Automatic speech recognition

Matthias Wächter, Jan Rademacher

Ameland, August 2007

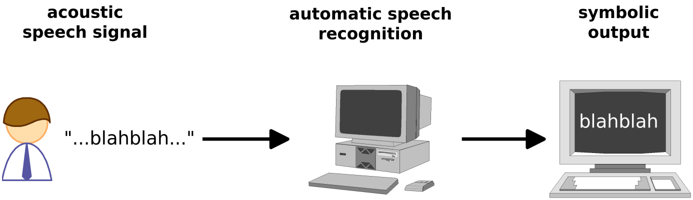
Overview

1. Introduction to automatic speech recognition
2. Feature extraction
3. Vocal tract length invariant features
4. Hidden Markov models
5. Excursion: connection to diffusion processes
6. Summary & concluding remarks

What’s next?

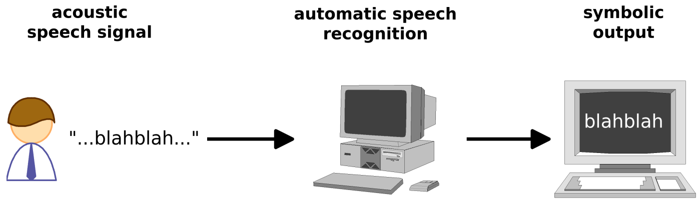
1. Introduction to automatic speech recognition
2. Feature extraction
3. Vocal tract length invariant features
4. Hidden Markov models
5. Excursion: connection to diffusion processes
6. Summary & concluding remarks

Goal of ASR



Goal of automatic speech recognition

*Symbolic representation of an utterance,
which is only available as acoustic signal.*



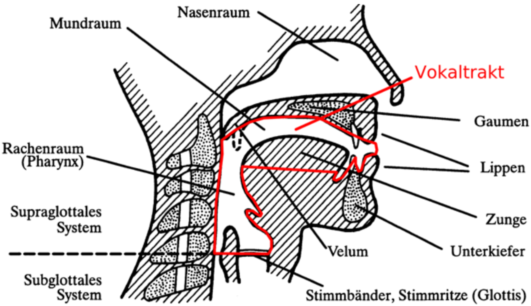
Goal of automatic speech recognition

Symbolic representation of an utterance, which is only available as acoustic signal.

Scopes of application:

- dictation,
- translation,
- input- or control functions,
- . . .

How does a word arise?

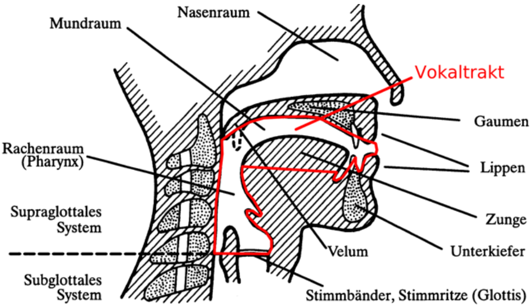


Creation of acoustic speech signal in two steps:

- Stimulus - airflow generates ocillations or noise
- Signal shaping - shaping of the stimuli by the vocal tract

The vocal tract and its length have a major influence on the formation of sounds.

How does a word arise?



Creation of acoustic speech signal in two steps:

- Stimulus - airflow generates ocillations or noise
- Signal shaping - shaping of the stimuli by the vocal tract

The vocal tract and its length have a major influence on the formation of sounds.

Properties of speech signals

Speech signals

- can be divided into temporary segments, e.g.,
  - words,
  - syllables,
  - phonemes – "smallest distinguishable units of a language"
- are bandpass signals (mainly 200-6000 Hz)
- contain – besides the message – information on
  - noise: environmental noise, . . .
  - way of articulation: emotions, cooperativeness, . . .
  - habitual characteristics of a speaker: dialect, non-native language, . . .
  - individual characteristics of a speaker: anatomy of the vocal tract → age, sex, . . .

Properties of speech signals

Speech signals

- can be divided into temporary segments, e.g.,
  - words,
  - syllables,
  - phonemes –
    "smallest distinguishable units of a language"
- are bandpass signals (mainly 200-6000 Hz)
- contain – besides the message – information on
  - noise: environmental noise, . . .
  - way of articulation: emotions, cooperativeness, . . .
  - habitual characteristics of a speaker:
    dialect, non-native language, . . .
  - individual characteristics of a speaker:
    anatomy of the vocal tract → age, sex, . . .

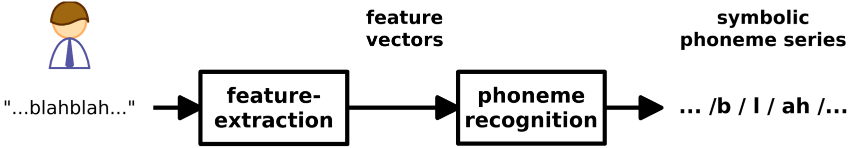
Properties of speech signals

Speech signals

- can be divided into temporary segments, e.g.,
  - words,
  - syllables,
  - phonemes – "smallest distinguishable units of a language"
- are bandpass signals (mainly 200-6000 Hz)
- contain – besides the message – information on
  - noise: environmental noise, . . .
  - way of articulation: emotions, cooperativeness, . . .
  - habitual characteristics of a speaker: dialect, non-native language, . . .
  - individual characteristics of a speaker: anatomy of the vocal tract → age, sex, . . .

Basic scheme of ASR

ASR is normally based on a phoneme recognition.



The phoneme recognition can be divided in two major parts:

- **feature extraction**
  extraction of specific features out of the speech signal
- **phoneme recognition**
  feature based recognition of corresponding phonemes

What's next?

1. Introduction to automatic speech recognition
2. Feature extraction
3. Vocal tract length invariant features
4. Hidden Markov models
5. Excursion: connection to diffusion processes
6. Summary & concluding remarks

Problem: Discrete time representation of a speech signal is not suitable for phoneme recognition.

Feature extraction:

Transformation of speech signals in a more suitable representation w.r.t. phoneme recognition.

Aims:

- reduction of the amount of data
- preservation of phoneme discriminative properties
- robustness against variabilities

Standard feature set for phoneme recognition:

Mel-Frequency Cepstral Coefficients (MFCC)

which are related to the human perception model.

Problem: Discrete time representation of a speech signal is not suitable for phoneme recognition.

Feature extraction:

Transformation of speech signals in a more suitable representation w.r.t. phoneme recognition.

Aims:

- reduction of the amount of data
- preservation of phoneme discriminative properties
- robustness against variabilities

Standard feature set for phoneme recognition:

Mel-Frequency Cepstral Coefficients (MFCC)

which are related to the human perception model.

Feature extraction

Problem: Discrete time representation of a speech signal is not suitable for phoneme recognition.

Feature extraction:

Transformation of speech signals in a more suitable representation w.r.t. phoneme recognition.

Aims:

- reduction of the amount of data
- preservation of phoneme discriminative properties
- robustness against variabilities

Standard feature set for phoneme recognition:

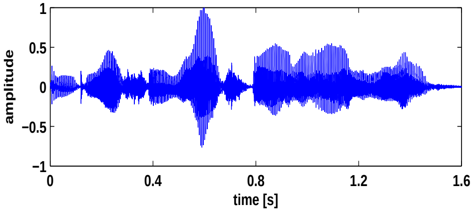
Mel-Frequency Cepstral Coefficients (MFCC)

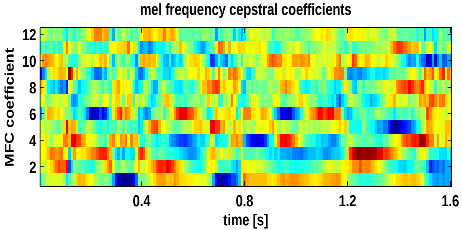
which are related to the human perception model.

Mel Frequency Cepstral Coefficients

Realization:

1. Cut signal in time frames of 10-30ms, overlap ≈ 50%
2. Calculate Hann-windowed spectrum per frame
3. Pool frequencies w.r.t. psychoacoustically motivated MEL-scale
4. Take log of magnitudes
5. Decorrelate each frame by discrete cosine transform



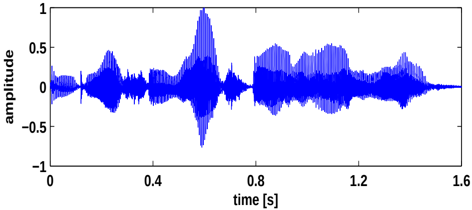


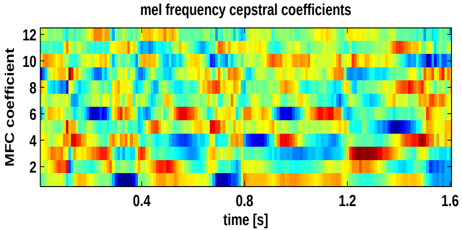
Result: One feature vector per time frame.

Mel Frequency Cepstral Coefficients

Realization:

1. Cut signal in time frames of 10-30ms, overlap ≈ 50%
2. Calculate Hann-windowed spectrum per frame
3. Pool frequencies w.r.t. psychoacoustically motivated MEL-scale
4. Take log of magnitudes
5. Decorrelate each frame by discrete cosine transform

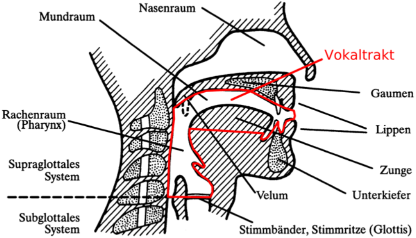




Result: One feature vector per time frame.

Vocal tract length variation

Important individual speaker properties (like pitch and sex) are directly connected to the vocal tract length.

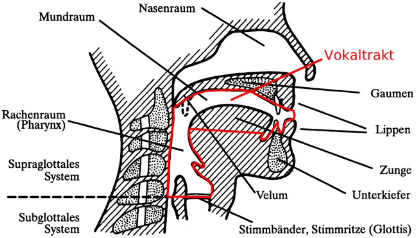


Problem for e.g. MFCC feature vectors:

Variation of vocal tract length leads to warping of the MEL magnitudes. As a consequence, the same utterance of different speakers results in different features!

Vocal tract length variation

Important individual speaker properties (like pitch and sex) are directly connected to the vocal tract length.



Problem for e.g. MFCC feature vectors:

Variation of vocal tract length leads to warping of the MEL magnitudes. As a consequence, the same utterance of different speakers results in different features!

Tackling VTL variations

Usual approach: Vocal tract length normalization (VTLN)

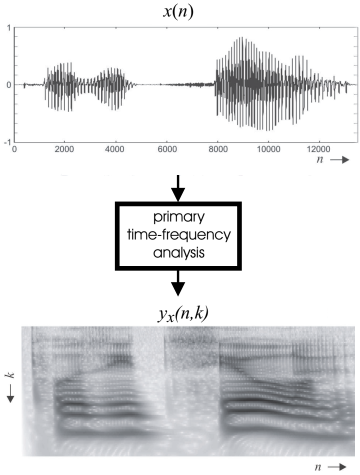
- Assume linear frequency scaling (warping) of short time spectra
- Estimate warping factor α according to highest recognition rate of a subsequent HMM recognizer
- Disadvantage: high computational load

Alternative approach: Vocal tract length invariant (VTLI) features

- Apply translation invariant transformation to short time spectra: use autocorrelation sequence

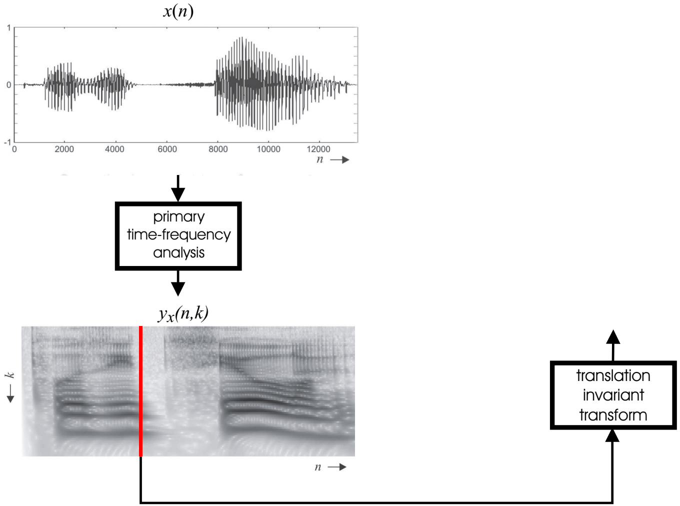
Vocal tract length invariant features

Example:



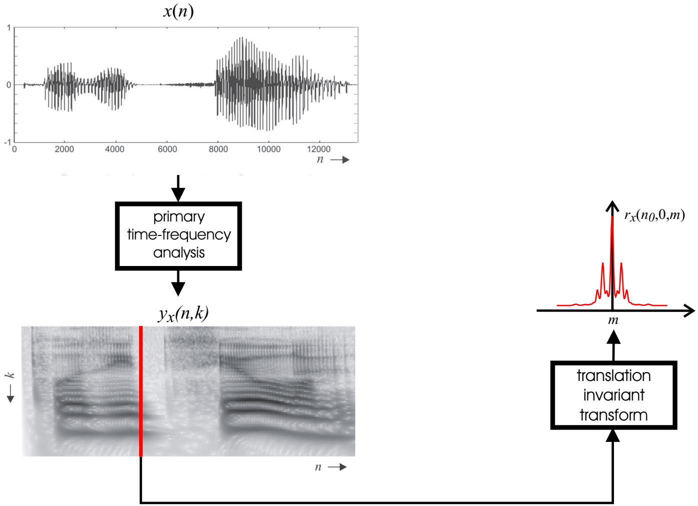
Vocal tract length invariant features

Example:



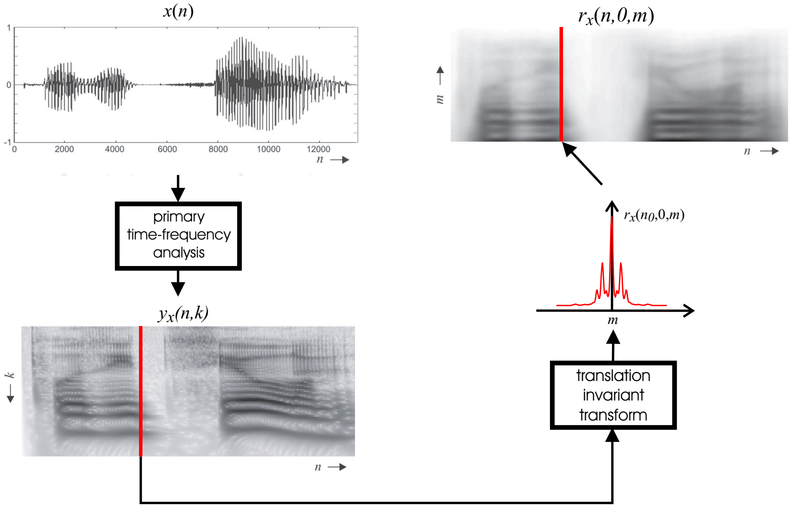
Vocal tract length invariant features

Example:



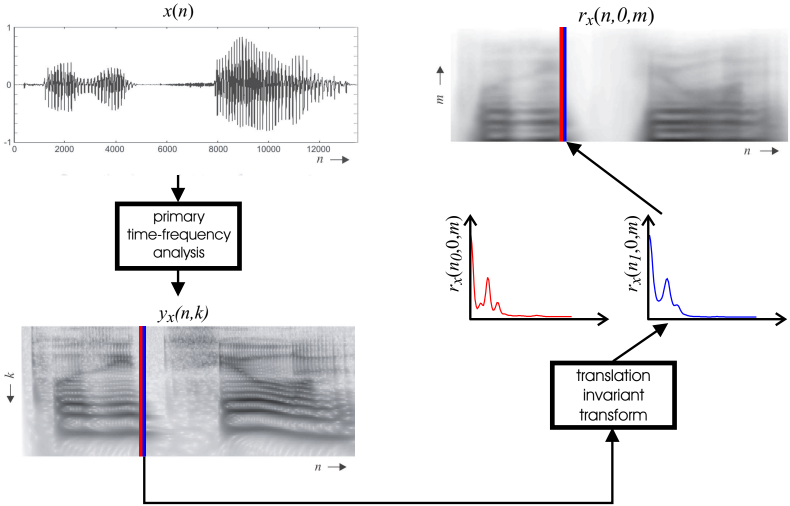
Vocal tract length invariant features

Example:



Vocal tract length invariant features

Example:



Invariances of VTLI features

VTLI features provide

- wanted invariance against frequency warping.
- additional (unwanted) invariances against a great class of operations.
- Example:
  - reversed sequence leads to identical autocorrelation

$$\tilde{y}(k) = y(K-k) \quad \Rightarrow \quad r_{\tilde{y}\tilde{y}}(m) = r_{yy}(m)$$

  - in general:
    Inversion of any zero of the z-transformed

$$Y(z) = \sum_k y(k) z^{-k}$$

    has no impact on either the absolute value $|Y(e^{j\omega})|$ or the autocorrelation $r_{yy}(m)$.

Unwanted invariances may reduce discriminative properties.

VTLI features provide

- wanted invariance against frequency warping.
- additional (unwanted) invariances against a great class of operations.
- Example:
  - reversed sequence leads to identical autocorrelation

$$\tilde{y}(k) = y(K-k) \quad \Rightarrow \quad r_{\tilde{y}\tilde{y}}(m) = r_{yy}(m)$$

  - in general:
    Inversion of any zero of the z-transformed

$$Y(z) = \sum_k y(k) z^{-k}$$

    has no impact on either the absolute value $|Y(e^{j\omega})|$ or the autocorrelation $r_{yy}(m)$.

Unwanted invariances may reduce discriminative properties.

Invariances of VTLI features

VTLI features provide

- wanted invariance against frequency warping.
- additional (unwanted) invariances against a great class of operations.
- Example:
  - reversed sequence leads to identical autocorrelation

$$\tilde{y}(k) = y(K-k) \quad \Rightarrow \quad r_{\tilde{y}\tilde{y}}(m) = r_{yy}(m)$$

  - in general:
    Inversion of any zero of the z-transformed

$$Y(z) = \sum_k y(k) z^{-k}$$

    has no impact on either the absolute value $|Y(e^{j\omega})|$ or the autocorrelation $r_{yy}(m)$.

Unwanted invariances may reduce discriminative properties.

Invariances of VTLI features

VTLI features provide

- wanted invariance against frequency warping.
- additional (unwanted) invariances against a great class of operations.
- Example:
  - reversed sequence leads to identical autocorrelation

$$\tilde{y}(k) = y(K - k) \quad \Rightarrow \quad r_{\tilde{y}\tilde{y}}(m) = r_{yy}(m)$$

  - in general:
    Inversion of any zero of the z-transformed

$$Y(z) = \sum_k y(k) z^{-k}$$

    has no impact on either the absolute value $|Y(e^{j\omega})|$ or the autocorrelation $r_{yy}(m)$.

Unwanted invariances may reduce discriminative properties.

Enhanced VTLI features

Extension of $y(k)$ to the complex plane:

$$u_x(k) = y_x(k) \cdot \exp\left(j\left(\frac{y_x(k)}{\sqrt{\sum_k |y_x(k)|^2}}\right)^{\kappa} \cdot \frac{\pi}{4}\right)$$

Extension to complex plane reduces unwanted invariances.

Experiments:

- VTLI featureset composed of
  - Magnitude and phase of $r_{uu}(m)$ or $r_{yy}(m)$
  - Different correlation terms of $y(k)$ and $\log(y(k))$
  - Classical MFCCs
  - Gammtone features $\log(y(k))$
- Reduction of feature set dimension via LDA.

VTLI featureset gives improved results compared to MFCCs for non-matching training and test conditions.

Enhanced VTLI features

Extension of $y(k)$ to the complex plane:

$$u_x(k) = y_x(k) \cdot \exp\left(j\left(\frac{y_x(k)}{\sqrt{\sum_k |y_x(k)|^2}}\right)^{\kappa} \cdot \frac{\pi}{4}\right)$$

Extension to complex plane reduces unwanted invariances.

Experiments:

- VTLI featureset composed of
  - Magnitude and phase of $r_{uu}(m)$ or $r_{yy}(m)$
  - Different correlation terms of $y(k)$ and $\log(y(k))$
  - Classical MFCCs
  - Gammtone features $\log(y(k))$
- Reduction of feature set dimension via LDA.

VTLI featureset gives improved results compared to MFCCs for non-matching training and test conditions.

Enhanced VTLI features

Extension of $y(k)$ to the complex plane:

$$u_x(k) = y_x(k) \cdot \exp\left(j \left(\frac{y_x(k)}{\sqrt{\sum_k |y_x(k)|^2}} \right)^{\kappa} \cdot \frac{\pi}{4} \right)$$

Extension to complex plane reduces unwanted invariances.

Experiments:

- VTLI featureset composed of
  - Magnitude and phase of $r_{uu}(m)$ or $r_{yy}(m)$
  - Different correlation terms of $y(k)$ and $\log(y(k))$
  - Classical MFCCs
  - Gammtone features $\log(y(k))$
- Reduction of feature set dimension via LDA.

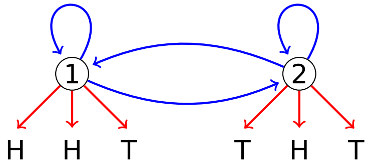
VTLI featureset gives improved results compared to MFCCs for non-matching training and test conditions.

What’s next?

1. Introduction to automatic speech recognition
2. Feature extraction
3. Vocal tract length invariant features
4. Hidden Markov models
5. Excursion: connection to diffusion processes
6. Summary & concluding remarks

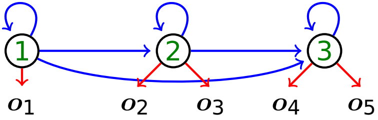
Example: coin toss behind a curtain
with two different (unfair) coins,
observation: H H T T H T H H T T H

Model: coins → “states” (hidden by the curtain)
H H T T H T H H T T H → “observation sequence”



Hidden-Markov-models (HMM)

HMM: Model of a system generating an observation sequence $O = \{o_1, \ldots, o_T\}$.



HMM has different states $q = 1, \ldots, N$ with transition probabilities $A = \{a_{ij}\}$.

States have emission probabilities $b = \{b_j(k)\}$ and start probabilities $\pi = \{\pi_j\}$.

"Hidden": state sequence $q = \{q_1, \ldots, q_T\}$ is a free parameter.

"Markov": next step depends only on present state.

Notation: Hidden Markov model $\lambda = (A, b, \pi)$

Task: Search λ_{max}, maximizing $P(O|\lambda) = \sum_{\{q\}} P(O, q|\lambda)$ (production probability) for given O:

$\lambda_{max} = \arg\max_{\lambda} \{P(O|\lambda)\}$

Method: Expectation Maximization (EM) algorithm

- universal process for parameter estimation in the case of missing data.
- missing data:
  state sequence resp. state probabilities

Iterative EM-algorithm

1. Estimate initial values $\lambda = (\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{\pi})$.
2. Calculate the state probabilities for O and λ (E-step), $P(O|\lambda)$ comes for free.
3. Calculate improved model $\overline{\lambda} = (\overline{\boldsymbol{A}}, \overline{\boldsymbol{b}}, \overline{\boldsymbol{\pi}})$ based on state probabilities (M-step).
4. Go to 2.

Result: Optimal adaptation of HMM λ to training data.

Iterative EM-algorithm

1. Estimate initial values $\lambda = (A, b, \pi)$.
2. Calculate the state probabilities for O and λ (E-step), $P(O|\lambda)$ comes for free.
3. Calculate improved model $\overline{\lambda} = (\overline{A}, \overline{b}, \overline{\pi})$ based on state probabilities (M-step).
4. Go to 2.

Result: Optimal adaptation of HMM λ to training data.

Iterative EM-algorithm

1. Estimate initial values $\lambda = (A, b, \pi)$.
2. Calculate the state probabilities for O and λ (E-step), $P(O|\lambda)$ comes for free.
3. Calculate improved model $\overline{\lambda} = (\overline{A}, \overline{b}, \overline{\pi})$ based on state probabilities (M-step).
4. Go to 2.

Result: Optimal adaptation of HMM λ to training data.

Iterative EM-algorithm

1. Estimate initial values $\lambda = (\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{\pi})$.
2. Calculate the state probabilities for O and λ (E-step), $P(O|\lambda)$ comes for free.
3. Calculate improved model $\overline{\lambda} = (\overline{\boldsymbol{A}}, \overline{\boldsymbol{b}}, \overline{\boldsymbol{\pi}})$ based on state probabilities (M-step).
4. Go to 2.

Result: Optimal adaptation of HMM λ to training data.

Iterative EM-algorithm

1. Estimate initial values $\lambda = (A, b, \pi)$.
2. Calculate the state probabilities for O and λ (E-step), $P(O|\lambda)$ comes for free.
3. Calculate improved model $\overline{\lambda} = (\overline{A}, \overline{b}, \overline{\pi})$ based on state probabilities (M-step).
4. Go to 2.

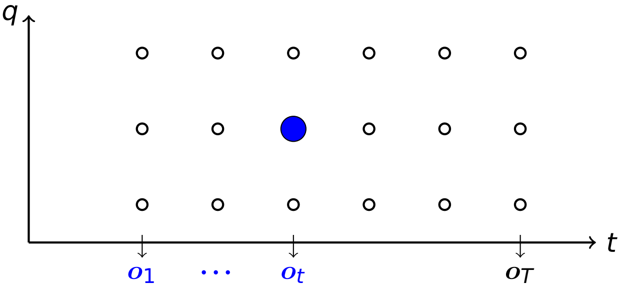
Result: Optimal adaptation of HMM λ to training data.

CARL VON OSSIETZKY universität OLDENBURG

Algorithm example: forward probability

First algorithm for caclulation of state probabilities:

$$
\begin{aligned}
\text{Def.:}\quad \alpha_t(i) &= P(o_1, \ldots, o_t, q_t = i|\lambda) \\
\text{Recursion:}\quad \alpha_1(i) &= \pi_i\, b_i(o_1) \\
\alpha_{t+1}(j) &= \left\{ \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right\} b_j(o_{t+1})
\end{aligned}
$$



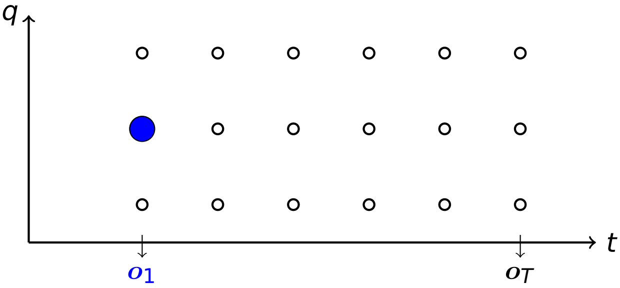
Algorithm example: forward probability

First algorithm for caclulation of state probabilities:

$$\text{Def.:} \quad \alpha_t(i) = P(o_1, \ldots, o_t, q_t = i | \lambda)$$

$$\text{Recursion:} \quad \alpha_1(i) = \pi_i \, b_i(o_1)$$

$$\alpha_{t+1}(j) = \left\{ \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right\} b_j(o_{t+1})$$



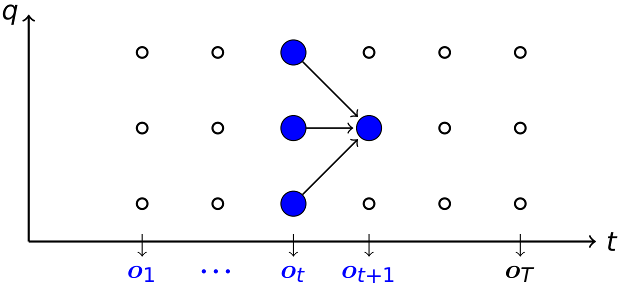
Algorithm example: forward probability

First algorithm for caclulation of state probabilities:

$$\text{Def.:}\quad \alpha_t(i) = P(o_1, \ldots, o_t, q_t = i|\lambda)$$

$$\text{Recursion:}\quad \alpha_1(i) = \pi_i\, b_i(o_1)$$

$$\alpha_{t+1}(j) = \left\{ \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right\} b_j(o_{t+1})$$



Effective calculation of $P(O|\lambda)$ by Forward algorithm:

$$P(O|\lambda) = \sum_{i=1}^{N} P(O, q_T = i|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

$\Rightarrow$ for $N = 5, T = 100$ about $10^{70} \times$ less operations than

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

Production probability

Effective calculation of $P(O|\lambda)$ by Forward algorithm:

$$P(O|\lambda) \quad = \quad \sum_{i=1}^{N} P(O, q_T = i|\lambda) \quad = \quad \sum_{i=1}^{N} \alpha_T(i)$$

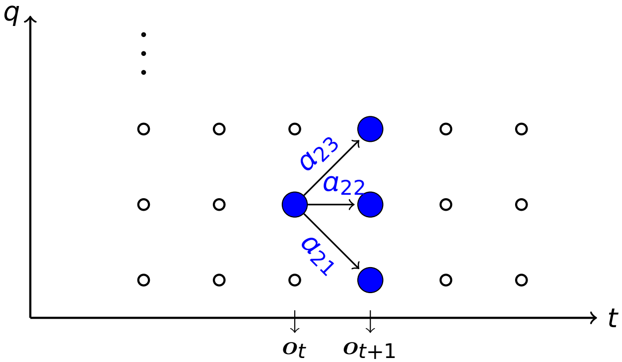
$\Rightarrow$ for $N = 5, T = 100$ about $10^{70} \times$ less operations than

$$P(O|\lambda) \quad = \quad \sum_{\{q\}} P(O, q|\lambda)$$

Connection to diffusion processes

after Benabdallah, Löser & Radons, submitted to PRE

- Let states $\Leftrightarrow$ measurement values (typically several thousand)



- $A = \{a_{ij}\}$ contains information of $D^{(k)}(q)$
- Problem: EM algorithm not practicable for many states

Recognition: Viterbi decoding

Now given:

- Observation sequence $O = \{o_1, \ldots, o_T\}$ and
- Modells $\lambda_1, \ldots, \lambda_L$.

Searching model λ^* with

$$P(O|\lambda^*) = \max_l \{\hat{P}(O|\lambda_l)\}$$

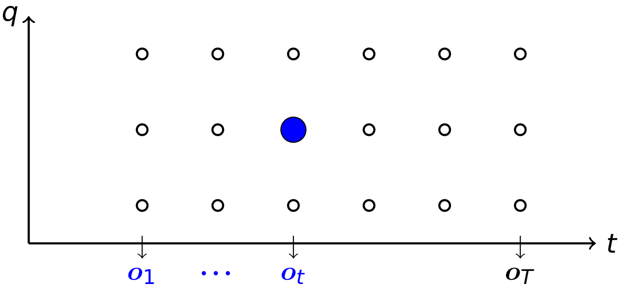
where $$\hat{P}(O|\lambda_l) = \max_{\{q\}} \{P(O, q|\lambda_l)\}$$

Viterbi decoding algorithm

$$\text{Def.:}\quad \phi_i(t) = \hat{P}(o_1, \ldots, o_t, q_t = i|\lambda)$$

$$\text{recursion:}\quad \phi_i(1) = \pi_i b_i(o_1)$$

$$\phi_j(t+1) = \left[\max_i \{\phi_i(t) a_{ij}\}\right] b_j(o_{t+1})$$

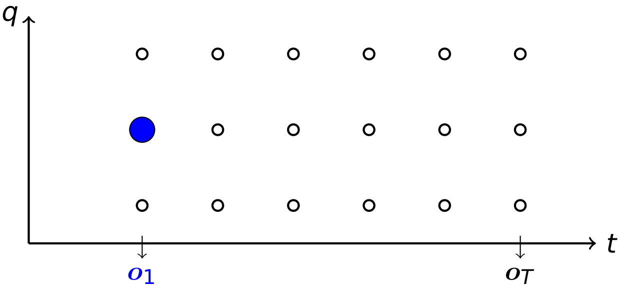


Viterbi decoding algorithm

$$\text{Def.:} \quad \phi_i(t) = \hat{P}(o_1, \ldots, o_t, q_t = i|\lambda)$$

$$\text{recursion:} \quad \phi_i(1) = \pi_i \, b_i(o_1)$$

$$\phi_j(t+1) = \left[\max_i \{\phi_i(t) a_{ij}\}\right] b_j(o_{t+1})$$

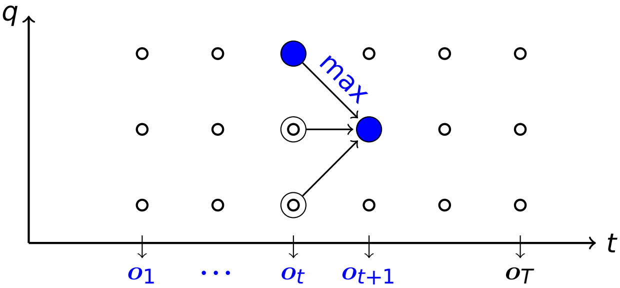


Viterbi decoding algorithm

$$\text{Def.:} \quad \phi_i(t) = \hat{P}(o_1, \ldots, o_t, q_t = i | \lambda)$$

$$\text{recursion:} \quad \phi_i(1) = \pi_i \, b_i(o_1)$$

$$\phi_j(t+1) = \left[\max_i \{ \phi_i(t) a_{ij} \} \right] b_j(o_{t+1})$$



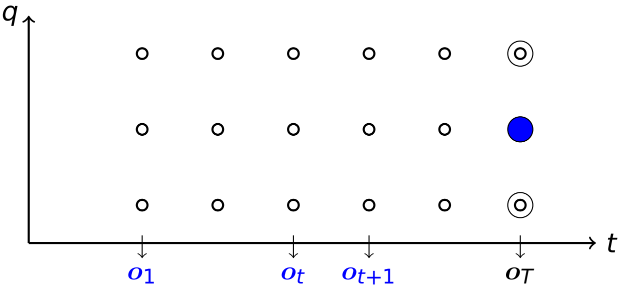
Viterbi decoding algorithm

$$\text{Def.:} \quad \phi_i(t) = \hat{P}(o_1, \ldots, o_t, q_t = i|\lambda)$$

$$\text{recursion:} \quad \phi_i(1) = \pi_i b_i(o_1)$$

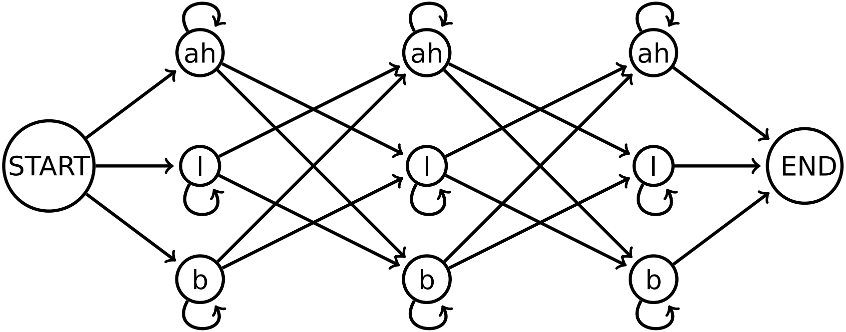
$$\phi_j(t+1) = \left[\max_i \{\phi_i(t) a_{ij}\}\right] b_j(o_{t+1})$$

$$\hat{P}(O|\lambda) = \max_j \{\phi_j(T)\}$$



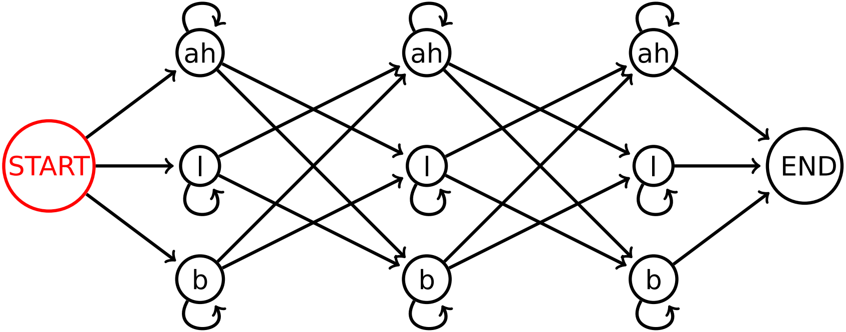
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a “one state” HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b/l/ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



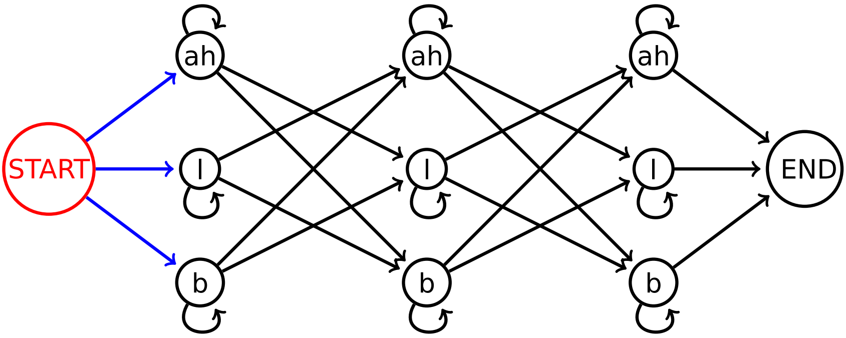
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a "one state" HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b*/*l*/*ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



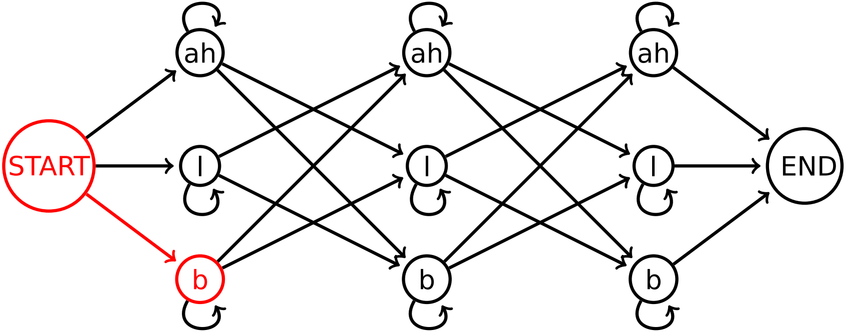
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a "one state" HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b/l/ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



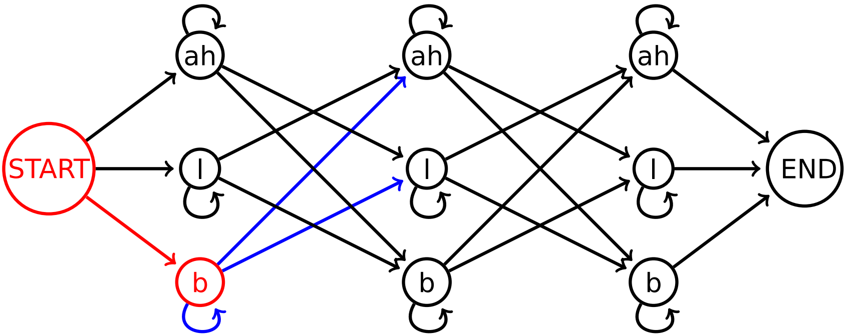
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a “one state” HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b/l/ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



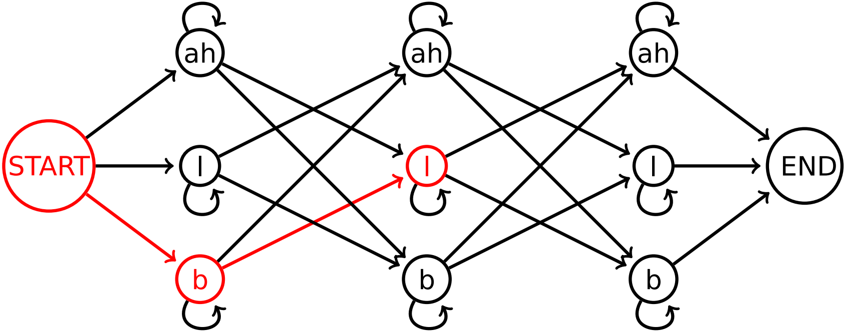
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a “one state” HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b/l/ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



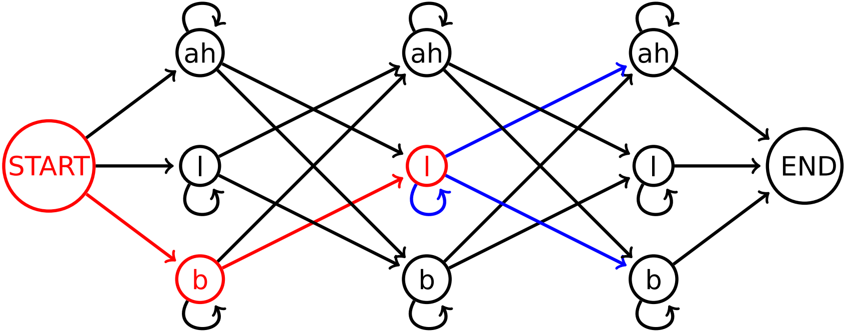
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a “one state” HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b*/*l*/*ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



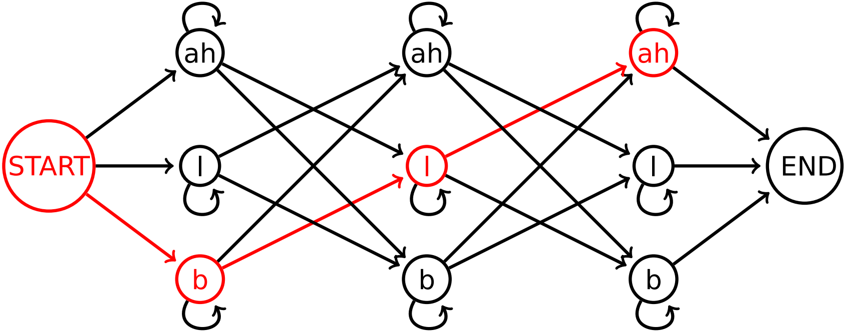
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a "one state" HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b/l/ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



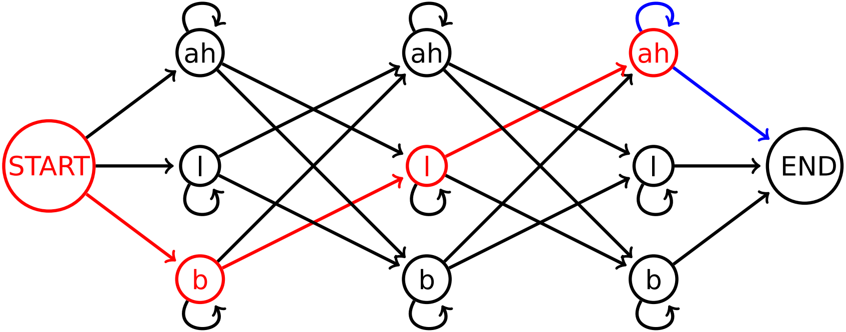
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a "one state" HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b/l/ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



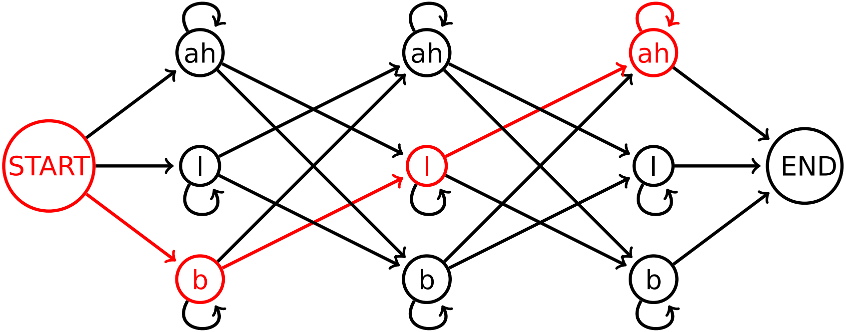
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a “one state” HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b*/*l*/*ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



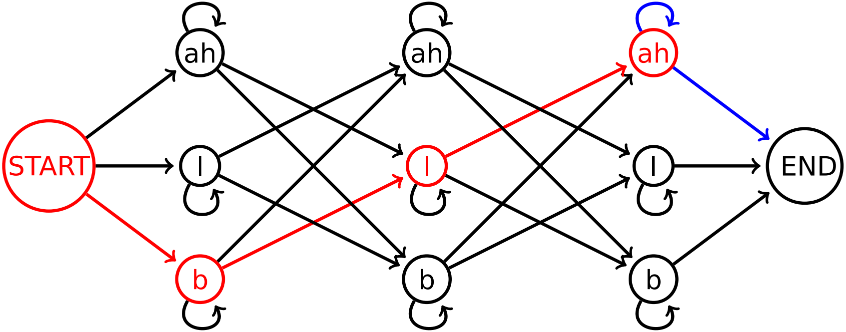
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a “one state” HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b*/*l*/*ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



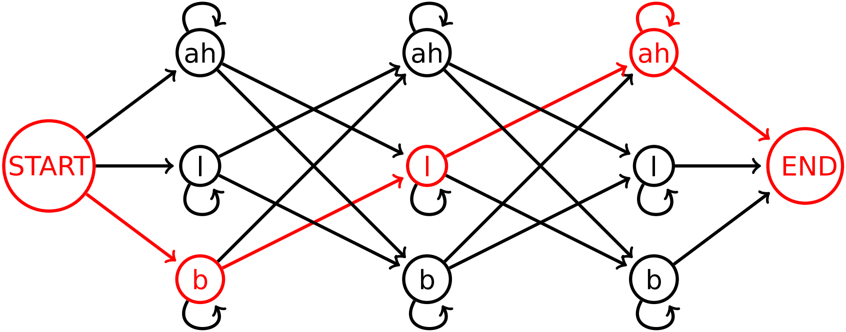
Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a “one state” HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b*/*l*/*ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$

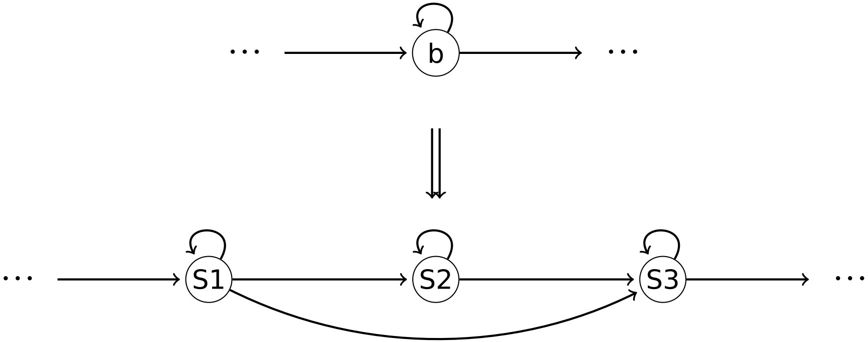


Example: simple phoneme recognition

- Observable phoneme sequences consist of 3 different phonemes
- Each phoneme is represented by a “one state” HMM
- Each phoneme can be one ore more time frames long
- Alphabet consists of three phonemes (*b/l/ah*)
- Given: observation sequence $O = \{/b/l/ah/ah/\}$



In reality, ”one state” HMMs are not sufficient:



In general, a phoneme is represented by a three state left-to-right HMM.

What's next?

1. Introduction to automatic speech recognition
2. Feature extraction
3. Vocal tract length invariant features
4. Hidden Markov models
5. Excursion: connection to diffusion processes
6. Summary & concluding remarks

Summary on HMM

Hidden-Markov-models:

- universal approach for pattern recognition

Avantages:

- flexible, adaptable to many problems
- efficient training and test algorithms available

Disadvantages / limitations:

- model design and initialization requires expert know-how

Concluding remarks

- We saw an overwiew of some common concepts
- Many advanced techniques exist:
  - Artificial neural networks (ANN) for feature extraction
    - Pure ANN
    - “Tandem” features: MFCC processed by ANN
  - Inclusion of context information
    - Grammars for natural speech or special tasks
    - Context in feature extraction