Hidden Markov Models

Lecture Notes

Speech Communication 2, SS 2004

Erhard Rank/Franz Pernkopf

Signal Processing and Speech Communication Laboratory

Graz University of Technology

Inffeldgasse 16c, A-8010 Graz, Austria

Tel.: +43 316 873 4436

E-Mail: rank@inw.tugraz.at

Word Recognition

Given:

- Word dictionary: $\mathcal{W} = \{W_1, \ldots, W_L\}$
- Time-series of features from unknown word: $X = \{x_1, \ldots, x_N\}$

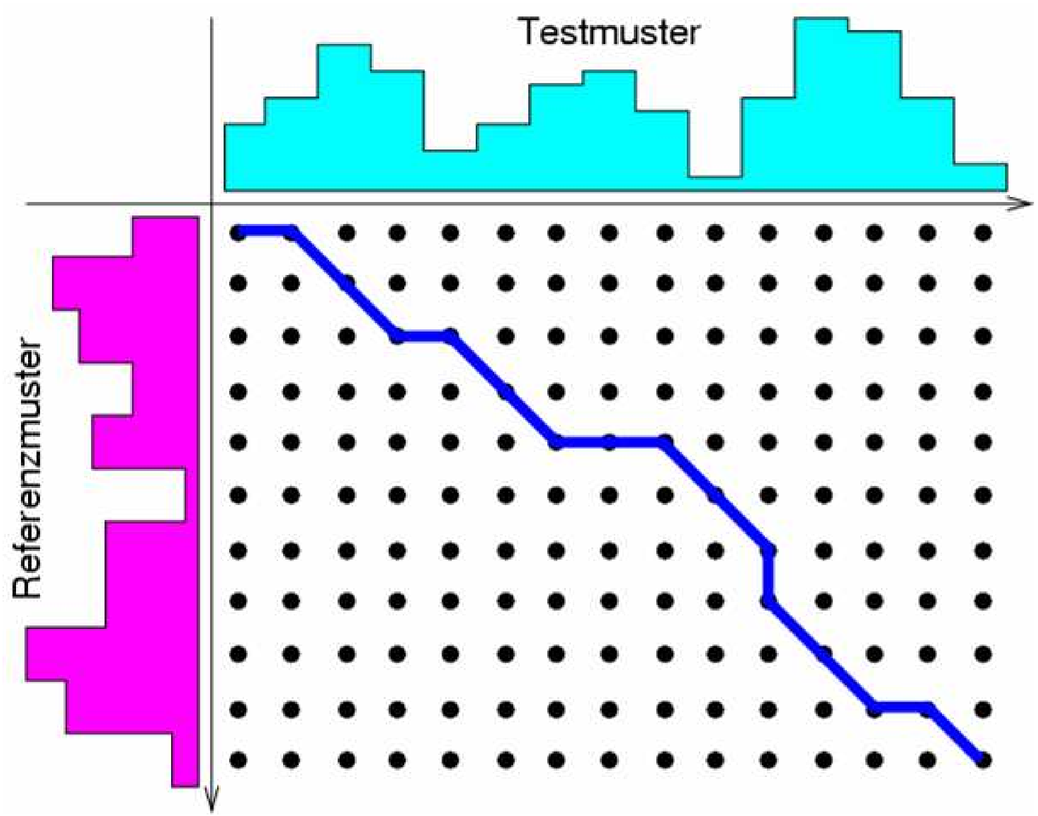
Wanted:

- Most probable spoken word: $W_{l^*} \in \mathcal{W}$

Target:

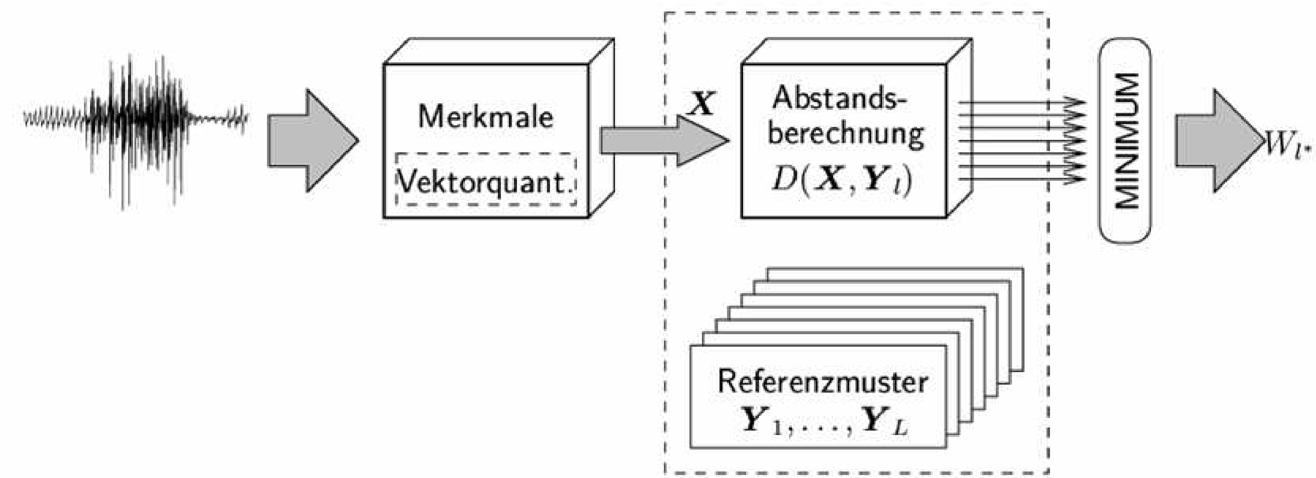
- Minimization of *word error rate (WER)*

Alignment of observed and reference pattern



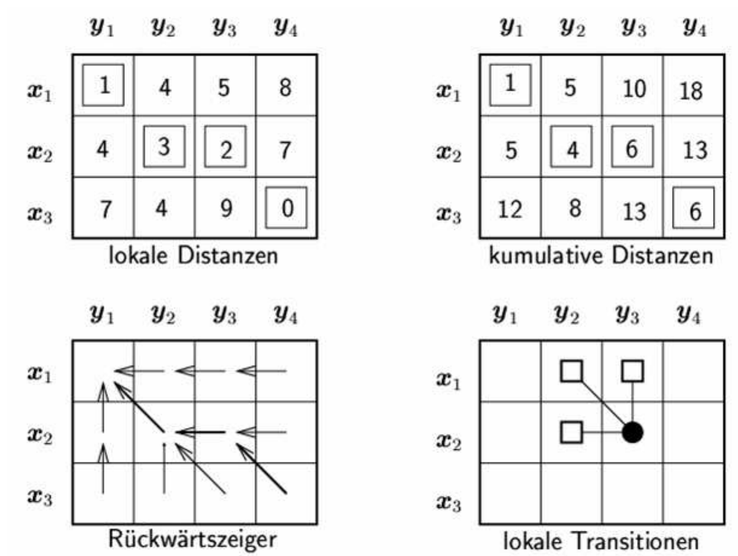
Word Recognition/Dynamic Time Warping

Pattern matching



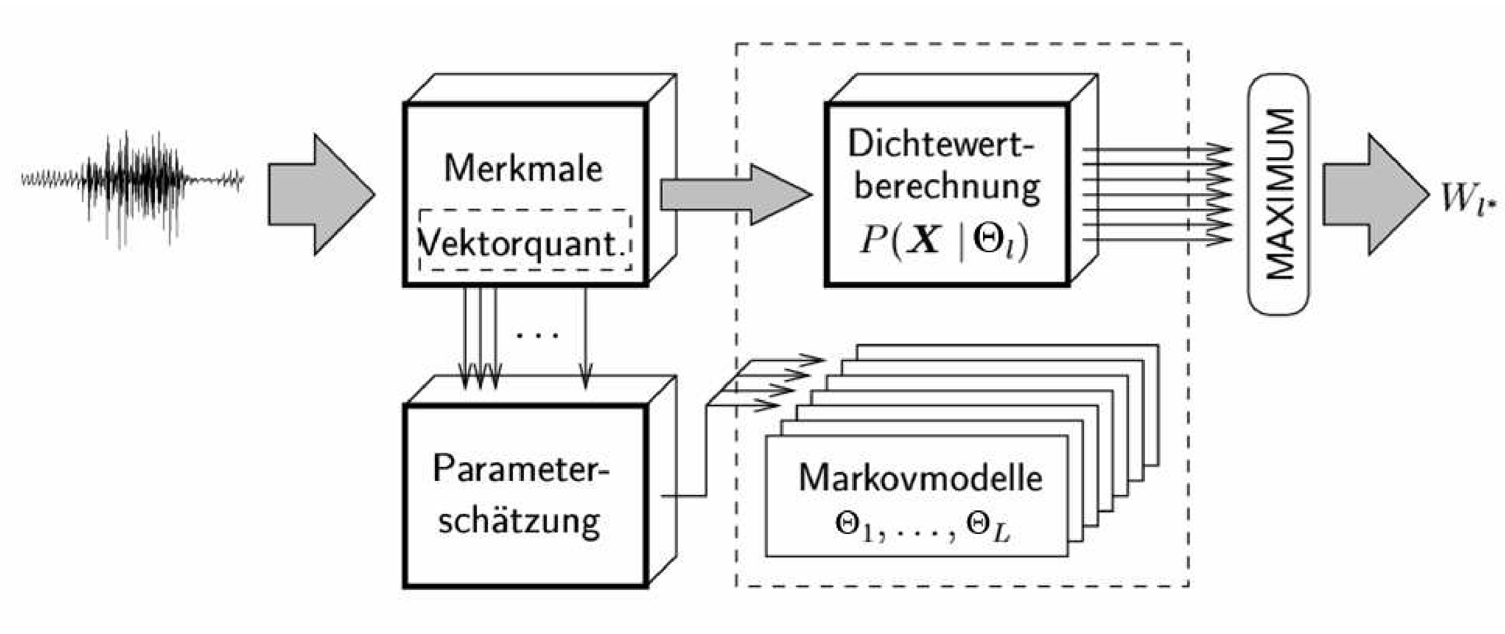
Dynamic time warping (DTW):

Dynamic programming, complexity: $\mathcal{O}(SN)$



Word Recognition/Markov Models

- Word recognition by maximizing the probability of Markov model $\mathbf{\Theta}_l$ of word W_l for the observed time-series of feature vectors X:



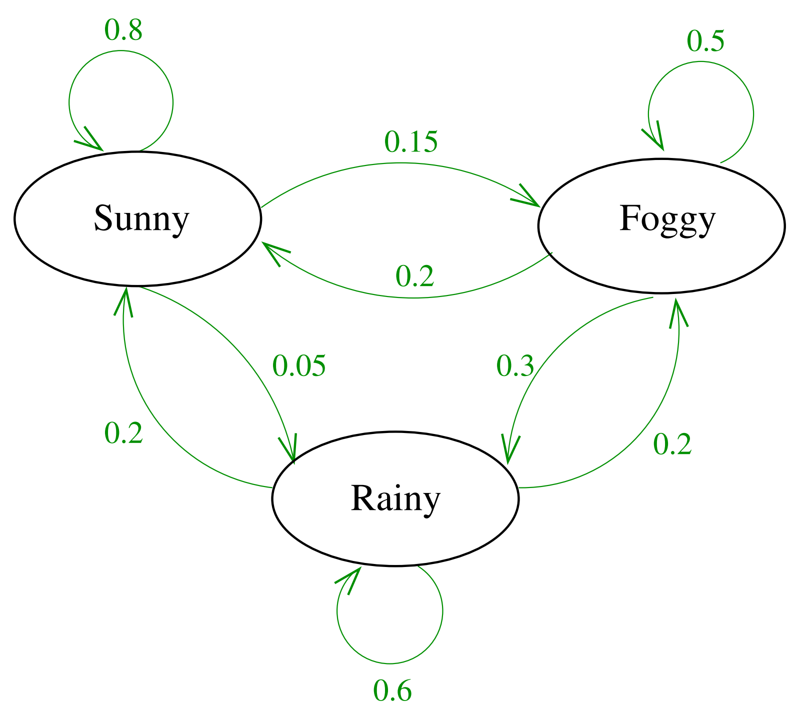
$$l^* = \operatorname*{argmax}_l P(\mathbf{\Theta}_l|X) = \operatorname*{argmax}_l \frac{P(X|\mathbf{\Theta}_l) \cdot P(\mathbf{\Theta}_l)}{P(X)}$$

Markov Model/Graz Weather

Transition probabilities:

Today's weather	Tomorrow's weather		
			
	0.8	0.05	0.15
	0.2	0.6	0.2
	0.2	0.3	0.5

State transition diagram:



Markov Model

A Markov Model is specified by

- The *set of states*
  $$S = \{s_1, s_2, \ldots, s_{N_s}\}.$$

and characterized by

- The *prior probabilities*
  $$\pi_i = P(q_1 = s_i)$$
  Probabilities of s_i being the first state of a state sequence. Collected in vector $\boldsymbol{\pi}$. (The prior probabilities are often assumed equi-probable, $\pi_i = 1/N_s$.)
- The *transition probabilities*
  $$a_{ij} = P(q_{n+1} = s_j | q_n = s_i)$$
  probability to go from state i to state j. Collected in matrix $\mathbf{A}$.

The Markov model produces

- A *state sequence*
  $$Q = \{q_1, \ldots, q_N\}, \quad q_n \in S$$
  over time $1 \leq n \leq N$.

Hidden Markov Model

Additionally, for a Hidden Markov model we have

- *Emission probabilities*:
  - for *continuous observations*, e.g., $x \in \mathbb{R}^D$:
    $$b_i(x) = p(x_n|q_n = s_i)$$
    pdfs of the observation x_n at time n, if the system is in state s_i. Collected as a vector of functions $\mathbf{B}(x)$. Often parametrized, e.g, by mixtures of Gaussians.
  - for *discrete observations*, $x \in \{v_1, \ldots, v_K\}$:
    $$b_{i,k} = P(x_n = v_k|q_n = s_i)$$
    Probabilities for the observation of $x_n = v_k$, if the system is in state s_i. Collected in matrix $\mathbf{B}$.

and we get

- Observation sequence:
  $$X = \{x_1, x_2, \ldots, x_N\}$$

HMM parameters (for fixed number of states N_s) thus are

$$\boldsymbol{\Theta} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$$

HMM/Graz Weather

The above weather model turns into a hidden Markov model, if we can not observe the weather directly. Suppose you were locked in a room for several days, and you can only observe if a person is carrying an umbrella ($v_1 =$ ) or not ($v_2 =$ ).
Example emission probabilities could be:

Weather	Probability of "umbrella"
Sunny	$b_{1,1} = 0.1$
Rainy	$b_{2,1} = 0.8$
Foggy	$b_{3,1} = 0.3$

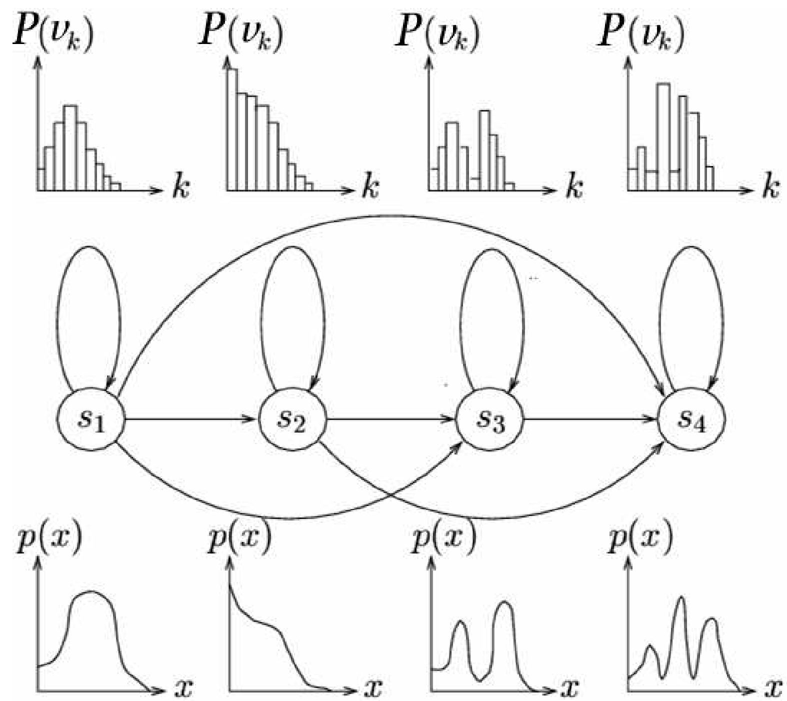
Since there are only two possible states for the *discrete observations*, the probabilities for "no umbrella" are $b_{i,2} = 1 - b_{i,1}$.

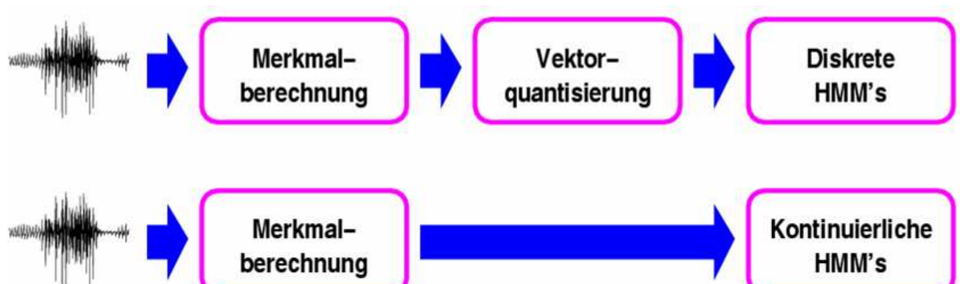
HMM/Discrete vs. Continuous Features

Discrete features/
emission probability:

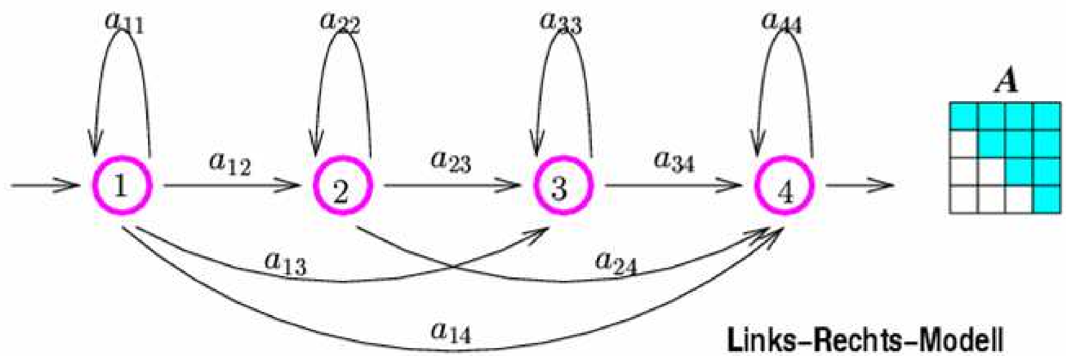
HMM:

Continuous features/
emission probability:

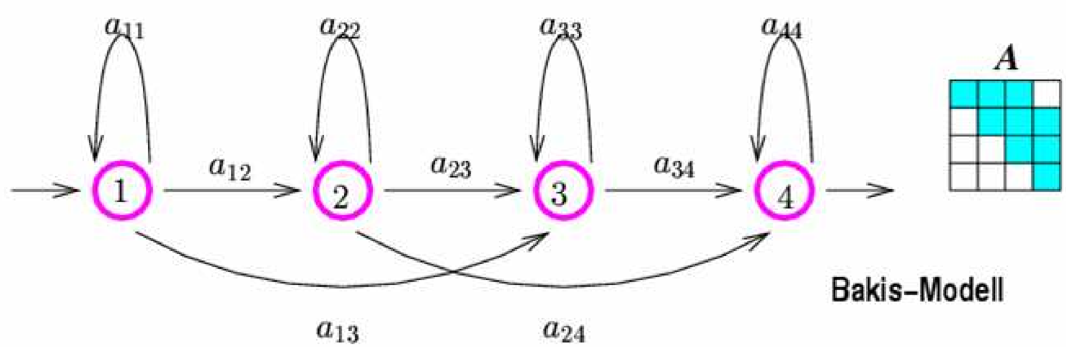




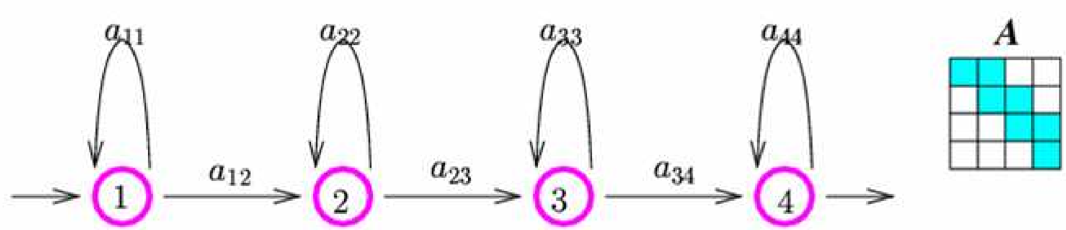
Markov Model/Left-to-Right Models



Links-Rechts-Modell

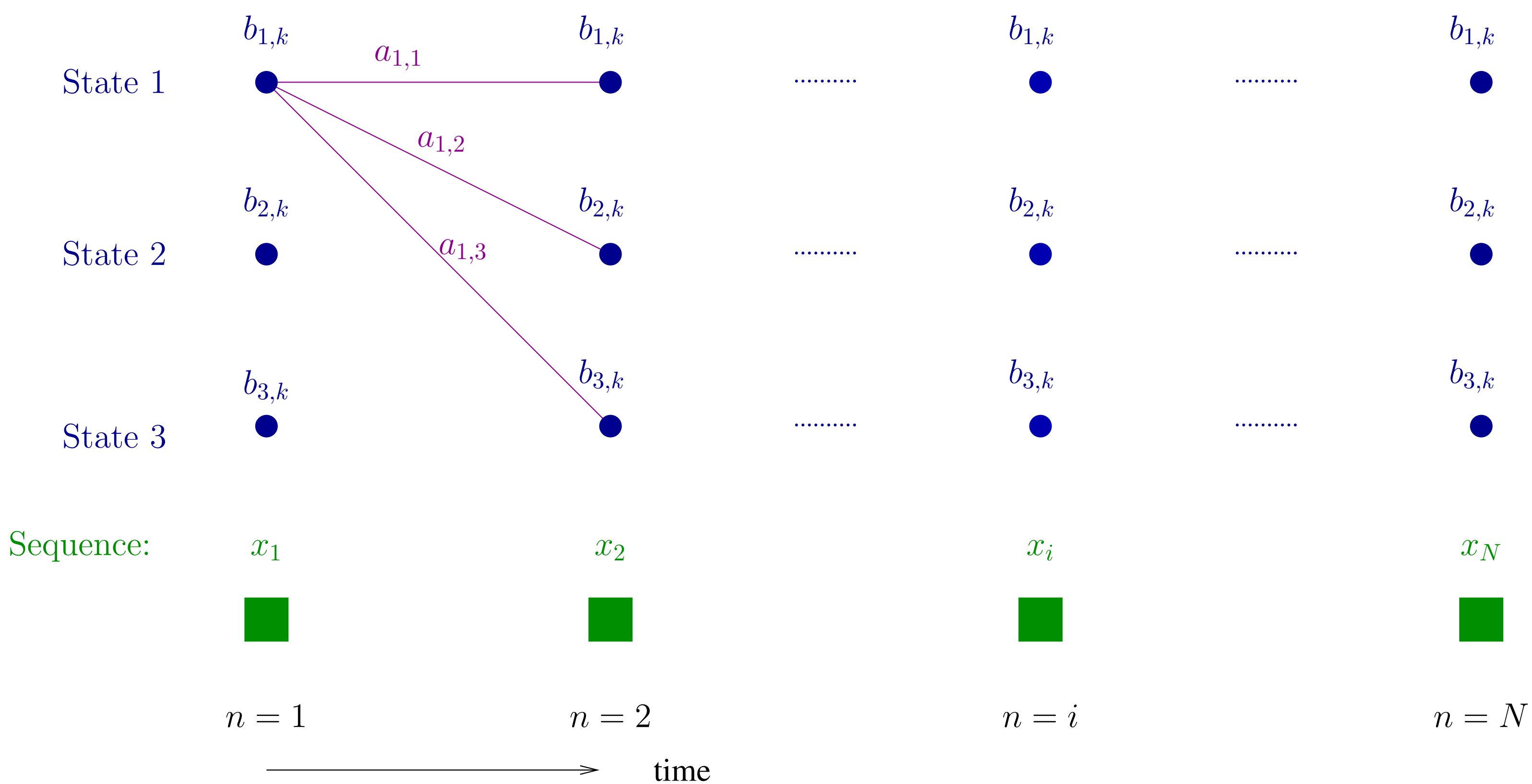


Bakis-Modell

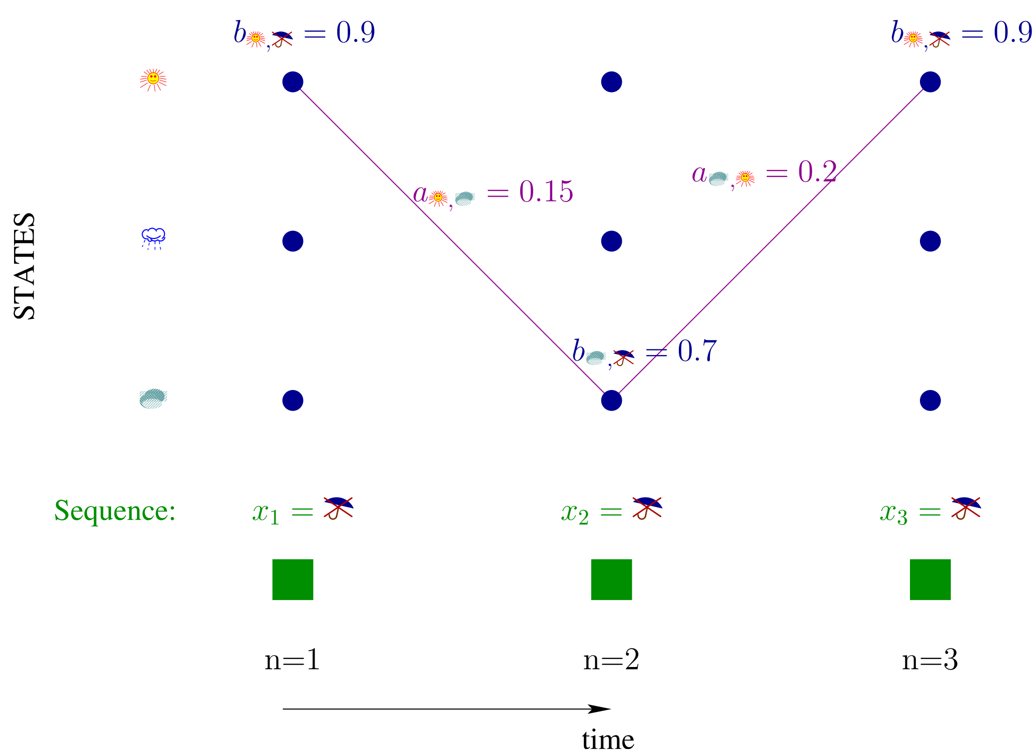


Lineares Modell

Trellis: Model description over time



HMM/Trellis example



Joint likelihood for observed sequence X and state sequence (path) Q:

$$P(X,Q|\boldsymbol{\Theta}) = \pi_{☀} \cdot b_{☀,\text{umbrella}} \cdot a_{☀,\text{cloud}} \cdot b_{\text{cloud},\text{umbrella}} \cdot a_{\text{cloud},☀} \cdot b_{☀,\text{umbrella}}$$

$$= 1/3 \cdot 0.9 \cdot 0.15 \cdot 0.7 \cdot 0.2 \cdot 0.9$$

HMM/Parameters

Parameters $\{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$ are probabilities:

- positive

$$\pi_i \geq 0, \quad a_{i,j} \geq 0, \quad b_{i,k} \geq 0 \text{ or } b_i(x) \geq 0$$

- normalization conditions

$$\sum_{i=1}^{N_s} \pi_i = 1, \quad \sum_{j=1}^{N_s} a_{i,j} = 1, \quad \sum_{k=1}^{K} b_{i,k} = 1 \text{ or } \int_{\mathbb{X}} b_i(x)\, dx = 1$$

Hidden Markov Models: 3 Problems

The "three basic problems" for HMMs:

- Given a HMM with parameters $\mathbf{\Theta} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, efficiently compute the *production probability* of an observation sequence X

$$P(X|\mathbf{\Theta}) = ? \tag{1}$$

- Given model $\mathbf{\Theta}$, what is the *hidden state sequence* Q that best explains an observation sequence X

$$Q^* = \operatorname*{argmax}_Q P(X, Q|\mathbf{\Theta}) = ? \tag{2}$$

- How do we *adjust the model parameters* to maximize $P(X|\mathbf{\Theta})$

$$\hat{\mathbf{\Theta}} = (\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\boldsymbol{\pi}}) = ?, \quad P(X|\hat{\mathbf{\Theta}}) = \max_{\mathbf{\Theta}} P(X|\mathbf{\Theta}) \tag{3}$$

Problem 1: Production probability

- Given: HMM parameters $\mathbf{\Theta}$
- Given: Observed sequence X (length N)
- Wanted: Probability $P(X|\mathbf{\Theta})$, for X being produced by $\mathbf{\Theta}$

Probability of a certain state sequence

$$P(Q|\mathbf{\Theta}) = P(q_1, \ldots, q_N|\mathbf{\Theta}) = \pi_{q_1} \cdot \prod_{n=2}^{N} a_{q_{n-1},q_n}$$

Emission probabilities for the state sequence

$$P(X|Q, \mathbf{\Theta}) = P(x_1, \ldots, x_N|q_1, \ldots, q_n, \mathbf{\Theta}) = \prod_{n=1}^{N} b_{q_n,x_n}$$

Joint probability of hidden state sequence and observation sequence

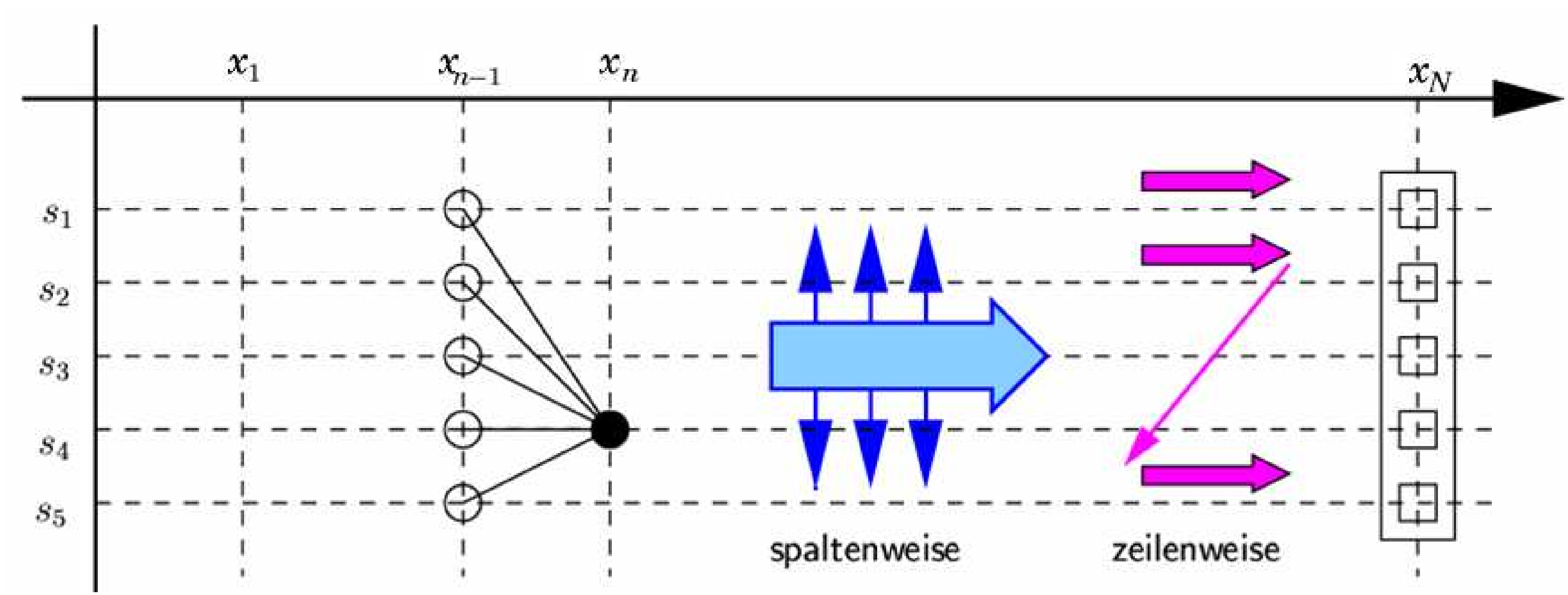
$$P(X, Q|\mathbf{\Theta}) = P(X|Q, \mathbf{\Theta}) \cdot P(Q|\mathbf{\Theta}) = \pi_{q_1} \cdot b_{q_1}(x_1) \cdot \prod_{n=2}^{N} a_{q_{n-1},q_n} \cdot b_{q_n,x_n}$$

Production probability

$$P(X|\mathbf{\Theta}) = \sum_{Q \in \mathcal{Q}^N} P(X,Q|\mathbf{\Theta}) = \sum_{Q \in \mathcal{Q}^N} \left(\pi_{q_1} \cdot b_{q_1}(x_1) \cdot \prod_{n=2}^{N} a_{q_{n-1},q_n} \cdot b_{q_n,x_n} \right)$$

Exponential complexity $\mathcal{O}(2N \cdot N_s^N)$

$\Rightarrow$ use recursive algorithm (complexity linear in $N \cdot N_s$):



HMM/Prod. Probability

Forward algorithm

Computation of *forward probabilities*

$$\alpha_n(j) = P(x_1, \ldots, x_n, q_n = s_j | \mathbf{\Theta})$$

- Initialization: for all $j = 1 \ldots N_s$

$$\alpha_1(j) = \pi_i \cdot b_{j,x_1}$$

- Recursion: for all $n > 1$ and all $j = 1 \ldots N_s$

$$\alpha_n(j) = \left(\sum_{i=1}^{N_s} \alpha_{n-1}(i) \cdot a_{i,j} \right) \cdot b_{j,x_n}$$

- Termination:

$$P(X|\mathbf{\Theta}) = \sum_{j=1}^{N_s} \alpha_N(j)$$

Backward algorithm

Computation of *backward probabilities*

$$\beta_n(i) = P(x_n + 1, \ldots, x_N | q_n = s_i, \mathbf{\Theta})$$

- Initialization: for all $i = 1 \ldots N_s$

$$\beta_N(i) = 1$$

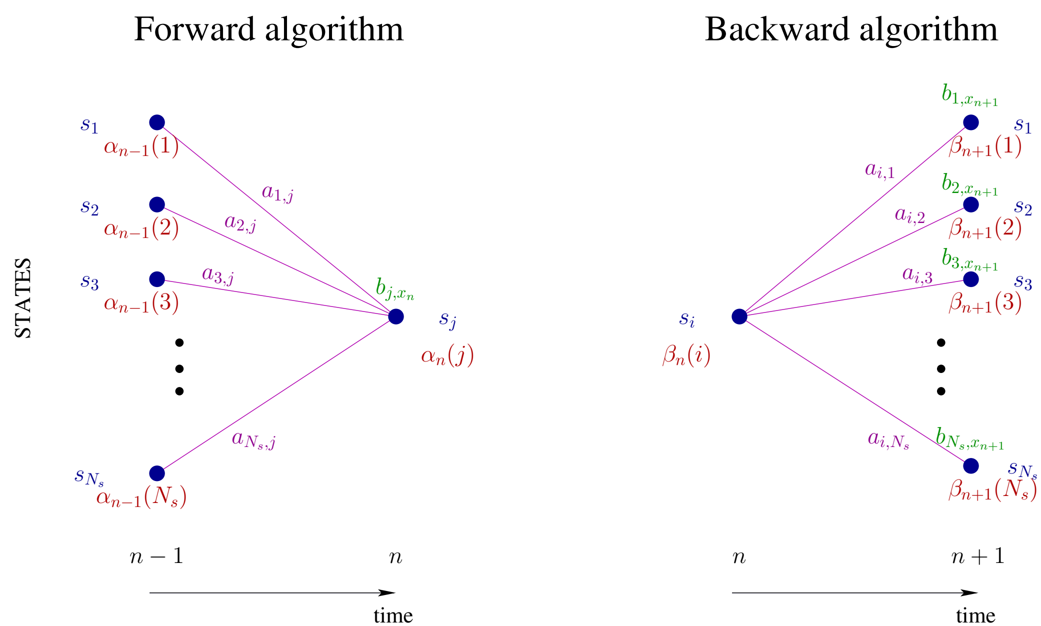
- Recursion: for all $n < N$ and all $i = 1 \ldots N_s$

$$\beta_n(i) = \sum_{j=1}^{N_s} a_{i,j} \cdot b_{j,x_{n+1}} \cdot \beta_{n+1}(j)$$

- Termination:

$$P(X|\mathbf{\Theta}) = \sum_{j=1}^{N_s} \pi_j \cdot b_{j,x_1} \cdot \beta_1(j)$$

HMM/Prod. Probability



- At each time n

$$\alpha_n(j) \cdot \beta_n(j) = P(X, q_n = s_j | \boldsymbol{\Theta})$$

  is the joint probability of the observation sequence X and all state sequences (paths) passing through state s_j at time n,

- and

$$P(X|\boldsymbol{\Theta}) = \sum_{j=1}^{N_s} \alpha_n(j) \cdot \beta_n(j)$$

HMM/State Sequence

Problem 2: Hidden state sequence

- Given: HMM parameters $\mathbf{\Theta}$
- Given: Observed sequence X (length N)
- Wanted: A posteriori most probable state sequence Q^*

$\Rightarrow$ Viterbi algorithm

- a posteriori probabilities

$$P(Q|X,\mathbf{\Theta}) = \frac{P(X,Q|\mathbf{\Theta})}{P(X|\mathbf{\Theta})}$$

- Q^* is the optimal state sequence if

$$P(X,Q^*|\mathbf{\Theta}) = \max_{Q\in\mathcal{Q}^N} P(X,Q|\mathbf{\Theta}) =: P^*(X|\mathbf{\Theta})$$

- Viterbi algorithm computes

$$\delta_n(j) = \max_{Q\in\mathcal{Q}^n} P(x_1,\ldots,x_n,q_1,\ldots,q_n|\mathbf{\Theta}) \quad \text{for} \quad q_n = s_j$$

Viterbi Algorithm

Computation of *optimal state sequence*

- Initialization: for all $j = 1 \dots N_s$

$$\delta_1(j) = \pi_j \cdot b_{j,x_1}, \quad \psi_1(j) = 0$$

- Recursion: for $n > 1$ and all $j = 1 \dots N_s$

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

$$\psi_n(j) = \operatorname*{argmax}_i(\delta_{n-1}(i) \cdot a_{i,j})$$

- Termination:

$$P^*(X|\boldsymbol{\Theta}) = \max_j(\delta_N(j)), \quad q_N^* = \operatorname*{argmax}_j(\delta_N(j))$$

- Backtracking of optimal state sequence:

$$q_n^* = \psi_{n+1}(q_{n+1}^*), \quad n = N-1, N-2, \dots, 1$$

HMM/Viterbi Algorithm

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_1(1) & \delta_2(1) & \delta_2(1) & \cdots & \delta_N(1) \\ \delta_1(2) & \delta_2(2) & \delta_2(2) & \cdots & \delta_N(2) \\ \delta_1(3) & \delta_2(3) & \delta_2(3) & \cdots & \delta_N(3) \\ \delta_1(4) & \delta_2(4) & \delta_2(4) & \cdots & \delta_N(4) \end{bmatrix} \qquad \boldsymbol{\psi} = \begin{bmatrix} ? & \leftarrow & \swarrow & \cdots & \swarrow \\ ? & \nwarrow & \leftarrow & \cdots & \leftarrow \\ ? & \swarrow & \nwarrow & \cdots & \swarrow \\ ? & \leftarrow & \nwarrow & \cdots & \nwarrow \end{bmatrix}$$

Example:

For our weather HMM $\boldsymbol{\Theta}$, find the most probable hidden weather sequence for the observation sequence $X = \{x_1 = \lnot\text{umbrella}, x_2 = \text{umbrella}, x_3 = \text{umbrella}\}$

1. Initialization ($n = 1$):

$$\delta_1(\text{sunny}) = \pi_{\text{sunny}} \cdot b_{\text{sunny},\lnot\text{umbrella}} = 1/3 \cdot 0.9 = 0.3$$

$$\psi_1(\text{sunny}) = 0$$

$$\delta_1(\text{rainy}) = \pi_{\text{rainy}} \cdot b_{\text{rainy},\lnot\text{umbrella}} = 1/3 \cdot 0.2 = 0.0667$$

$$\psi_1(\text{rainy}) = 0$$

$$\delta_1(\text{foggy}) = \pi_{\text{foggy}} \cdot b_{\text{foggy},\lnot\text{umbrella}} = 1/3 \cdot 0.7 = 0.233$$

$$\psi_1(\text{foggy}) = 0$$

HMM/Viterbi Algorithm

2. Recursion ($n = 2$):

   We calculate the likelihood of getting to state '☀' from all possible 3 predecessor states, and choose the most likely one to go on with:

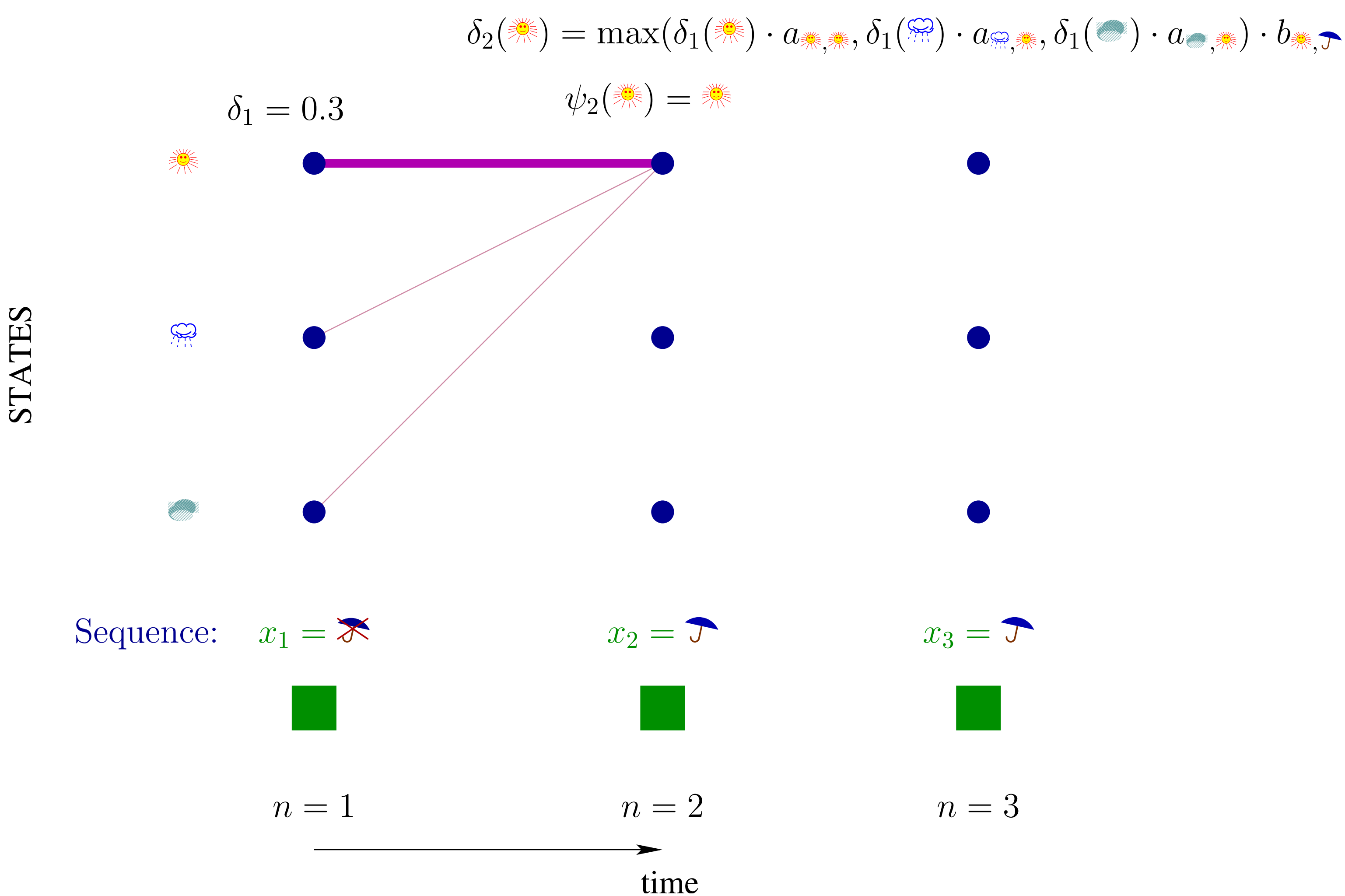
$$\delta_2(\text{☀}) = \max(\delta_1(\text{☀}) \cdot a_{\text{☀},\text{☀}}, \delta_1(\text{🌧}) \cdot a_{\text{🌧},\text{☀}}, \delta_1(\text{☁}) \cdot a_{\text{☁},\text{☀}}) \cdot b_{\text{☀},\text{☂}}$$
$$= \max(0.3 \cdot 0.8, 0.0667 \cdot 0.2, 0.233 \cdot 0.2) \cdot 0.1 = 0.024$$
$$\psi_2(\text{☀}) = \text{☀}$$

   The likelihood is stored in δ_2, the most likely predecessor in ψ_2.

   The same procedure is executed with states 🌧 and ☁:

$$\delta_2(\text{🌧}) = \max(\delta_1(\text{☀}) \cdot a_{\text{☀},\text{🌧}}, \delta_1(\text{🌧}) \cdot a_{\text{🌧},\text{🌧}}, \delta_1(\text{☁}) \cdot a_{\text{☁},\text{🌧}}) \cdot b_{\text{🌧},\text{☂}}$$
$$= \max(0.3 \cdot 0.05, 0.0667 \cdot 0.6, 0.233 \cdot 0.3) \cdot 0.8 = 0.056$$
$$\psi_2(\text{🌧}) = \text{☁}$$
$$\delta_2(\text{☁}) = \max(\delta_1(\text{☀}) \cdot a_{\text{☀},\text{☁}}, \delta_1(\text{🌧}) \cdot a_{\text{🌧},\text{☁}}, \delta_1(\text{☁}) \cdot a_{\text{☁},\text{☁}}) \cdot b_{\text{☁},\text{☂}}$$
$$= \max(0.3 \cdot 0.15, 0.0667 \cdot 0.2, 0.233 \cdot 0.5) \cdot 0.3 = 0.0350$$
$$\psi_2(\text{☁}) = \text{☁}$$

HMM/Viterbi Algorithm



HMM/Viterbi Algorithm

Recursion ($n = 3$):

$$\delta_3(\text{sunny}) = \max(\delta_2(\text{sunny}) \cdot a_{\text{sunny},\text{sunny}}, \delta_2(\text{rainy}) \cdot a_{\text{rainy},\text{sunny}}, \delta_2(\text{cloudy}) \cdot a_{\text{cloudy},\text{sunny}}) \cdot b_{\text{sunny},\text{umbrella}}$$

$$= \max(0.024 \cdot 0.8, 0.056 \cdot 0.2, 0.035 \cdot 0.2) \cdot 0.1 = 0.0019$$

$$\psi_3(\text{sunny}) = \text{sunny}$$

$$\delta_3(\text{rainy}) = \max(\delta_2(\text{sunny}) \cdot a_{\text{sunny},\text{rainy}}, \delta_2(\text{rainy}) \cdot a_{\text{rainy},\text{rainy}}, \delta_2(\text{cloudy}) \cdot a_{\text{cloudy},\text{rainy}}) \cdot b_{\text{rainy},\text{umbrella}}$$

$$= \max(0.024 \cdot 0.05, 0.056 \cdot 0.6, 0.035 \cdot 0.3) \cdot 0.8 = 0.0269$$

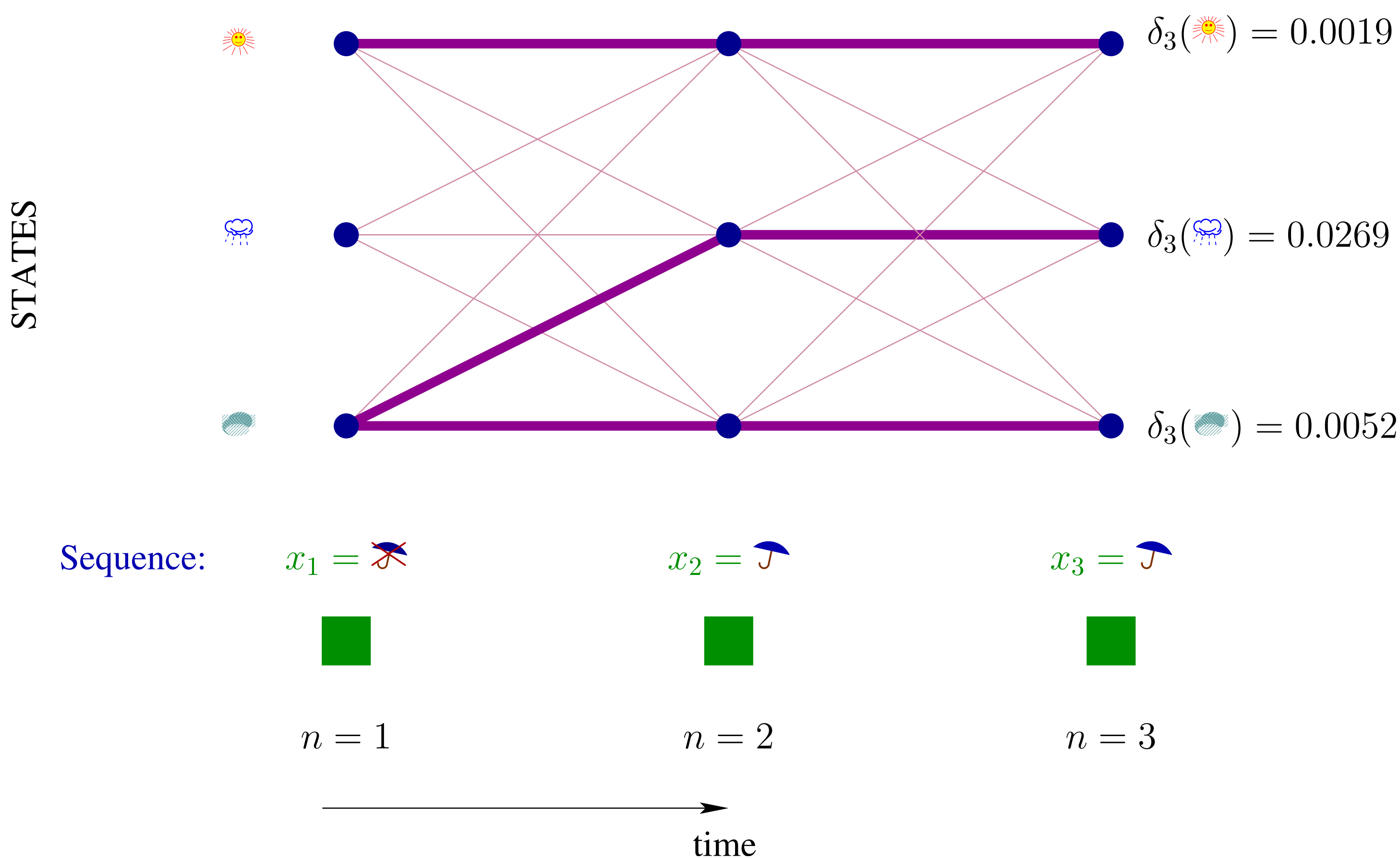
$$\psi_3(\text{rainy}) = \text{rainy}$$

$$\delta_3(\text{cloudy}) = \max(\delta_2(\text{sunny}) \cdot a_{\text{sunny},\text{cloudy}}, \delta_2(\text{rainy}) \cdot a_{\text{rainy},\text{cloudy}}, \delta_2(\text{cloudy}) \cdot a_{\text{cloudy},\text{cloudy}}) \cdot b_{\text{cloudy},\text{umbrella}}$$

$$= \max(0.0024 \cdot 0.15, 0.056 \cdot 0.2, 0.035 \cdot 0.5) \cdot 0.3 = 0.0052$$

$$\psi_3(\text{rainy}) = \text{cloudy}$$

HMM/Viterbi Algorithm



HMM/Viterbi Algorithm

3. Termination
   The globally most likely path is determined, starting by looking for the last state of the most likely sequence.

$P^*(X|\boldsymbol{\Theta}) = \max(\delta_3(i)) = \delta_3($$) = 0.0269$

$q_3^* = \operatorname{argmax}(\delta_3(i)) =$ 

4. Backtracking
   The best sequence of states can be read from the ψ vectors.
   $n = N - 1 = 2$:

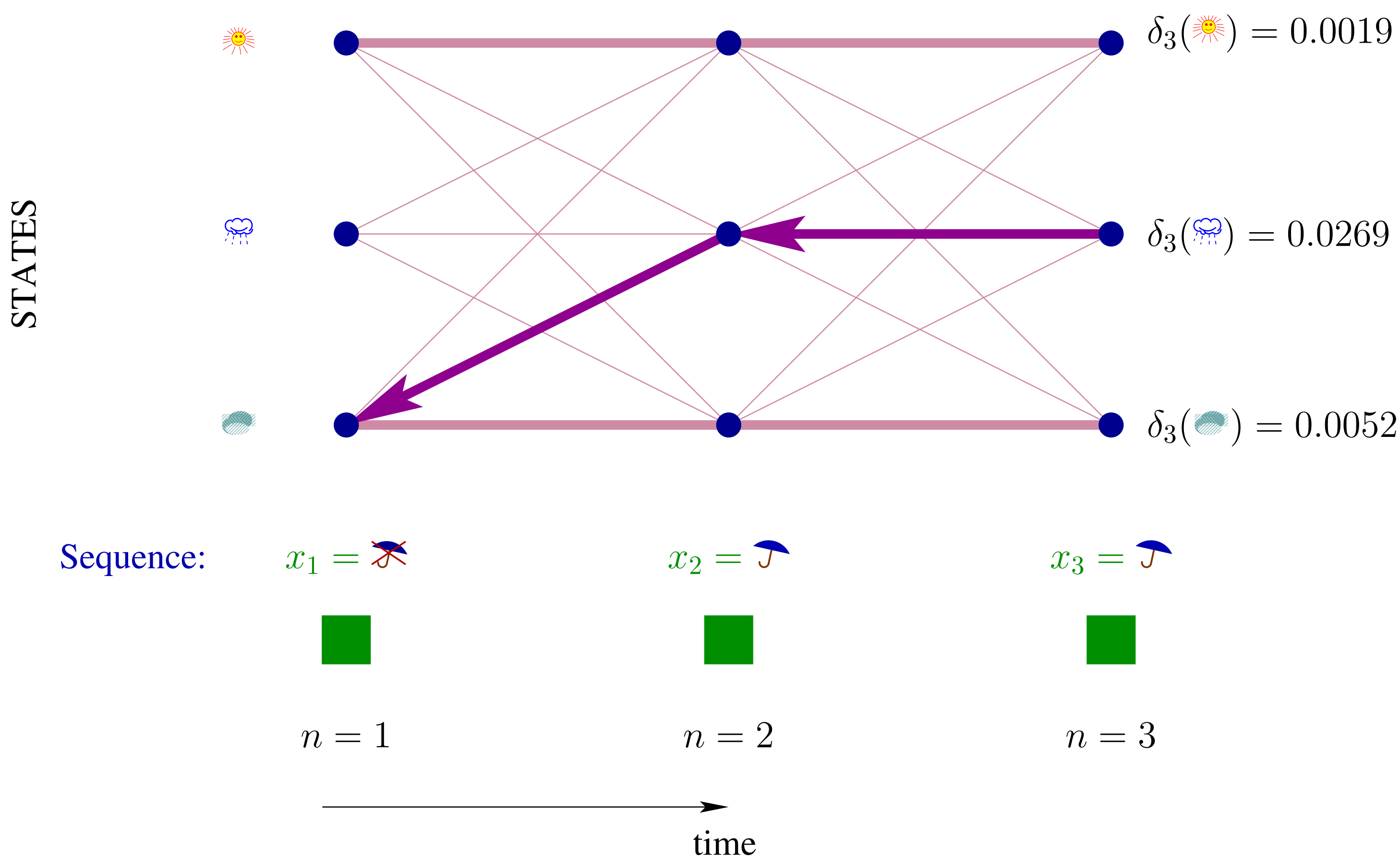
   $q_2^* = \psi_3(q_3^*) = \psi_3($$) =$ 

   $n = N - 1 = 1$:

   $q_1^* = \psi_2(q_2^*) = \psi_2($$) =$ 

The most likely weather sequence is: $Q^* = \{q_1^*, q_2^*, q_3^*\} = \{$, , $\}$.

Backtracking:



HMM/Parameter Estimation

Problem 3: Parameter estimation for HMMs

- Given: HMM structure (N_s states, K observation symbols)
- Given: Training sequence $X = \{x_1, \ldots, x_N\}$
- Wanted: optimal parameter values $\hat{\boldsymbol{\Theta}} = \{\hat{\boldsymbol{\pi}}, \hat{\mathbf{A}}, \hat{\mathbf{B}}\}$

$$P(X|\hat{\boldsymbol{\Theta}}) = \max_{\boldsymbol{\Theta}} P(X|\boldsymbol{\Theta}) = \max_{\boldsymbol{\Theta}} \sum_{Q \in \mathcal{Q}^N} P(X, Q|\hat{\boldsymbol{\Theta}})$$

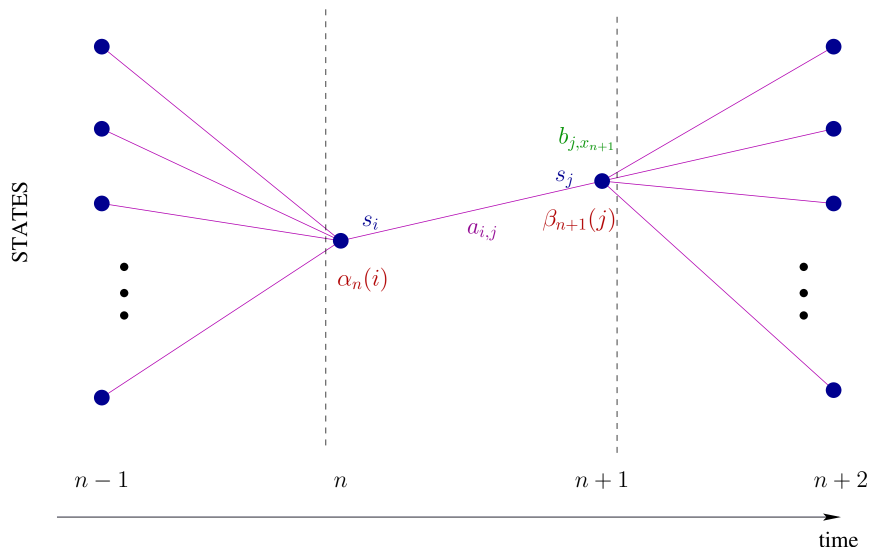
Baum-Welch Algorithm or EM (Expectation-Maximization) Algorithm

- Iterative optimization of parameters $\boldsymbol{\Theta} \rightarrow \hat{\boldsymbol{\Theta}}$
- In the terminology of the EM algorithm we have
  - observable variables: observation sequence X
  - hyper-parameters: state sequence Q

HMM/Parameter Estimation

Transition probabilities for $s_i \rightarrow s_j$ at time n (for given $\boldsymbol{\Theta}$):

$$\xi_n(i,j) := P(q_n = s_i, q_{n+1} = s_j | X, \boldsymbol{\Theta}) = \frac{\alpha_n(i) \cdot a_{i,j} \cdot b_{j,x_{n+1}} \cdot \beta_{n+1}(j)}{P(X|\boldsymbol{\Theta})}$$



State probability for s_i at time n (for given $\boldsymbol{\Theta}$):

$$\gamma_n(i) := P(q_n = s_i | X, \boldsymbol{\Theta}) = \frac{\alpha_n(i) \cdot \beta_n(i)}{P(X|\boldsymbol{\Theta})} = \sum_{j=1}^{N_s} \xi_n(i,j)$$

$$P(X|\boldsymbol{\Theta}) = \sum_{i=1}^{N_s} \alpha_n(i) \cdot \beta_n(i) \quad \text{(cf. forward/backward algorithm)}$$

HMM/Parameter Estimation

Summing over time n gives expected numbers # (frequencies) for

$$\sum_{n=1}^{N} \gamma_n(i) \quad \ldots \text{\# of transitions from state } s_i$$

$$\sum_{n=1}^{N} \xi_n(i,j) \quad \ldots \text{\# of transitions from state } s_i \text{ to state } s_j$$

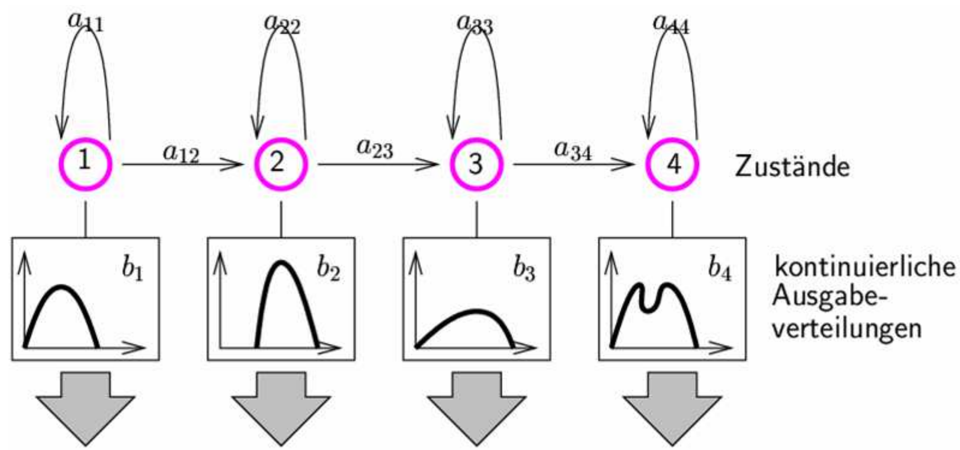
Baum-Welch update of HMM parameters:

$$\bar{\pi}_i = \gamma_1(i) \qquad \ldots \text{\# of state } s_i \text{ at time } n = 1$$

$$\bar{a}_{i,j} = \frac{\sum_{n=1}^{N-1} \xi_n(i,j)}{\sum_{n=1}^{N-1} \gamma_n(i,j)} \qquad \ldots \frac{\text{\# of transitions from state } s_i \text{ to state } s_j}{\text{\# of transitions from state } s_i}$$

$$\bar{b}_{j,k} = \frac{\sum_{n=1}^{N} \gamma_n(i,j) \cdot [x_n = v_k]}{\sum_{n=1}^{N} \gamma_n(i,j)} \qquad \ldots \frac{\text{\# of state } s_i \text{ with } v_k \text{ emitted}}{\text{\# of state } s_i}$$

HMM/Parameter Estimation



- Gaussian (normal distributed) emission probabilities:

$$b_j(x) = \mathcal{N}(x|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

- Mixtures of Gaussians

$$b_j(x) = \sum_{k=1}^{K} c_{jk} \, \mathcal{N}(x|\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}), \quad \sum_{k=1}^{K} c_{jk} = 1$$

- "Semi-continuous" emission probabilities:

$$b_j(x) = \sum_{k=1}^{K} c_{jk} \, \mathcal{N}(x|\boldsymbol{\mu}_{k}, \boldsymbol{\Sigma}_{k}), \quad \sum_{k=1}^{K} c_{jk} = 1$$

HMM/Parameter Estimation

Problems encountered for HMM parameter estimation

- many word models/HMM states/parameters
- ...always too less training data!

⇒ Consequences:

- large variance of estimated parameters
- large variance in objective function $P(X|\mathbf{\Theta})$
- vanishing statistics
- ⇒ zero valued parameters $\hat{a}_{i,j}, \hat{b}_{j,k}, \hat{\mathbf{\Sigma}}_k, \hat{\mathbf{\Sigma}}_{jk}, \ldots$

⇒ Possible remedies (besides using more training data):

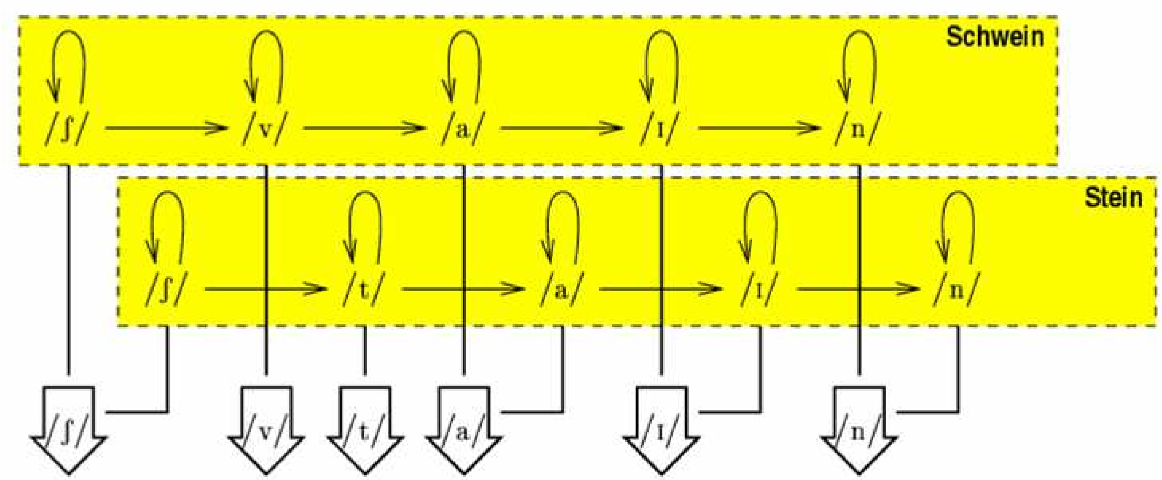
- fix some parameter values
- tying parameter values for similar models
- interpolation of sensible parameters by robust parameters
- smoothing of probability density functions
- defining limits for sensible density parameters

HMM/Parameter Estimation

Parameter tying

- simultaneous identification of parameters for similar models
- ⇒ forces identical parameter values
- ⇒ reduces parameter space dimension

Example (state tying):



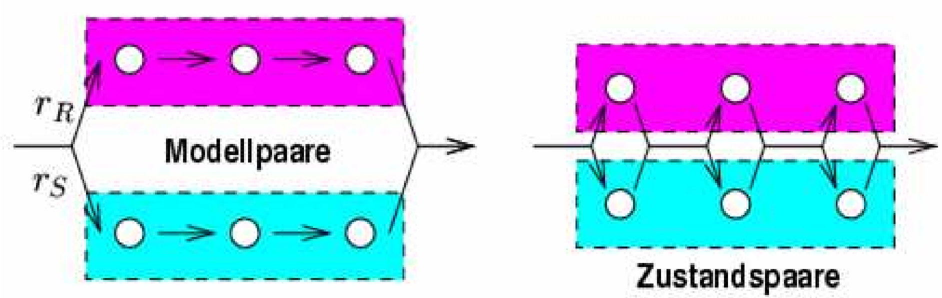
Automatic determination of states that can be tied, e.g., by mutual information

HMM/Parameter Estimation

Parameter interpolation

- instead of fixed tying of states:
- interpolate parameters of similar models

$$P(X|\mathbf{\Theta}_R, \mathbf{\Theta}_S, r_R, r_S) = r_R \cdot P(X|\mathbf{\Theta}_R) + r_S \cdot P(X|\mathbf{\Theta}_S), \quad r_R + r_S = 1$$



- especially suited for semi-continuous emission pdfs
- weights r_R, r_S can be chosen heuristically or included in the Baum-Welch algorithm

References

- R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*. Wiley&Sons, Inc., 1973.
- S. Bengio, *An Introduction to Statistical Machine Learning – EM for GMMs*, Dalle Molle Institute for Perceptual Artificial Intelligence.
- E.G. Schukat-Talamazzini, *Automatische Spracherkennung*, Vieweg-Verlag, 1995.
- L.R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, Proceedings of the IEEE, Vol. 77, No. 2, pp. 257-286, 1989.