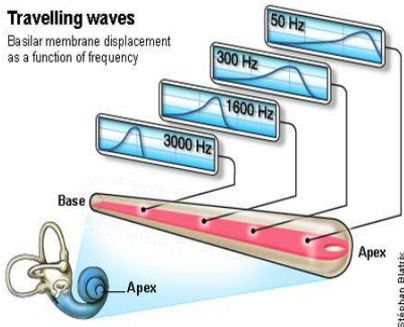
- the ear does frequency analysis (tonotopic)
- position \sim log frequency
- non-linear frequency sensitivity

The Mel subjective frequency scale

- sample in freq. \sim human perception:
$$m(f) = 1127 \log(1 + f/700)$$
- critical band \approx 100 mel
- determined indirectly: perception experiments

Travelling waves

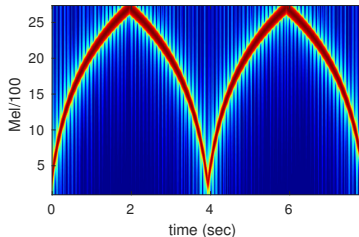
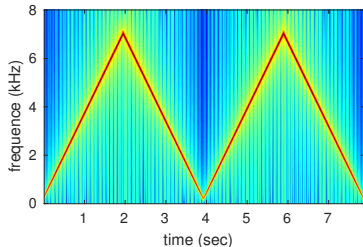
Basilar membrane displacement as a function of frequency



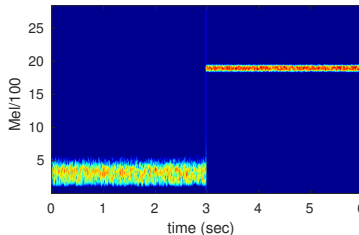
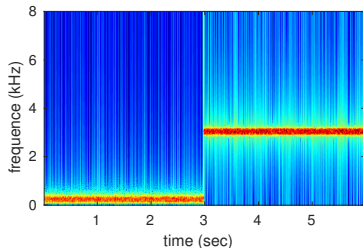
Speech perception: functional description

Subjective frequency scales – examples

- how does a linearly increasing tone sounds (perception)?



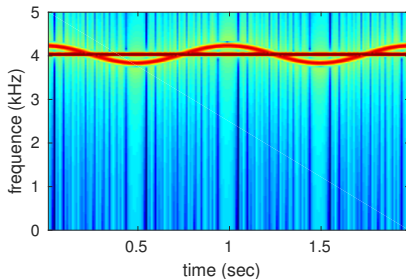
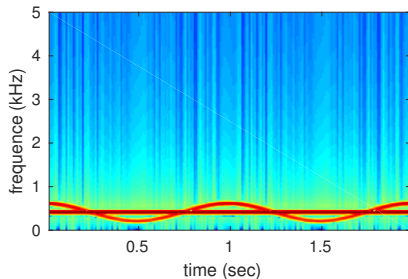
- is this noise with a certain bandwidth or an airy tone?



Speech perception: functional description

Subjective frequency scales – examples

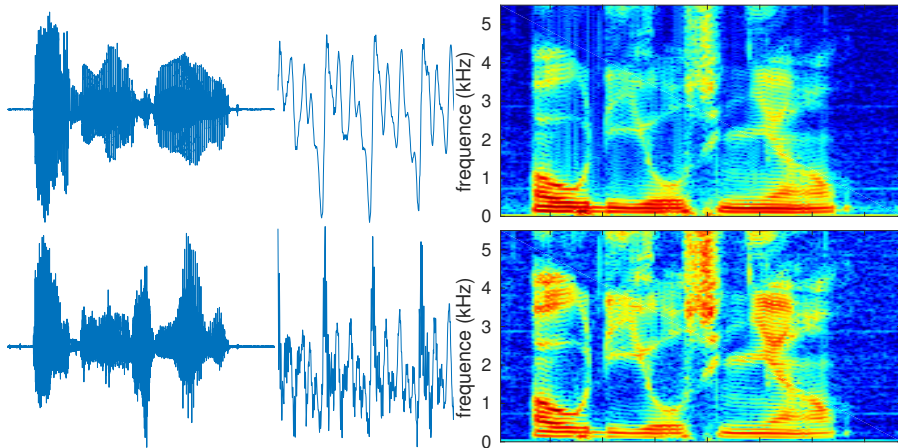
- minimal audible differences between two tones:



MFCCs: computation

Step1: pre-emphasis

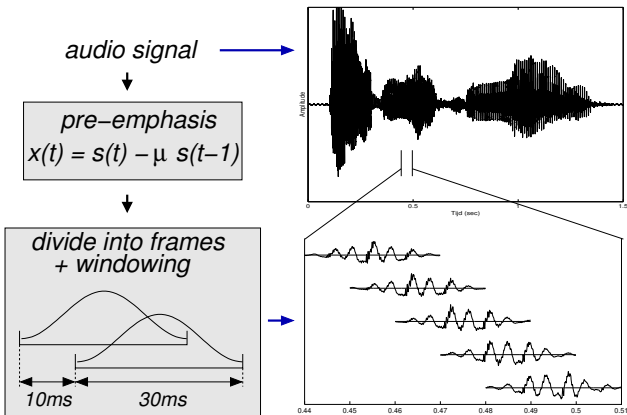
- first-order filter: $x(t) = s(t) - \mu s(t - 1)$
- suppresses DC-components: $\mu = 0.95 \Rightarrow -26\text{dB}$
- amplifies HF-components: $\mu = 0.95 \Rightarrow +6\text{dB}$



MFCCs: computation

Step2: divide into frames + windowing

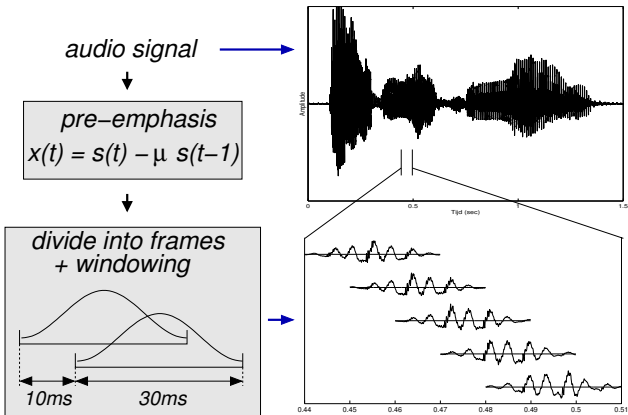
- overlapping frames: $T_f = 10\text{ms}$, $T_w = 30\text{ms} = L$ samples
- 10ms sampling \Rightarrow track the movement of the articulators till 50 Hz
- 30ms window \Rightarrow at least 3 pitch periods (100Hz for a low male voice)
 \Rightarrow stability of the spectral analysis



MFCCs: computation

Step2: divide into frames + windowing

- aim: power spectrum of the signal in a frame (= signal \times rectangle)
- problem: signal transitions at the edges (periodic extension)
→ false frequency components (may mask weak, but informative info)
- solution: multiply with $w(t)$ (smooth, symmetric, bell-shaped)



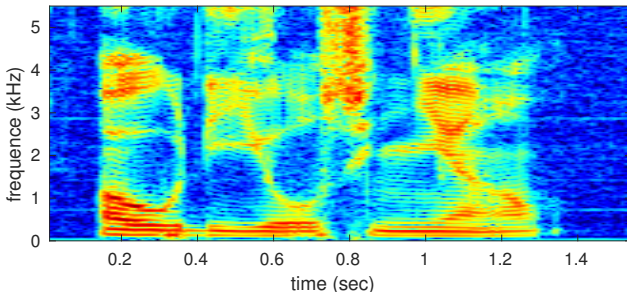
MFCCs: computation

Step3: spectral analysis (DFT, FFT)

- Fourier transform of $x_w(t) = w(t) x(t)$, $t = 0 \dots L - 1$
- zero padding till length N , with N a power of 2
($s(t)$ sampled at 16kHz, $T_w = 30\text{ms} \Rightarrow N = 512$)
- FFT (Fast Fourier Transform):

$$X(m) = \sum_{t=0}^{L-1} x_w(t) e^{-j2\pi mt/N} \quad m = 0, \dots, N-1$$

- $|X(m)|^2$: measure of power in $(m \frac{f_s}{N} - 1.5 \frac{f_s}{L}, m \frac{f_s}{N} + 1.5 \frac{f_s}{L})$, $n = 0 \dots \frac{N}{2}$
- \Rightarrow spectrogram (after taking log)

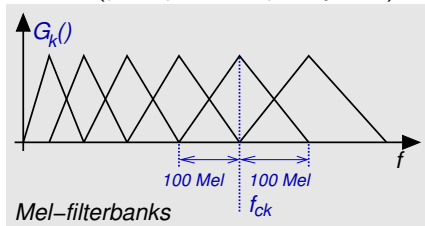


MFCCs: computation

Step4: Mel-spectrum

- $|X(m)|^2$ = measure of power per Hz (linear frequency axis)
- desired: power distribution per critical band (perceptual frequency-axis)
⇒ critical band filters

- ▶ central frequencies f_{ck}
equidistant on Mel-scale:
 $\text{mel}(f) = 1127 \log(1 + f/700)$
- ▶ band width $G_k(f)$:
1 critical band (100 Mel)



- Mel-spectrum by summing the power contributions

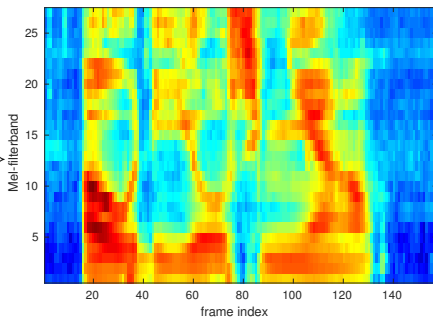
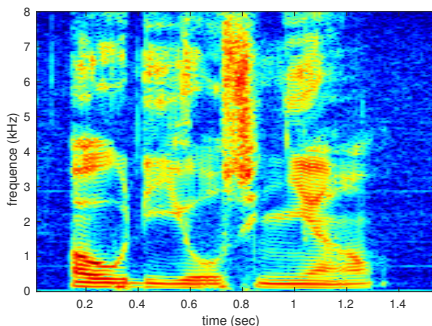
$$S_{\text{mel}}(k) = \sum_{m=0}^{N/2} G_k(m) |X(m)|^2 \quad f_m = \frac{mf_s}{N}$$

- data reduction: from $N/2$ (typical 128...256) to 20...25 numbers

MFCCs: computation

Step5: Compression of the signal range

- human ear: log-power (dB-scale)
- distances between log-spectra are perceptually relevant
- result: log Mel-spectrum $LS_{mel}(k) = \log [S_{mel}(k) + S_0]$



MFCCs: computation

Step6: Mel-cepstra

Desired

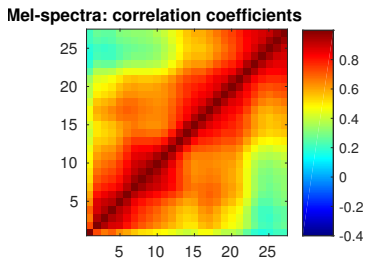
- compact set of parameters
- simple 'distance metric' (or probabilistic model)

Starting point: Mel-spectra

- correlated data (intra-frame)
- correlation between individual parameters frequently indicates redundancies

Solution: Mel-cepstra

- transformation to a minimal set of decorrelated parameters



MFCCs: computation

Step6: Mel-cepstra

How1: (I)DCT of log Mel-spectrum

- decomposes the spectrum in sinusoidal components

$$c_m = \sum_{k=1}^K LS_{mel}(k) \cos \frac{\pi(k - 0.5)m}{K}$$

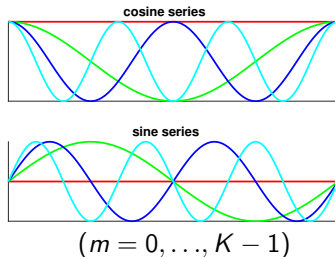
- result: DCT with K degrees of freedom

How2: truncation of the DCT

- c_m , the smaller indices m describe slow variations ($c_0 = \text{log-energy}$)
- c_m , the larger indices m describe fast variations \Rightarrow can be dropped
- the truncated DCT describes the trends in the log Mel-spectrum

Result:

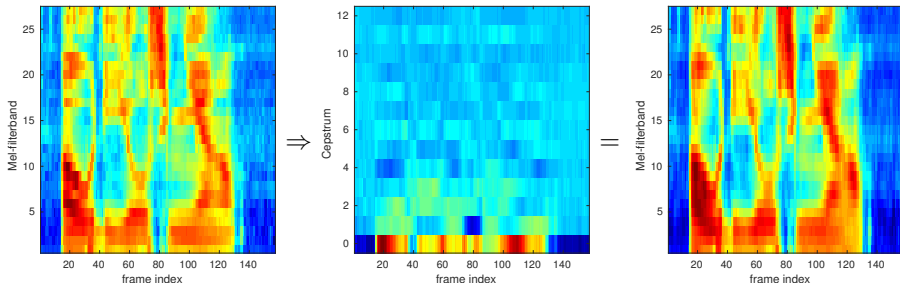
Mel-Frequency Cepstral Coefficients (MFCC)



MFCCs: computation

Step6: Mel-cepstra

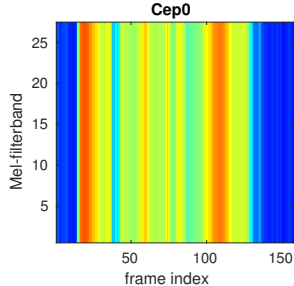
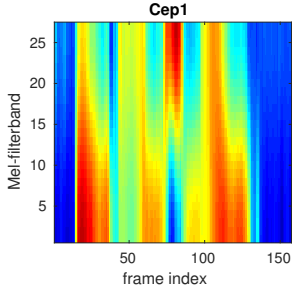
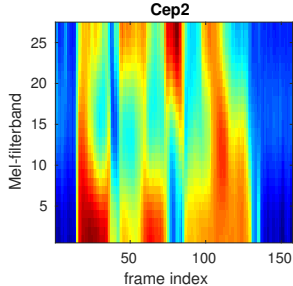
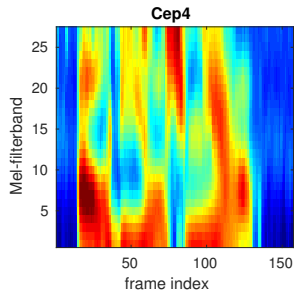
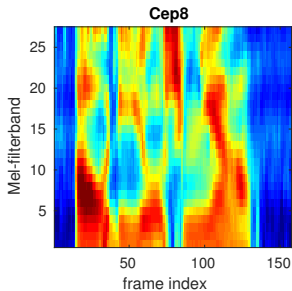
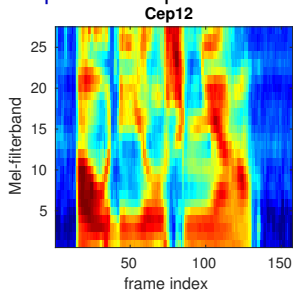
Advantage1: data reduction



- the models have fewer free parameters
- training the models requires less training data
- removes the pitch (F_0)

MFCCs: computation

Step6: Mel-cepstra

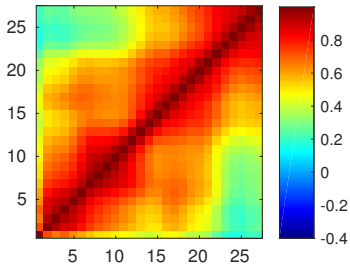


MFCCs: computation

Step6: Mel-cepstra

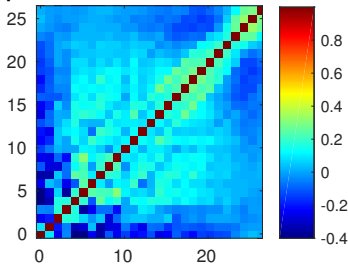
Advantage2: decorrelation \Rightarrow simpler distance metrics / models

Mel-spectra: correlation coefficients



\Rightarrow

Mel-cepstra: correlation coefficients



- Euclidean distances instead of Mahalanobis distances
 - Gaussians with diagonal covariance matrix instead of full matrix
- \Rightarrow fewer free parameters

MFCCs: computation

Step7: time differences

Problem: inter-frame correlations

- consecutive frames show a fair amount of correlation (time evolution)
 - most techniques assume independent feature vectors (simpler models)
 - extra step needed to incorporate the time evolution anyhow
- ⇒ extend the features X_n with dynamic information

Result: vector with 3 components

- static features (X_{nk}): MFCC-vector (dimension = 13)
- first order differences = velocity features (ΔX_{nk})

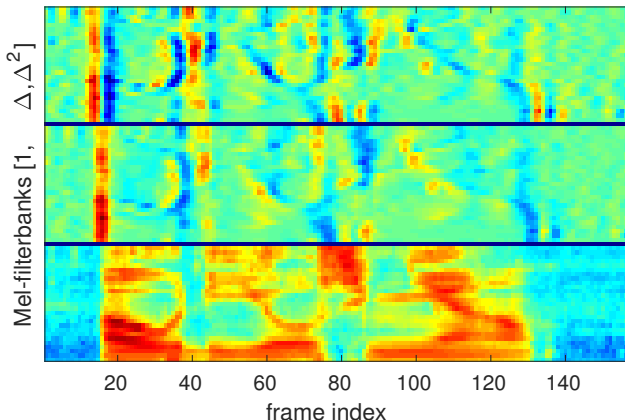
$$\Delta X_{nk} = \frac{\sum_{m=1}^2 m \times (X_{n+m,k} - X_{n-m,k})}{2 \sum_{m=1}^2 m^2}$$

- second order differences = acceleration features ($\Delta^2 X_{nk}$)

$$\Delta^2 X_{nk} = \frac{\sum_{m=1}^2 m \times (\Delta X_{n+m,k} - \Delta X_{n-m,k})}{2 \sum_{m=1}^2 m^2}$$

MFCCs: computation

Step7: differences

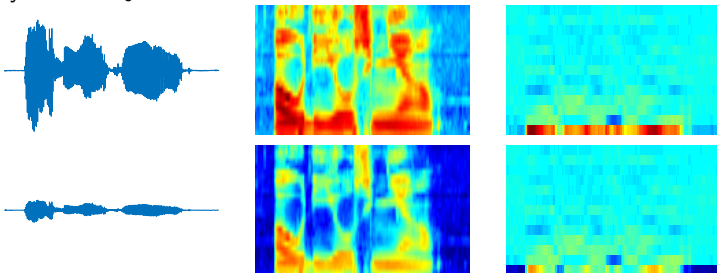


- the fine temporal aspects are handled via the input features
- the models only needs to handle the coarse temporal aspects
- note: the operations $[1, \Delta, \Delta^2]$ and (I)DCT are interchangeable (associativity of linear operators)

MFCCs: computation

Step8: mean-normalisation (optional)

- changes in recording volume
 - offset on log Mel-spectrum
 - only affects c_0



- compensating the features for the volume dependency:
$$c'_0(t) = c_0(t) - \frac{\sum_{t'=1}^T c_0(t')}{T}$$
- assumes that volume differences are not the sole information source to differentiate between two patterns

MFCCs: computation

Step8: mean-normalisation (optional)

- extension to recording channel $h(t)$, $H(f)$ (e.g. freq. characteristic μ -phone)

$s(t)$	$\xrightarrow{h(t)}$	$s(t) \otimes h(t)$	time
$S(f)$	$\xrightarrow{H(f)}$	$S(f) \times H(f)$	<u>spectrum</u>
$\log S(f)$	$\xrightarrow{H(f)}$	$\log S(f) + \log H(f)$	<u>log-spectrum</u>
$\log S_w(t, f)$	$\xrightarrow{H(f)}$	$\log S_w(t, f) + \log H(f)$	<u>log-spectrum; frames</u>
$c_k(t)$	$\xrightarrow{H(f)}$	$c_k(t) + h_k$	<u>cepstrum; frames</u>

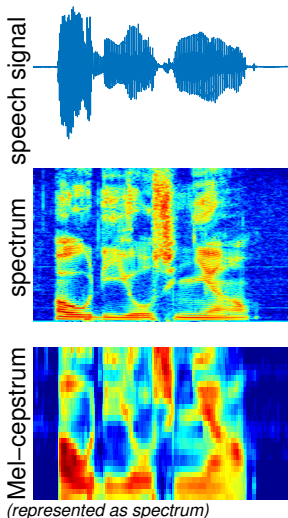
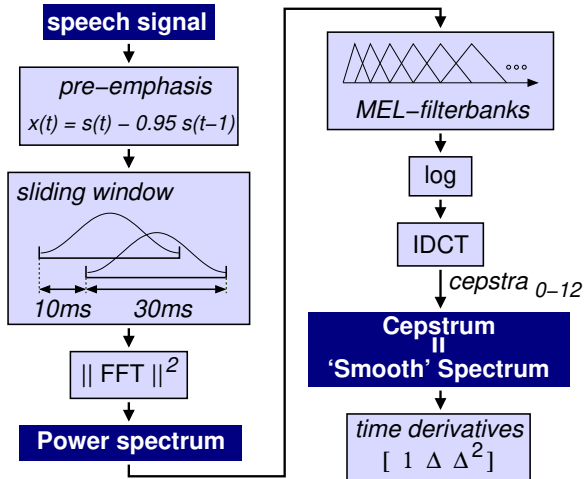
with $S(t, f)$ a non-stationary signal and $H(f)$ a stationary channel

- removing the influence of the recording channel:

$$c'_k(t) = c_k(t) - \frac{\sum_{t'=1}^T c_k(t')}{T}$$

- assumes that $\sum_{t=1}^T c_k(t) = C_k$ (constant);
OK if T is sufficiently large, i.e. $s(t)$ contains enough different phones
- $h(t)$ should not spread the information in $s(t)$ across frame boundaries;
OK for microphone characteristics; not for reverberation
- linear operator \Rightarrow order w.r.t. $[1, \Delta, \Delta^2]$ and (I)DCT can be chosen
- removes, up to a certain extent, also speaker characteristics

MFCCs: summary



■ OpenSMILE:

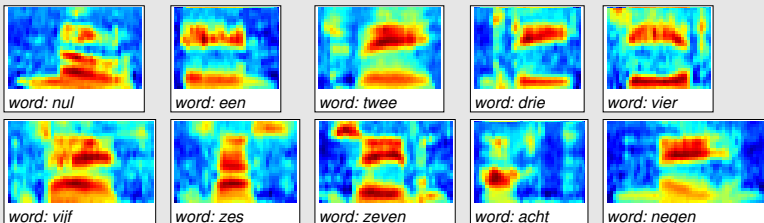
```
SMILEExtract -C config/MFCC12_0.D.A.conf -I sample.wav -csvoutput mfcc.csv
```


MFCCs: use case1 – exemplar-based speech recognition

■ Training

- ▶ make one or more recordings of all words to be recognised
- ▶ the MFCCs of the recorded words form the reference patterns (exemplars)

Training: store one or more reference patterns per word



$$D\left(\begin{array}{c} \text{spectrogram} \\ \text{word: ???} \end{array}, \downarrow \right) = ?$$

Recognition: compare (compute distance) with all reference patterns

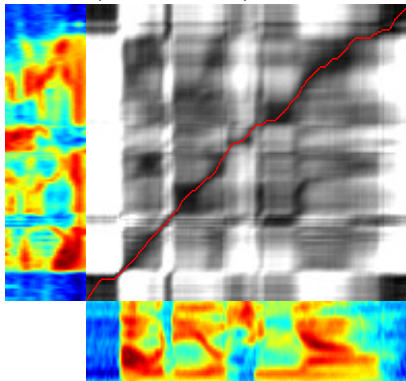
■ Recognition

- ▶ record a word; compare the corresponding MFCCs with all reference patterns
- ▶ recognised word = most similar reference pattern (smallest distance)

Dynamic Time Warping (DTW)

compare vector **time series** showing non-trivial **variations in timing**

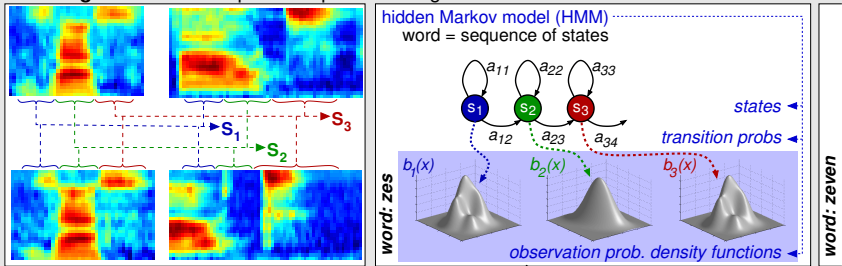
- **Starting point:** the feature extraction generates a sequence of frames
 - ▶ perceptually motivated MFCC features (1 vector per 10ms)
 - ▶ straightforward comparison of two sounds on a 10ms basis (Eucl. dist.)
 - ▶ \neq pronunciations of same word \Rightarrow comparable trajectories in feature space
- **Challenge:** combine the local distance metrics (10ms frames) to a global distance metric between words when every utterance (delivery) of a word or even a phone shows (slightly) different timings
- **Solution:** Dynamic Time Warping
 - ▶ find optimal alignment
 - ▶ sum local distances
- Example: “audiosignaal” ($2\times$)
black-white: local distance
- Demos:
 - `media/dtw_steps.m`
 - `media/dtw_demo.m`



MFCCs: use case2 – statistical speech recognition

■ Training

Training: condense multiple examples in a single model



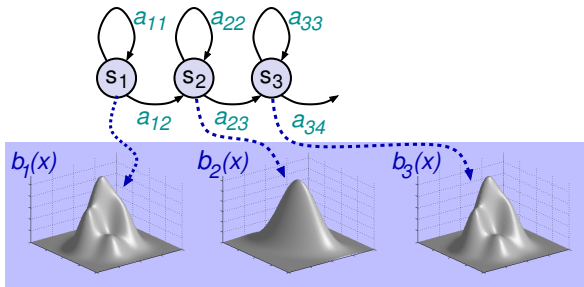
Recognition: compute $p(w|X)$ for all models

$$P\left(\begin{array}{c} \text{[Spectrogram]} \\ \text{word: ???} \end{array} \mid \right) = ?$$

■ Recognition

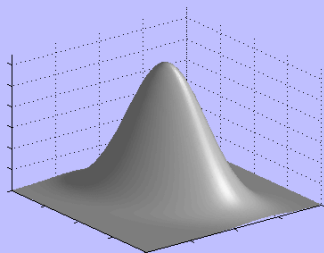
- ▶ compute $p(w|X)$ for all word models
- ▶ recognised word = model that provided the highest likelihood

The “hidden Markov model” (HMM)



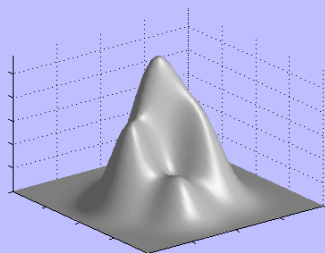
- free parameters:
 - ▶ the number of states
 - ▶ the transition probabilities a_{qs} ($s = q$ or $s = q + 1$), this for every state q
 - ▶ the emission (observation) distributions $b_q(X) = P(X|q)$
- before 2014: Gaussian Mixture Models (GMMs)
 - ▶ 39 dimensional (MFCCs, Δ , Δ^2)
 - ▶ diagonal covariance matrices
 - ▶ parameters: mean, variances, mixture weights
- nowadays: deep neural networks (DNNs) $b_q(X) \sim \frac{P(q|X)}{P(q)}$
 - ▶ MFCCs + Δ (+ Δ^2) or Mel-spectrum or spectrum

The “Gaussian mixture model” (GMM)



Gaussian distributed

$$P(x) = \mathcal{N}(x; \mu, \Sigma)$$



Weighted sum of Gaussians

$$P(x) = \sum \lambda_k \mathcal{N}(x; \mu_k, \Sigma_k)$$

diagonal matrix ←

- in HMMs: emission (observation) **distributions** $b_q(X) = P(X|q)$
- can also be seen as a 1-state HMM (no modelling of time evolution); handy in many applications (see part II)
- parameters:
 - ▶ mixture weights λ (or w or g)
 - ▶ mean vector μ
 - ▶ variances Σ, σ or V (usually a diagonal matrix – requires decorrelation)

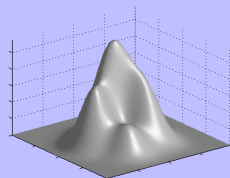
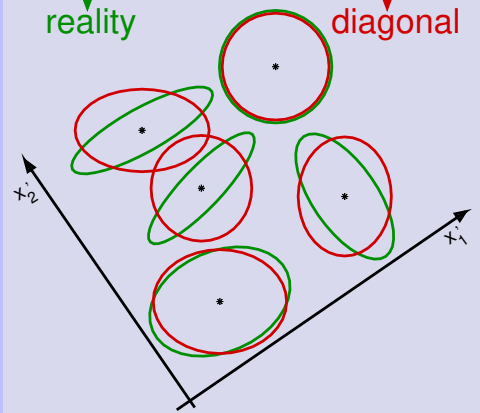
The “Gaussian mixture model” (GMM)

Effect “Cepstrum” – Mel-spectrum \rightarrow Mel-cepstrum (decorrelation)

$$f(x) = \sum \lambda_k \mathcal{N}(x; \mu_k, \Sigma_k)$$

reality

diagonal



\Rightarrow tune features to better match the properties of the acoustic model

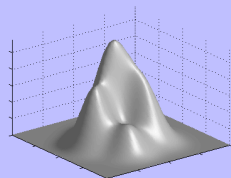
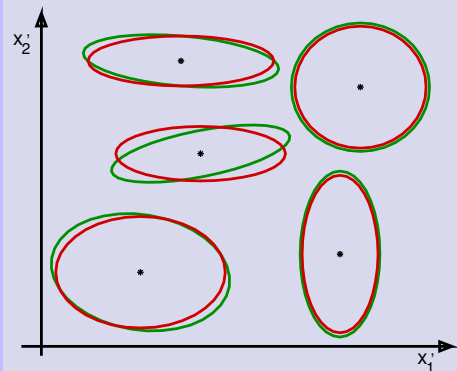
The “Gaussian mixture model” (GMM)

Effect “Cepstrum” – Mel-spectrum \rightarrow Mel-cepstrum (decorrelation)

$$f(x) = \sum \lambda_k \mathcal{N}(x; \mu_k, \Sigma_k)$$

reality

diagonal



\Rightarrow tune features to better match the properties of the acoustic model

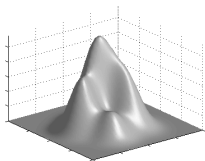
(Deep) Neural Networks (DNNs/MLPs)

- model

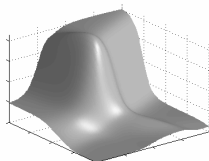
$$p(s|x)$$

instead of

$$p(x|s);$$

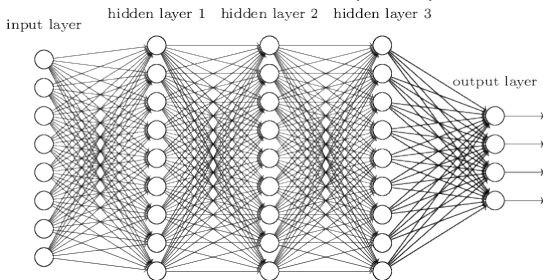


probability density



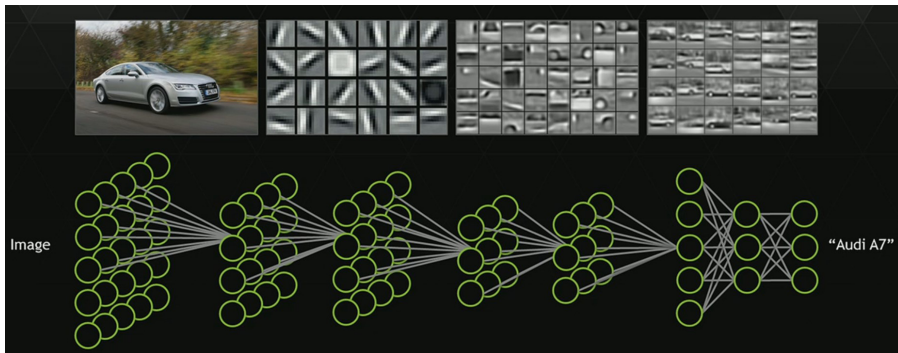
posterior probability

- better: $p(\mathbf{q}|\mathbf{X})$ instead of $p(\mathbf{X}|\mathbf{q})$ \mathbf{q} = state seq.; training via graph
- currently mainly: “deep neural networks” (DNNs)



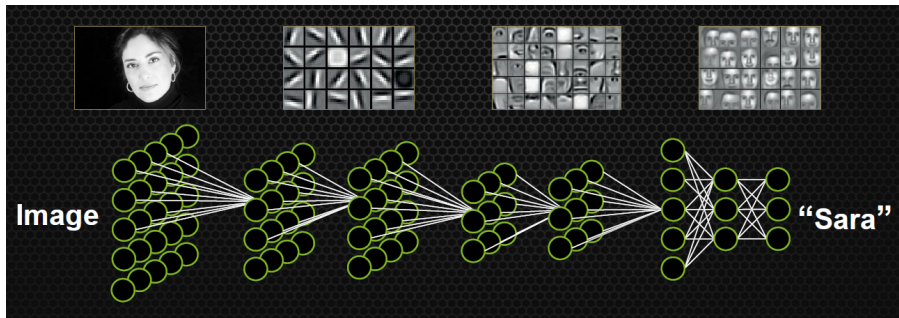
(Deep) Neural Networks (DNNs/MLPs)

- image recognition: 40% \rightarrow 4% error (Cifar-10; 10 image categories)
- hierarchical decomposition of the problem (convolutional NNs; CNNs): works like a charm



(Deep) Neural Networks (DNNs/MLPs)

- image recognition: 40% \rightarrow 4% error (Cifar-10; 10 image categories)
- hierarchical decomposition of the problem (convolutional NNs; CNNs):
works like a charm



(Deep) Neural Networks (DNNs/MLPs)

Why do DNNs finally work

- first wave (begin '90s):
 - ▶ MLP with 2-layers is a universal function approximator
 - ▶ vanishing gradient problem (not known at that time)
 - ⇒ deep did not work
 - ▶ anyhow, only CPU power to handle small MLPs (2 layers)
- what changed:
 - ▶ compute power (GPUs 40× faster than CPUs for DNNs operations)
 - ▶ vanishing gradient problem “solved” (took insight + time)
 - batch normalisation
 - non-linearities with less near-zero gradient regions
 - more iterations
 - ▶ techniques to counteract overfitting (deep = huge → overfitting)

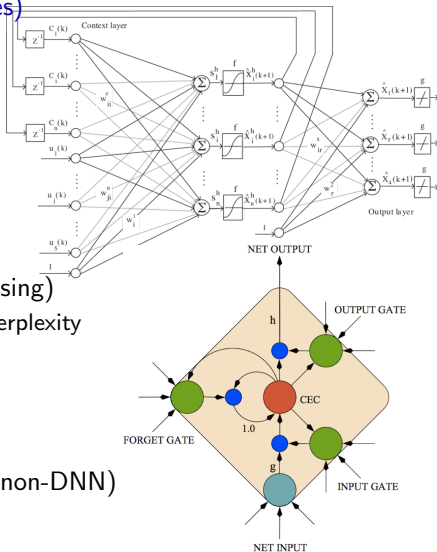
(Acoustic) modelling with DNNs

- speech recognition: 15 to 50% relative improvement
- linguists already made a hierarchical decomposition
 - ⇒ less improvement from the CNNs/DNNs
- structure (hierarchical decomposition) is essential for big improvements

(Deep) Neural Networks (DNNs/MLPs)

Modelling time series (e.g. word sequences)

- recurrent neural networks
- LSTM (long short-term memory)
extra gates so that “neurons” can learn which information is relevant, when to keep information, ...
- fewer gates (3→2→1) works better
- examples (speech & language processing)
 - ▶ language modelling: halving the perplexity compared to N-grams
 - ▶ word vector space models (distance \propto syntax & semantics)
- RNN/LSTM is not an FST!
 \Rightarrow complex to integrate with other (non-DNN) models



Finite State Transducers

Decomposing the problem into smaller elements

search the **words** that fit best with the **sound**



search the most probable **word sequence** (sentence)
given the **speech signal**



$$\hat{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w}} P(\mathbf{w} | \mathbf{X}) = \operatorname{argmax}_{\mathbf{w}} \frac{f(\mathbf{X} | \mathbf{w}) P(\mathbf{w})}{P(\mathbf{X})}$$



$$\hat{\mathbf{w}} = \operatorname{argmax}_{\mathbf{w}} \frac{f(\mathbf{X} | \mathbf{w}) P(\mathbf{w})}{\cancel{P(\mathbf{X})}}$$

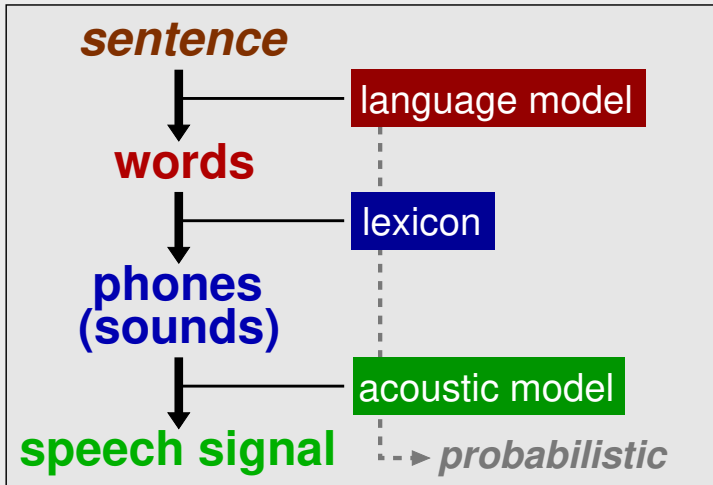
pronunciation information
lexicon & acoustic model

linguistic knowledge
language model

Finite State Transducers

Decomposing the problem into smaller elements

Result: top-down decomposition



Finite State Transducers

Components in a large vocabulary continuous speech recogniser

- basic eqn. using phone-based acoustic models:

$$\hat{\mathbf{w}}, \hat{\mathbf{q}} = \arg \max_{\mathbf{w}, \mathbf{q}} P(\mathbf{X}|\mathbf{q})P(\mathbf{q}|\mathbf{f}_{\mathbf{w}})P(\mathbf{w})$$

- 3 knowledge sources:

acoustic model:	$P(\mathbf{X} \mathbf{q})$	
lexicon:	$w \rightarrow \mathbf{f}_w$	$\Rightarrow P(\mathbf{q} \mathbf{f}_w)$
language model:	$P(\mathbf{w})$	

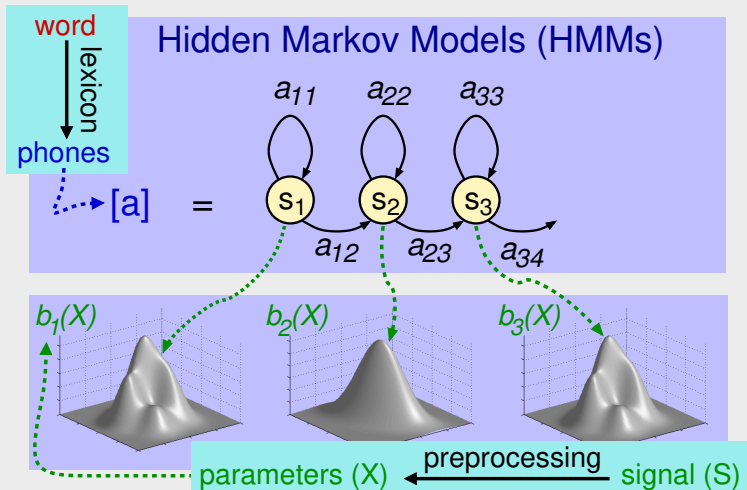
- combining everything
 - ▶ needs an efficient representation of the knowledge sources
 - ▶ needs an efficient search algorithm needed

FST : example1

Acoustic model

■ provides $P(\text{signal}|\text{phones})$

[phones \rightarrow signal]



FST : example2

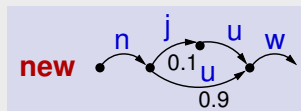
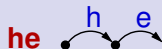
Lexicon

- provides $P(\text{phones}|\text{w})$ $[\text{words} \rightarrow \text{phones}]$
- a straightforward word list (optional: multiple pronunciations):

he h e

is l z

new n [(.1)j / (.9)[]] u w



- extensions:
 - ▶ assimilation rules
 - ▶ context dependent phones (+ tied states)

Language model

- provides $P(\mathbf{w})$ [sentence \rightarrow words]
(a priori probability of a word sequence)
- use in recogniser: progressive [left \rightarrow right] (cf. signal)
 \Rightarrow chain rules (Bayes):

$$\begin{aligned} P(\mathbf{w}) &= \prod_{i=1}^{N_w+1} P(w_i | w_0^{i-1}) & w_0 = \langle s \rangle, \quad w_{N_w+1} = \langle /s \rangle \\ &= \prod_{i=1}^{N_w+1} P(w_i | \text{LM-context}_{i-1}) \end{aligned}$$

- types of knowledge:

syntax

$\left. \begin{matrix} he \\ we \end{matrix} \right\} \leftrightarrow \left\{ \begin{matrix} works \\ work \end{matrix} \right.$

semantics

$\left. \begin{matrix} chair \\ \downarrow \end{matrix} \right\} \left\{ \begin{matrix} sit \\ table \\ legs \end{matrix} \right.$

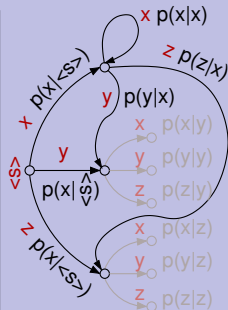
pragmatics

$\underbrace{this, that, he, \dots}_{\downarrow}$
refers to ...

Language model – N -gram language models

predict next word based on the $N-1$ preceeding words

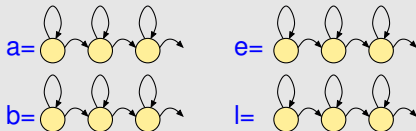
... the united states of ???



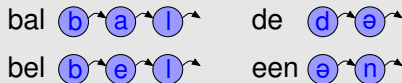
- reduce the context in Bayes chain rule to $N - 1$ words
- is in principle still a Finite State Grammar
 \Rightarrow a Markov-automaton that generates words; each state is an LM context

FST : combining information

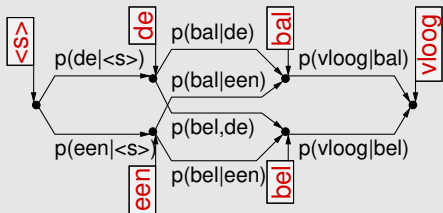
HMM



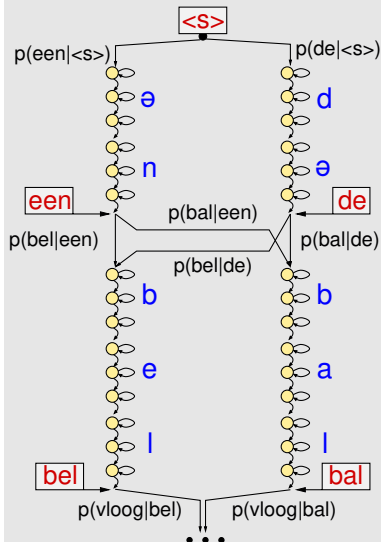
Lexicon (pronouncing dictionary)



Language Model



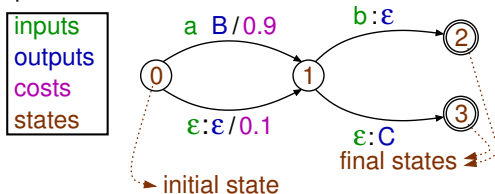
Search Space



Finite State Transducers (FST)

Efficient decoding with Finite State Transducers

- an FST is a finite state machine with:
 - input & output symbols (ϵ : no input/output)
 - converts input to output (transducer)
 - transition probabilities



- most important operator: \circ (compose)

$$\left. \begin{array}{l} \text{FST1 : } \text{abc} \rightarrow \text{ABC} \\ \text{FST2 : } \text{ABC} \rightarrow \alpha\beta\gamma \end{array} \right\} \text{FST1} \circ \text{FST2 : } \text{abc} \rightarrow \alpha\beta\gamma$$

- other frequently used operations:

- ▶ determinisation (left \rightarrow right optimisation, cf. lexicon tree structure)
- ▶ minimisation (cf. lexicon in network structure) + weight pushing
- ▶ ϵ -removal
- ▶ ...

Finite State Transducers (FST)

- determinisation & optimisation are unambiguously defined for loop-free FSTs; approximations needed for FSTs with loops
- all knowledge sources in a speech recogniser are FSTs:

- **Hidden Markov Models**

- H : obs. dens. function id \rightarrow HMM-states

- **context-dependency constraints (decision trees)**

- C : (tied context-dependent) HMM-states \rightarrow phones

- **lexicon**

- L : phones \rightarrow words

- **language model**

- G : words \rightarrow words

\Rightarrow search space:

$H \circ C \circ L \circ G$: obs. dens. function \rightarrow words

- static composition is doable for compact language models
($N \leq 3$ and k in Kneser-Ney is large)
- otherwise: dynamic (on-the-fly) composition of $H \circ C \circ L$ with G

Materials used

Tools

- Python3 (3.6 or later)
 - ▶ scipy (includes numpy)
 - ▶ matplotlib
 - ▶ optional: pyaudio or sounddevice
- OpenSMILE (v2.3)
 - ▶ <https://www.audeering.com/technology/opensmile/>
 - ▶ documentation: openSMILE-book-latest.pdf

Scripts

- spectrogram.py
 - ▶ make spectrogram
 - ▶ plot pitch (F_0)
 - ▶ generate artificial signal (freq. sweep, pulse train, noise)
 - ▶ play the audio signal
- demo{1,2,3}.py

Audio material

- speech @ 16kHz: audiosignal.wav, male.wav, female.wav, boy.wav, girl.wav
- music @ 16kHz: piano_short.wav, drum_short.wav, percussion_short.wav, music_short.wav