Crash Course in Speech Signal Processing and Recognition

Giampiero Salvi

KTH CSC TMH giampi@kth.se
UTL IST ISR gsalvi@isr.ist.utl.pt

Vislab, Mar. 2007

Outline

Outline

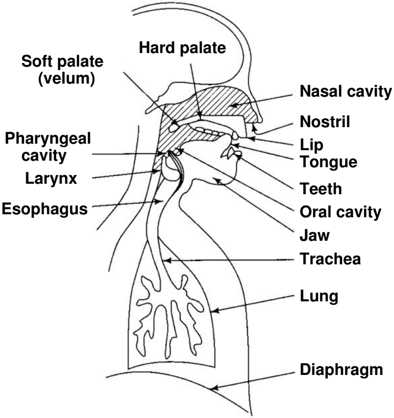
Models of Speech Production
- Vowel-like sounds
- Source/Filter Model, General Case

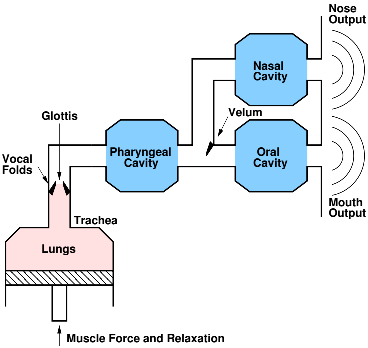
Acoustic Features
- Linear Prediction Analysis (LPA)
- Mel Frequency Cepstral Coefficients (MFCC)
- Features and Time Evolution

Hidden Markov Models (HMMs) and Automatic Speech Recognition (ASR)
- Definition
- Three problems
- Warnings

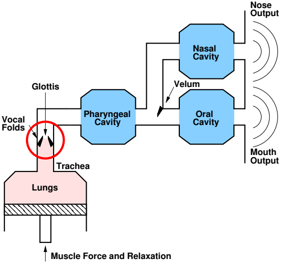
CONTACT Challenges

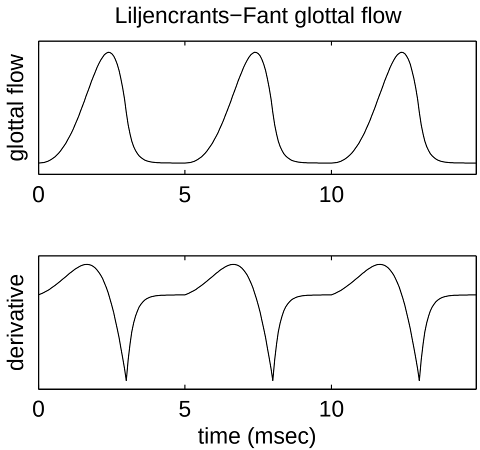
Physiology


Hard palate
Soft palate
(velum)
Nasal cavity
Nostril
Pharyngeal
cavity
Lip
Tongue
Larynx
Teeth
Esophagus
Oral cavity
Jaw
Trachea
Lung
Diaphragm


Nose
Output
Nasal
Cavity
Velum
Glottis
Pharyngeal
Cavity
Oral
Cavity
Vocal
Folds
Mouth
Output
Trachea
Lungs
Muscle Force and Relaxation

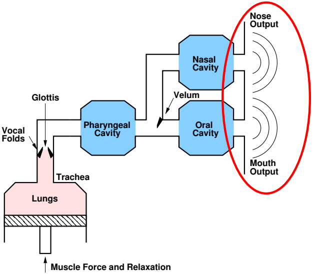
Glottal Flow





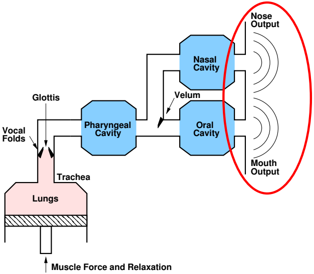
$$G(z) = \frac{1}{(1-\beta z)^2}, \quad \beta < 1$$

Radiation form the Lips/Nose

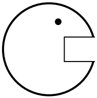


Problem of radiation at the lips plus diffraction about the head too complicated.

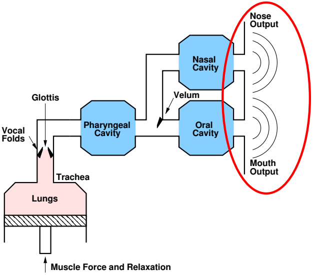
Radiation form the Lips/Nose



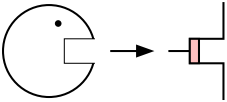
Approx. with a piston in a rigid sphere: solved but not in closed form



Radiation form the Lips/Nose

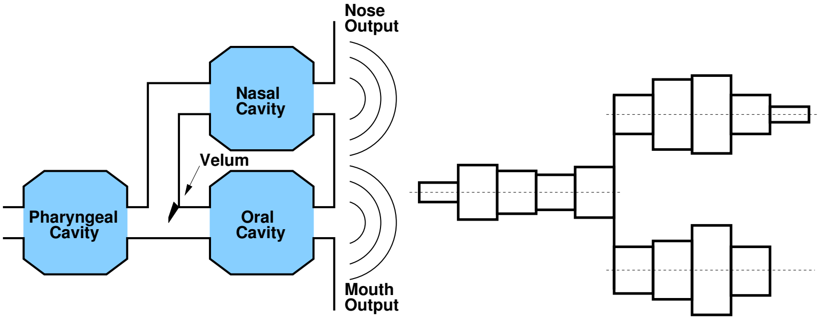


2nd approx: piston in an infinite wall

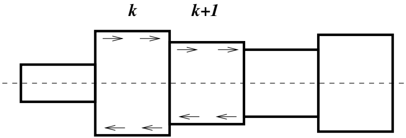


$$R(z) \approx 1 - \alpha z^{-1}$$

Tube Model of the Vocal Tract



Tube Model (cntd.)

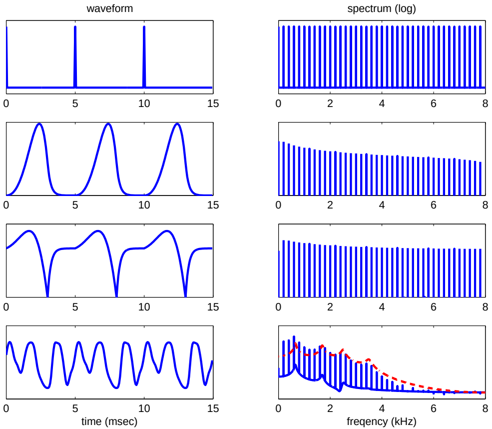


- assume planar wave propagation and lossless tubes
- solve pressure $p(x, t)$ and velocity $u(x, t)$ in each tube according to wave equation
- impose continuity of pressure and velocity at the junctions

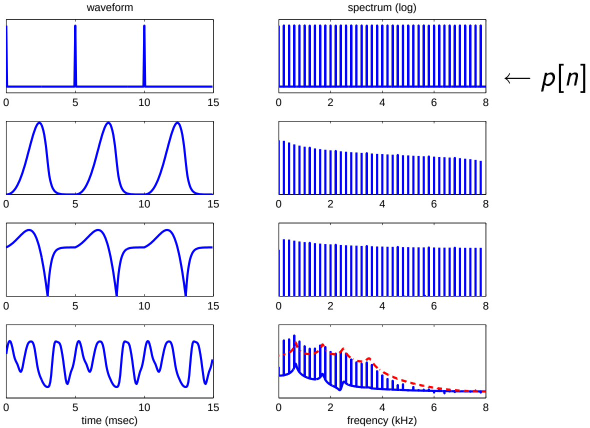
$\Rightarrow$ all-pole transfer function (N = number of tubes)

$$V(z) = \frac{Az^{-N/2}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$

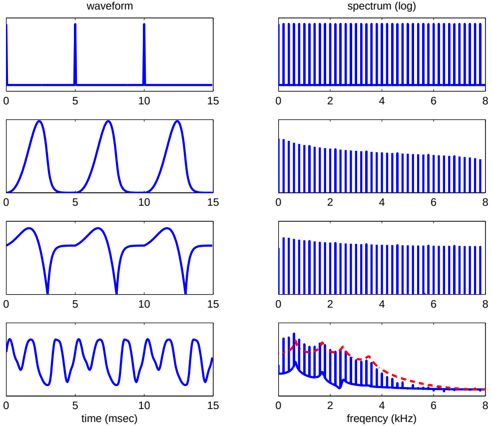
Source/Filter Model: vowel-like sounds



Source/Filter Model: vowel-like sounds



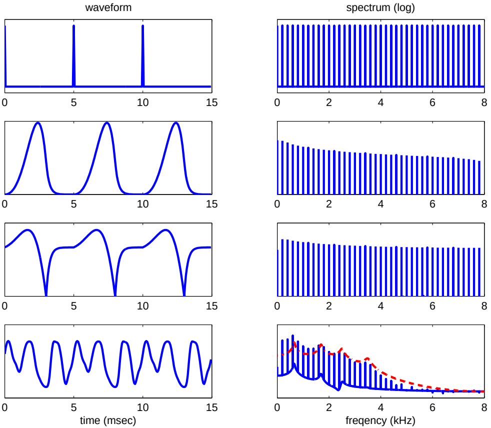
Source/Filter Model: vowel-like sounds



$\leftarrow p[n]$

$\leftarrow p[n] * g[n]$

Source/Filter Model: vowel-like sounds

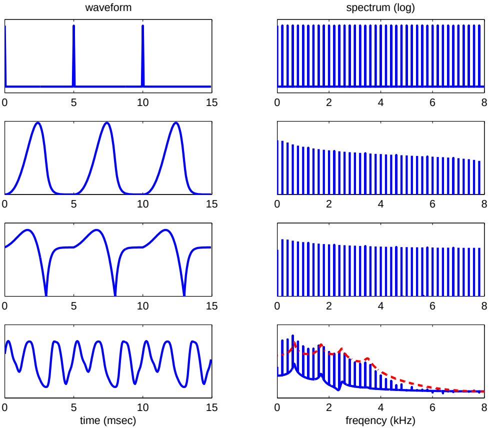


$\leftarrow p[n]$

$\leftarrow p[n] * g[n]$

$\leftarrow p[n] * g[n] * r[n]$

Source/Filter Model: vowel-like sounds



$\leftarrow p[n]$

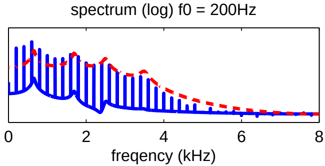
$\leftarrow p[n] * g[n]$

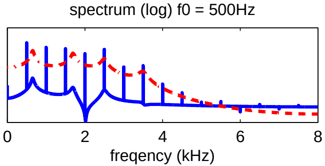
$\leftarrow p[n] * g[n] * r[n]$

$\leftarrow p[n] * g[n] * r[n] * v[n]$

F_0 and Formants

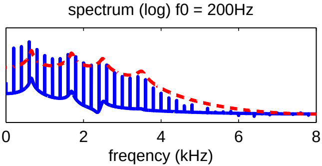
- Varying F_0 (vocal fold oscillation rate)

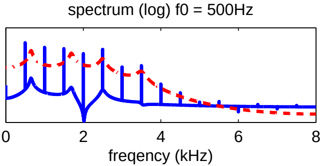




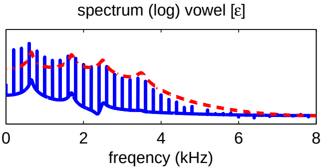
F_0 and Formants

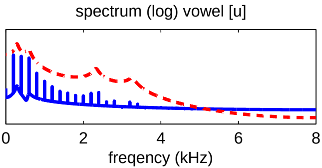
- Varying F_0 (vocal fold oscillation rate)





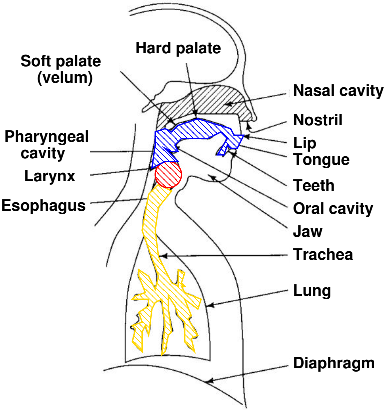
- Varying Formants (vocal tract shape)

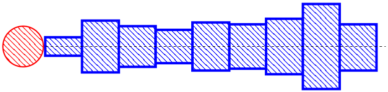




Source/Filter Model, General Case

Vowels

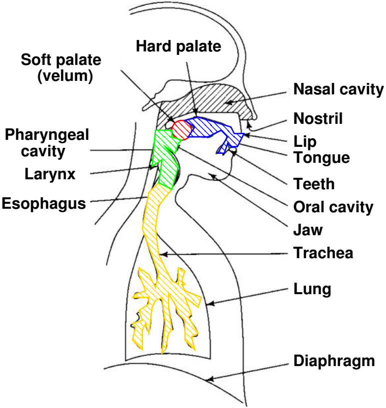


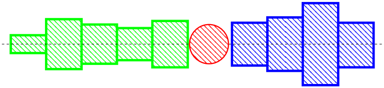


- Source (periodic)
- Front Cavity
- Back Cavity
- Back Cavity (2nd approx.)

Source/Filter Model, General Case

Fricatives (e.g. /ɧ/) or Plosive (e.g. /k/)

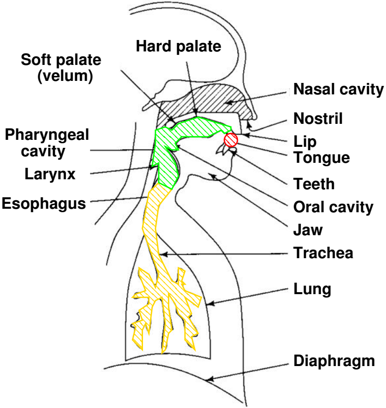


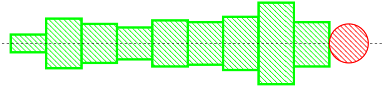


- Source (noise or impulsive)
- Front Cavity
- Back Cavity
- Back Cavity (2nd approx.)

Source/Filter Model, General Case

Fricatives (e.g. /s/) or Plosive (e.g. /t/)

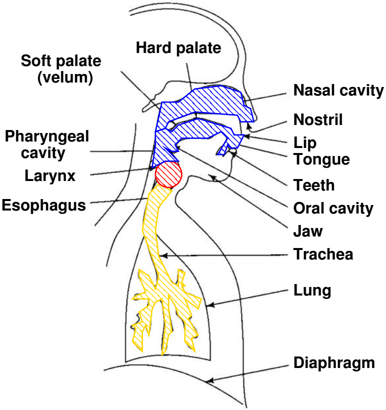


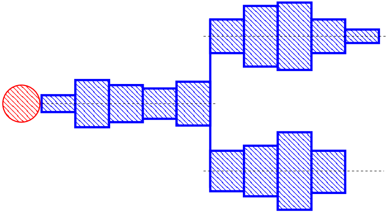


- Source (noise or impulsive)
- Front Cavity
- Back Cavity
- Back Cavity (2nd approx.)

Source/Filter Model, General Case

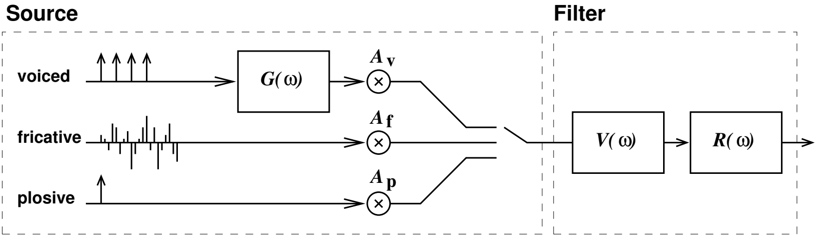
Nasalised Vowels



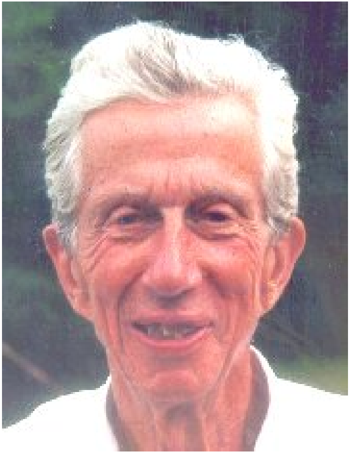


- Source (periodic)
- Front Cavity
- Back Cavity
- Back Cavity (2nd approx.)

Complete Source/Filter Model



Gunnar Fant



Outline

Linear Prediction Coefficients (LPC)

- assume all-pole model:

$$H(z) = \frac{S(z)}{U_g(z)} = AG(z)V(z)R(z) \triangleq \frac{A}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

Linear Prediction Coefficients (LPC)

- assume all-pole model:

$$H(z) = \frac{S(z)}{U_g(z)} = AG(z)V(z)R(z) \triangleq \frac{A}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

- the output signal $s[n]$ can be expressed as the sum of the input $u_g[n]$ and a number of previous samples $a_k s[n-k]$:

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + A u_g[n]$$

Linear Prediction Coefficients (LPC)

- assume all-pole model:

$$H(z) = \frac{S(z)}{U_g(z)} = AG(z)V(z)R(z) \triangleq \frac{A}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

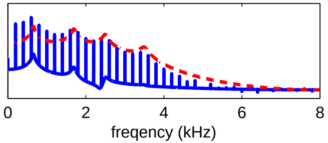
- the output signal $s[n]$ can be expressed as the sum of the input $u_g[n]$ and a number of previous samples $a_k s[n-k]$:

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + A u_g[n]$$

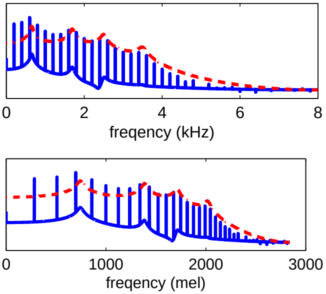
- given a linear predictor α_k of a_k, minimise the error:

$$e[n] = s[n] - \tilde{s}[n] = s[n] - \sum_{k=1}^{p} \alpha_k s[n-k]$$

Mel Frequency Cepstral Coefficients

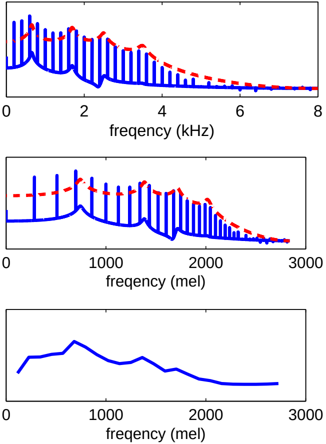


Mel Frequency Cepstral Coefficients



Linear to Mel frequency

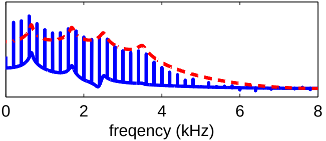
Mel Frequency Cepstral Coefficients



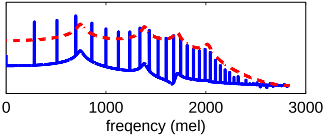
Linear to Mel frequency

log() + Filterbank ($\sim$ 20-25 filters)

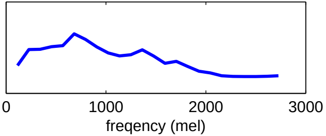
Mel Frequency Cepstral Coefficients



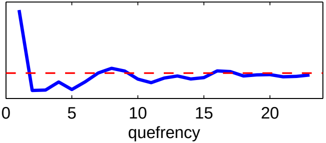
Linear to Mel frequency



log() + Filterbank ($\sim$ 20-25 filters)



Discrete Cosine Transform



MFCC (cntd.)

Rationale

- signals combined in a convolutive way: $a[n] * b[n] * c[n]$

MFCC (cntd.)

Rationale

- signals combined in a convolutive way: $a[n] * b[n] * c[n]$
- in the spectral domain: $A(z)B(z)C(z)$

MFCC (cntd.)

Rationale

- signals combined in a convolutive way: $a[n] * b[n] * c[n]$
- in the spectral domain: $A(z)B(z)C(z)$
- taking the log: $\log(A(z)) + \log(B(z)) + \log(C(z))$

MFCC (cntd.)

Rationale

- signals combined in a convolutive way: $a[n] * b[n] * c[n]$
- in the spectral domain: $A(z)B(z)C(z)$
- taking the log: $\log(A(z)) + \log(B(z)) + \log(C(z))$
- to analise the different contribution perfor Fourier transform (DCT if not interested in phase information).

MFCC (cntd.)

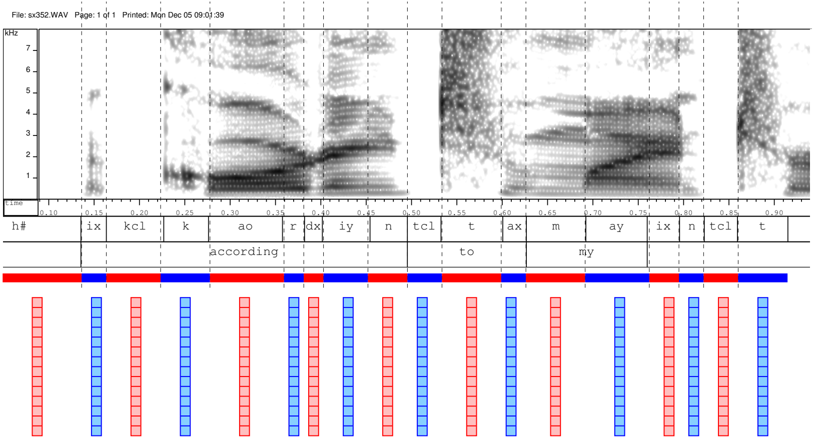
Rationale

- signals combined in a convolutive way: $a[n] * b[n] * c[n]$
- in the spectral domain: $A(z)B(z)C(z)$
- taking the log: $\log(A(z)) + \log(B(z)) + \log(C(z))$
- to analise the different contribution perfor Fourier transform (DCT if not interested in phase information).

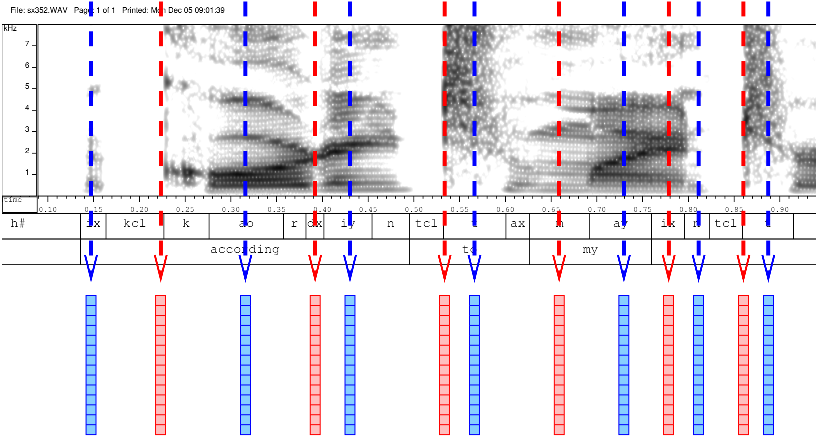
Advantages

- fairly uncorrelated coefficients (simpler statistical models)
- do not assume all-pole model

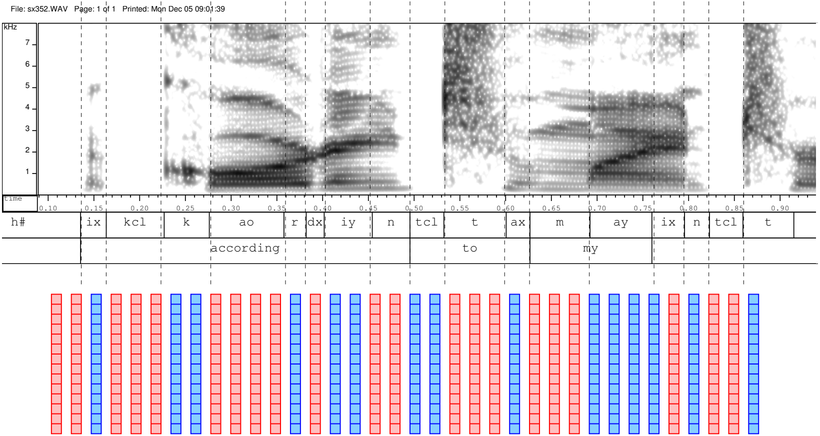
Segment-Based Processing



Landmark-Based Processing



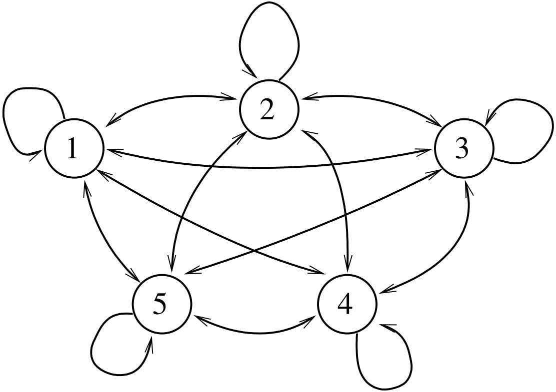
Frame-Based Processing



Outline

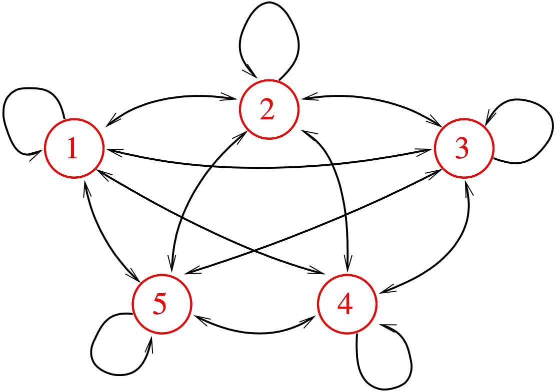
Hidden Markov Models

An HMM is defined by:



Hidden Markov Models

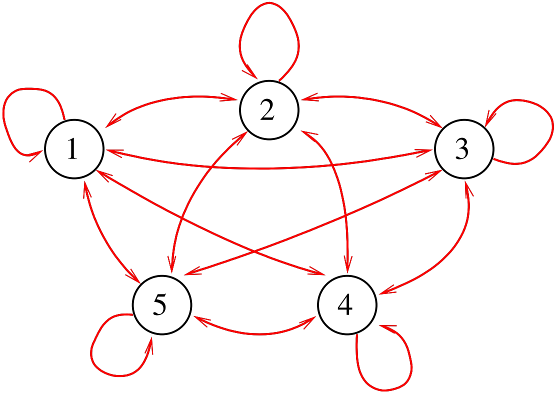
An HMM is defined by:



a set of N reachable states $S = \{s_1, s_2, ..., s_N\}$

Hidden Markov Models

An HMM is defined by:

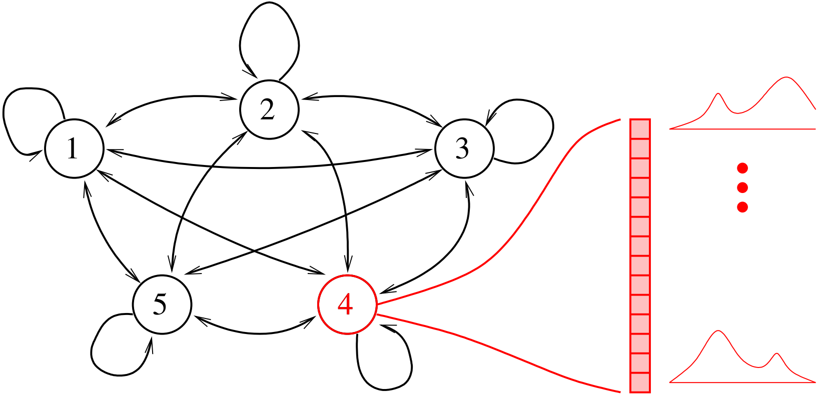


a state transition probability distribution $A = \{a_{ij}\}$ where

$$a_{ij} = Prob\{x_{t+1} = s_j | x_t = s_i\}$$

Hidden Markov Models

An HMM is defined by:

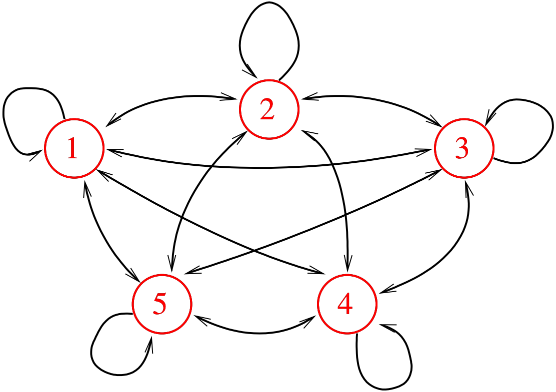


the probability distribution of an observation $\mathbf{o}_t \in \mathbb{R}^M$ given the state s_j,

$$b_j(\mathbf{o}_t) = P(\mathbf{o}_t | x_t = s_j)$$

Hidden Markov Models

An HMM is defined by:

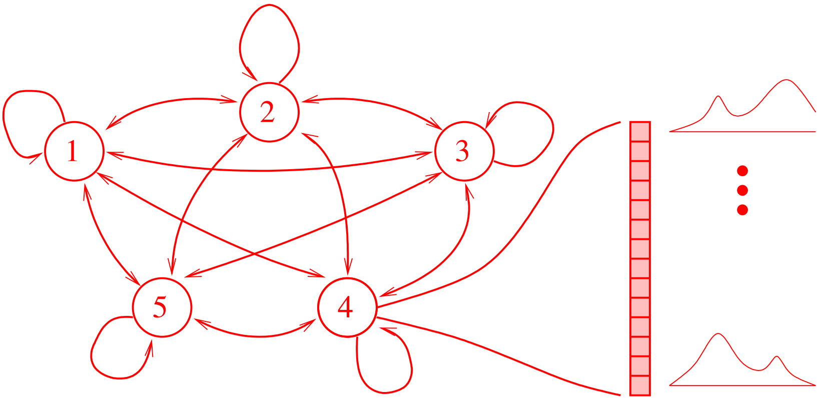


the initial state distribution $\pi = \{\pi_i\}$ where

$$\pi_i = Prob\{x_1 = s_i\}, \forall i \in [1, N]$$

Hidden Markov Models

An HMM is defined by:

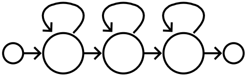


$$\lambda = \{S, \mathbb{R}^M, \pi, A, B\}$$

Example 1: Isolated Word Recognition

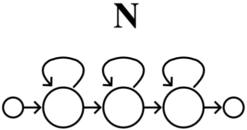
- each phoneme is modelled by a three-state left-to-right HMM:

N

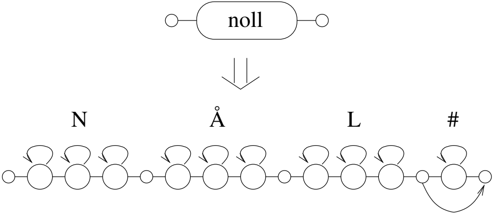


Example 1: Isolated Word Recognition

- each phoneme is modelled by a three-state left-to-right HMM:

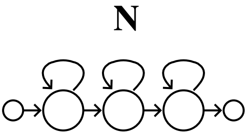


- each word is modelled as a sequence of phonemes:

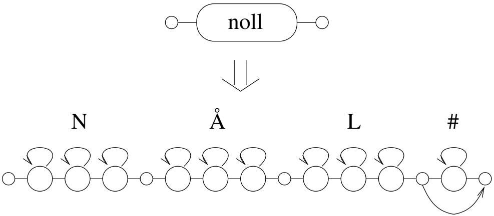


Example 1: Isolated Word Recognition

- each phoneme is modelled by a three-state left-to-right HMM:



- each word is modelled as a sequence of phonemes:



- there are two words in the vocabulary: “noll” (zero) and “ett” (one).

Example 1: Isolated Word Recognition (cntd.)

Problem: given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$, containing just one word, decide if the spoken work was “noll” or “ett”.

Example 1: Isolated Word Recognition (cntd.)

Problem: given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$, containing just one word, decide if the spoken work was "noll" or "ett".

Solution: compute the likelihood of the observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$ given the model: $P(\mathbf{O}|\lambda_i)$ for each model (word) and select $\arg\max_i P(\mathbf{O}|\lambda_i)$.

Example 1: Isolated Word Recognition (cntd.)

Problem: given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$, containing just one word, decide if the spoken work was “noll” or “ett”.

Solution: compute the likelihood of the observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$ given the model: $P(\mathbf{O}|\lambda_i)$ for each model (word) and select $\arg\max_i P(\mathbf{O}|\lambda_i)$.

Problem: Summing the log likelihood over the possible paths is not feasible.

Example 1: Isolated Word Recognition (cntd.)

Problem: given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$, containing just one word, decide if the spoken work was "noll" or "ett".

Solution: compute the likelihood of the observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$ given the model: $P(\mathbf{O}|\lambda_i)$ for each model (word) and select $\arg\max_i P(\mathbf{O}|\lambda_i)$.

Problem: Summing the log likelihood over the possible paths is not feasible.

Solution: Forward-Backward algorithm

$$\alpha_t(i) = Prob(\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_t, x_t = s_i|\lambda)$$

$$\beta_t(i) = Prob(\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, ..., \mathbf{o}_T|x_t = s_i; \lambda)$$

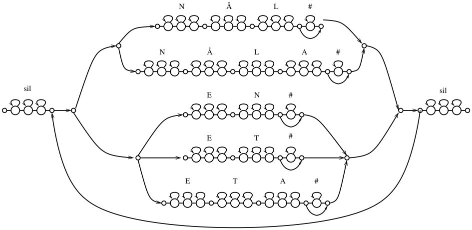
Example 2: Continuous Speech Recognition

Problem: given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$, containing a sequence of “noll” or “ett”, reconstruct the sequence.

Example 2: Continuous Speech Recognition

Problem: given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$, containing a sequence of “noll” or “ett”, reconstruct the sequence.

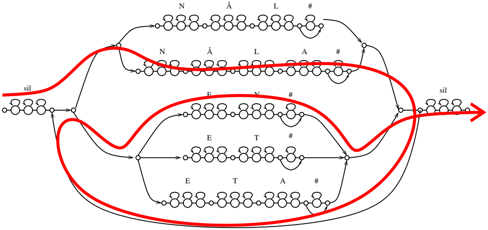
Solution (1): build an HMM describing the possible sequence of words:



Example 2: Continuous Speech Recognition

Problem: given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$, containing a sequence of "noll" or "ett", reconstruct the sequence.

Solution (1): build an HMM describing the possible sequence of words:

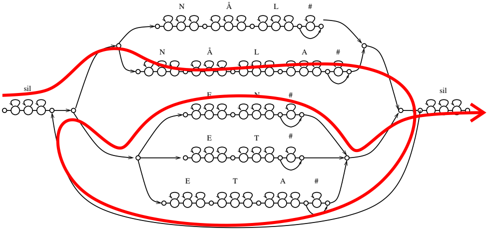


Solution (2): find the best path in the full model, given $\mathbf{O}$

Example 2: Continuous Speech Recognition

Problem: given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$, containing a sequence of "noll" or "ett", reconstruct the sequence.

Solution (1): build an HMM describing the possible sequence of words:



Solution (2): find the best path in the full model, given $\mathbf{O}$

Implementation: Viterbi algorithm

Example 3: Training

Problem: given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$, containing a **known** sequence of “noll” or “ett”, find the best values of $\lambda_i = \{\pi_i, A_i, B_i\}$.

Note: the association between HMM states and time steps is not known

Example 3: Training

Problem: given an observation sequence $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$, containing a **known** sequence of “noll” or “ett”, find the best values of $\lambda_i = \{\pi_i, A_i, B_i\}$.

Note: the association between HMM states and time steps is not known

Solution: Baum-Welsh Algorithm (instance of the Expectation Maximisation algorithm).

Warnings

- phones are **not** stationary sounds

Warnings

- phones are **not** stationary sounds
- phones are strongly affected by context

Warnings

- phones are **not** stationary sounds
- phones are strongly affected by context
- difference between phonemes (lexicon) and phones (sounds)

Warnings

- phones are **not** stationary sounds
- phones are strongly affected by context
- difference between phonemes (lexicon) and phones (sounds)
- assimilation, co-articulation, reduction...

Warnings

- phones are **not** stationary sounds
- phones are strongly affected by context
- difference between phonemes (lexicon) and phones (sounds)
- assimilation, co-articulation, reduction...
- spontaneous speech (!)

Outline

CONTACT Challenges

- the phonemes (speech categories) are not known in advance

CONTACT Challenges

- the phonemes (speech categories) are not known in advance
- the words are not given

CONTACT Challenges

- the phonemes (speech categories) are not known in advance
- the words are not given
- infer phonemes and words from experience (unsupervised/reinforcement learning)

CONTACT Challenges

- the phonemes (speech categories) are not known in advance
- the words are not given
- infer phonemes and words from experience (unsupervised/reinforcement learning)
- build associations between sounds (words) and images (objects) by interacting with the environment.

CONTACT Challenges

- the phonemes (speech categories) are not known in advance
- the words are not given
- infer phonemes and words from experience (unsupervised/reinforcement learning)
- build associations between sounds (words) and images (objects) by interacting with the environment.
- ...more next time!