Hidden Markov Models

CI/CI(CS) UE, SS 2015

Christian Knoll
Signal Processing and Speech Communication Laboratory
Graz University of Technology

June 23, 2015

Content

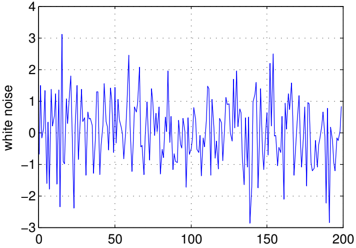
- Motivation/Goals
- Modeling of sequential data
- Definition of (Hidden) Markov Models
- Parameters of HMMs
- Calculations / Illustrations / Algorithms
- Examples

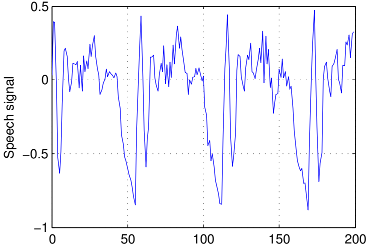
Motivation

- List of samples, calculation of joint likelihood:

$$p(\mathbf{X}|\mathbf{\Theta}) = \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{\Theta})$$

$\rightarrow$ Requires independence of samples $\mathbf{x}_n$ (i.i.d.)!





- Strong requirement that might not be met!
- How to model dependencies?

Example – Weather (1)

- Observations: $\mathbf{Q} = \{$, , , , $\}$ (Weather on a specific day)
- Data modeled as i.i.d.
- Probability of rain on one day depends *only* on relative frequency of Rain in $\mathbf{Q}$
- Correlations/Trends over multiple days are ignored

Example – Weather (2)

- Observations: $\mathbf{Q} = \{ , , \}$
- Weather on day n as $q_n \in \{ , , \}$
- We are interested in the following *conditional* probability

$$\mathrm{P}(q_n | q_{n-1}, q_{n-2}, \ldots, q_1)$$

- How probable is rain on day n, given the observations above:

$$\mathrm{P}(q_4 = \ | q_3 = \ , q_2 = \ , q_1 = \ )$$

- Possibility: Estimate based on prior observations of sequence $\{ , , , \}$
- Problem: Complexity grows exponentially

Simplification: Markov-Assumption

- The following assumption simplifies the problem:

$$\mathrm{P}(q_n | q_{n-1}, q_{n-2}, \ldots, q_1) = \mathrm{P}(q_n | q_{n-1})$$

- Weather on day n only depends on weather on day $n-1$

→ Markov-Assumption of first-order

- Model using this assumption is called *Markov-Model*, an output sequence of it *Markov-Chain*
- Probability of a sequence $\{q_n, \ldots, q_2, q_1\}$

$$\mathrm{P}(q_n, \ldots, q_1) = \prod_{i=1}^{n} \mathrm{P}(q_i | q_{i-1})$$

Simplification: Markov-Assumption (2)

- Now, only $3 \cdot 3 = 9$ statistics necessary
- Arrange in table:

Wetter today	Weather tomorrow		
			
	0.8	0.05	0.15
	0.2	0.6	0.2
	0.2	0.3	0.5

- For sure *any* weather tomorrow
- Table holds *transition probabilities* $\mathrm{P}(q_i | q_{i-1})$
- Representation as automata → Blackboard

Computation of Probabilities

- Reminder: Product rule: $\mathrm{P}(X, Y) = \mathrm{P}(Y|X)\mathrm{P}(X)$
- Tomorrow ☀, given yesterday ☔ and today ☁

$$\mathrm{P}(q_3 = \text{☀} | q_2 = \text{☁}, q_1 = \text{☔})$$

- Tomorrow ☀, day after tomorrow ☔, given today ☀

$$\mathrm{P}(q_3 = \text{☔}, q_2 = \text{☀} | q_1 = \text{☀})$$

- Day after tomorrow ☔, given today ☁

$$\mathrm{P}(q_3 = \text{☔} | q_1 = \text{☁})$$

→ Blackboard

Hidden Markov Models (HMMs)

- States (weather) of system no longer directly observable
- Observations that depend on the states (emission prob.)
- E.g. prison guard carries umbrella with him or not

Wetter	P. for umbrella 
	0.1
	0.8
	0.3

Hidden Markov Models (HMMs)

- States (weather) of system no longer directly observable
- Observations that depend on the states (emission prob.)
- E.g. prison guard carries umbrella with him or not

Wetter	P. for umbrella 
	0.1
	0.8
	0.3

- hidden states $q_i \in \{$, , $\}$ and observations $x_i \in \{$, $\}$

Hidden Markov Models (HMMs)

- States (weather) of system no longer directly observable
- Observations that depend on the states (emission prob.)
- E.g. prison guard carries umbrella with him or not

Wetter q_i	P. for umbrella  $P(x_i \| q_i)$
	0.1
	0.8
	0.3

- hidden states $q_i \in \{$, , $\}$ and observations $x_i \in \{$, $\}$
  $\rightarrow$ We want $P(q_i|x_i) \rightarrow$ Bayes

$$P(q_i|x_i) = \frac{P(x_i|q_i)P(q_i)}{P(x_i)},$$

HMMs – Definition and Parameters

- An HMM consists of
  - Set of N_s states $S = \{s_1, \dots, s_{N_s}\}$
- and Parameters $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$
  - $\pi_i = \mathrm{P}(q_1 = s_i)$, a-priori P. of states (first state in sequence)
  - $a_{i,j} = \mathrm{P}(q_{n+1} = s_j | q_n = s_i)$, Transition probabilities

HMMs – Definition and Parameters

- An HMM consists of
  - Set of N_s states $S = \{s_1, \dots, s_{N_s}\}$
- and Parameters $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$
  - $\pi_i = \mathrm{P}(q_1 = s_i)$, a-priori P. of states (first state in sequence)
  - $a_{i,j} = \mathrm{P}(q_{n+1} = s_j | q_n = s_i)$, Transition probabilities
  - Entries in $\mathbf{B}$ are the *emission probabilities*

HMMs – Definition and Parameters

- An HMM consists of
  - Set of N_s states $S = \{s_1, \ldots, s_{N_s}\}$
- and Parameters $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$
  - $\pi_i = \mathrm{P}(q_1 = s_i)$, a-priori P. of states (first state in sequence)
  - $a_{i,j} = \mathrm{P}(q_{n+1} = s_j | q_n = s_i)$, Transition probabilities
  - Entries in $\mathbf{B}$ are the *emission probabilities*
    - *discrete* observations $x_n \in \{v_1, \ldots, v_K\}$, $b_{i,k} = \mathrm{P}(x_n = v_k | q_n = s_i)$, are the P. for observing $x_n = v_k$ in state s_i

HMMs – Definition and Parameters

- An HMM consists of
  - Set of N_s states $S = \{s_1, \ldots, s_{N_s}\}$
- and Parameters $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$
  - $\pi_i = \mathrm{P}(q_1 = s_i)$, a-priori P. of states (first state in sequence)
  - $a_{i,j} = \mathrm{P}(q_{n+1} = s_j | q_n = s_i)$, Transition probabilities
  - Entries in $\mathbf{B}$ are the *emission probabilities*
    - *discrete* observations $x_n \in \{v_1, \ldots, v_K\}$, $b_{i,k} = \mathrm{P}(x_n = v_k | q_n = s_i)$, are the P. for observing $x_n = v_k$ in state s_i
    - *continuous* Observations, e.g. $x_n \in \mathbb{R}^D$, $b_i(x_n) = p(x_n | q_n = s_i)$, (PDFs of observations)

HMMs – Definition and Parameters – Transition Matrix **A**

q_n	q_{n+1}		
			
	0.8	0.05	0.15
	0.2	0.6	0.2
	0.2	0.3	0.5

$$a_{i,j} = \mathrm{P}(q_{n+1} = s_j | q_n = s_i)$$

$$\Rightarrow \mathbf{A} = \begin{bmatrix} 0.8 & 0.05 & 0.15 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$$

HMMs – Definition and Parameters – Emission Matrix **B**

q_i	$\mathrm{P}(x_i =$  $\mid q_i)$
	0.1
	0.8
	0.3

$x_n \in \{v_1 =$ $, v_2 =$ $\}$

$$b_{i,k} = \mathrm{P}(x_n = v_k \mid q_n = s_i)$$

$$\Rightarrow \mathbf{B} = \begin{bmatrix} 0.1 & 0.9 \\ 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$$

HMMs – Useful formulas

- Most is already known, here just in HMM notation
- Probability of state sequence $Q = \{q_1, \ldots, q_n\}$

$$\mathrm{P}(Q|\boldsymbol{\Theta}) = \pi_{q_1} \prod_{n=1}^{N} a_{q_n, q_{n+1}}$$

- Prob. of observation sequence given state sequence

$$\mathrm{P}(X|Q, \boldsymbol{\Theta}) = \prod_{n=1}^{N} b_{q_n, x_n}$$

- Joint Likelihood of X and Q (product rule)

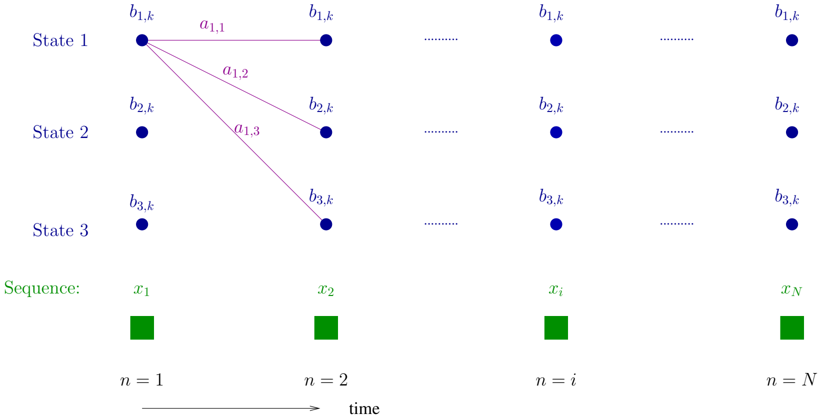
$$\mathrm{P}(X, Q|\boldsymbol{\Theta}) = \mathrm{P}(X|Q, \boldsymbol{\Theta}) \cdot \mathrm{P}(Q|\boldsymbol{\Theta})$$

- Likelihood of observation sequence given an HMM

$$\mathrm{P}(X|\boldsymbol{\Theta}) = \sum_{\mathrm{all}\,Q} \mathrm{P}(X, Q|\boldsymbol{\Theta})$$

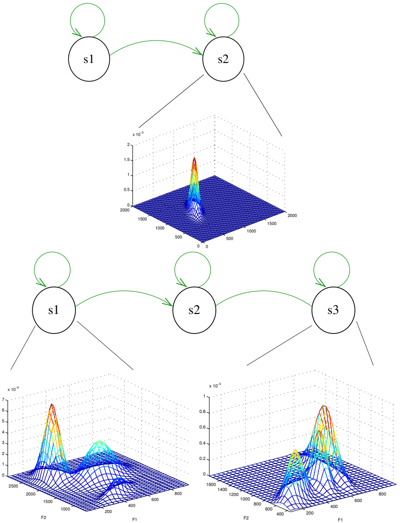
HMMs – Trellis Diagram

- Representation of HMM and parameters with time evolution



HMMs with continuous observations

- Observations continuous
- **B** might hold parameters of Gaussians
- Example here: Speech recognition
- Observations: Feature vectors
- State could be word element, observation feature vector



HMMs – Examples

$\rightarrow$ Matlab-examples

HMMs – 3 basic problems

1. Given an observation sequence X and model $\mathbf{\Theta}$, calculate $\mathrm{P}(X|\mathbf{\Theta})$, (P. that $\mathbf{\Theta}$ generated sequence)
   $\rightarrow$ e.g. for classification, forward-backward-algorithm
2. Given an observation sequence X and model $\mathbf{\Theta}$, estimate *optimal* state sequence Q
   - Find hidden states, Viterbi-algorithm
3. Given an observation sequence X (training-sequence), estimate model $\mathbf{\Theta}$, such that $\mathrm{P}(X|\mathbf{\Theta})$ is maximized
   $\rightarrow$ Learning problem, e.g. Baum-Welch-algorithm

$\rightarrow$ We cover 1) and 2)

Sequence classification with HMMs

- Given an observation sequence X and multiple HMMs $\boldsymbol{\Theta}_k$
- X could be extracted features of speech, HMMs model word elements
- Problem: Which of the HMMs has generated sequence?
- Calculate likelihood for X for every model
- We already know the equation

$$\mathrm{P}(X|\boldsymbol{\Theta}) = \sum_{\text{all}\,Q} \mathrm{P}(X, Q|\boldsymbol{\Theta})$$

- Sum over *all* state sequences that the model allows

$\rightarrow$ Matlab

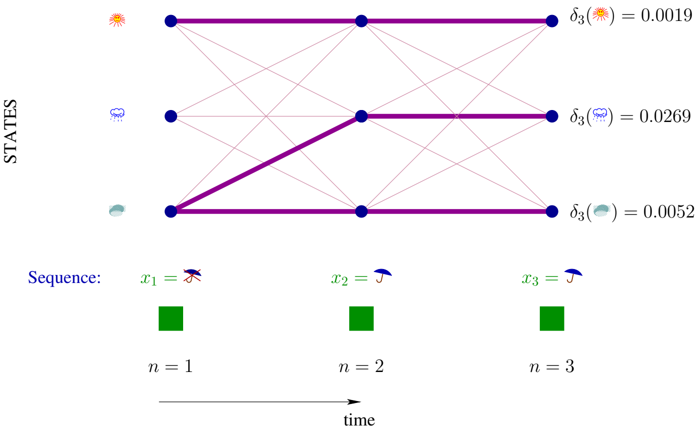
Optimal State Sequence

- Given an observation sequence X and HMM $\boldsymbol{\Theta}$
- Problem: What is the *optimal* state sequence?
- Q that explains X best
- That is Q, which maximizes $\mathrm{P}(Q|X, \boldsymbol{\Theta})$
- Sequence with maximum a-posteriori probability
- We can calculate $L(q_1, \ldots, q_n | x_1, \ldots, x_n) \rightarrow$ ML-path
- For *every* path: Complexity grows exponentially with n

$\rightarrow$ Viterbi-algorithm (GSM, WLAN, satellite communication, speech recognition, . . .)

Optimal state sequence – Weather example

- e.g. at $n = 3$: Every state possible, *one* path (per state) has largest L until here (*survivor*-path)



Viterbi Algorithm – General

$\rightarrow$ Blackboard

Viterbi Algorithm – Induction

- Assume $\delta_n(i)$ holds max. likelihood of path ending in $q_n = s_i$ at time n
- Likelihood in next time step is straightforward

$$\delta_{n+1}(j) = \max_i \{\delta_n(i) \cdot a_{i,j}\} \cdot b_{j,x_{n+1}}$$

- And the most probable predecessor is

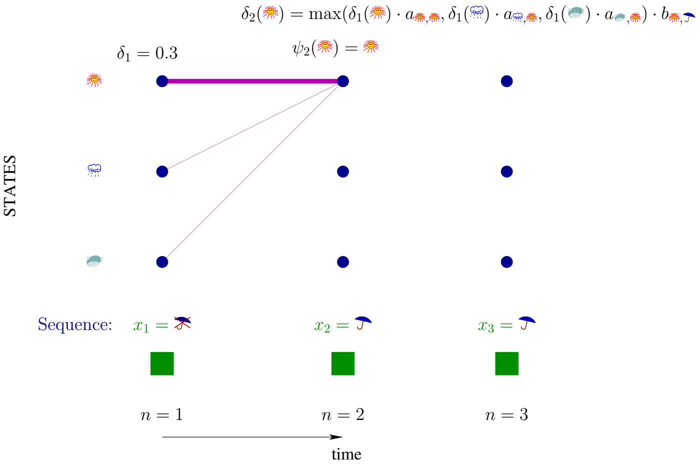
$$\psi_{n+1}(j) = \arg\max_i \{\delta_n(i) \cdot a_{i,j}\}$$

- If we also know $\delta_1(i)$, we are (almost) done
- For initialization, we take the a-priori-P. of the states

$$\begin{aligned} \delta_1(i) &= \pi_i \cdot b_{i,x_1} \\ \psi_1(i) &= 0 \end{aligned}$$

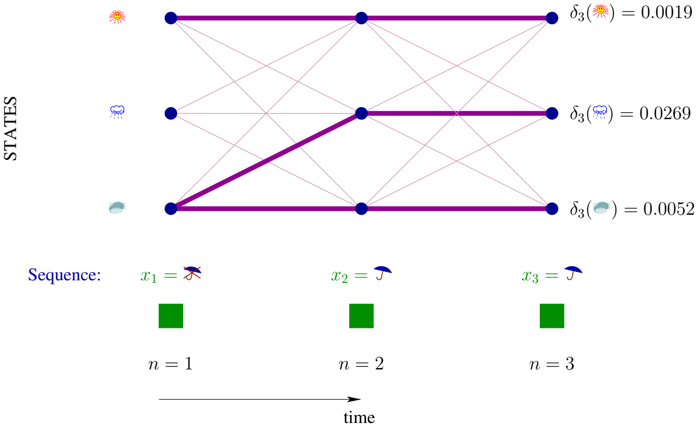
Viterbi Algorithm – Weather example (1)

- Most probable path to at $n = 2$:



Viterbi Algorithm – Weather example (2)

- Three most probable paths at $n = 3$:



Viterbi Algorithm – Termination and Backtracking

- Out of the $\delta_n(i)$ and $\psi_n(i)$ we want to recover the optimal state sequence $Q^* = \{q_1^*, \ldots, q_N^*\}$
- Last time step $n = N$:

$$\begin{aligned} p^*(X|\boldsymbol{\Theta}) &= \max_{1 \le i \le N_s} \delta_N(i), \quad \text{maximum likelihood} \\ q_N^* &= \arg\max_{1 \le i \le N_s} \delta_N(i), \quad \text{last state of ML path} \end{aligned}$$

- We have last state $\psi_n(i)$ prior ones can be traced back (*Backtracking*)

$$q_{n-1}^* = \psi_n(q_n^*)$$

Viterbi Algorithm – Weather example (3)

- Backtracking of optimal path