Lecture

Speech and Audio Signal Processing

TECHNISCHE UNIVERSITÄT DARMSTADT

Lecture 11: Hidden Markov Models (HMM) and Speech Recognition, Part I

Content

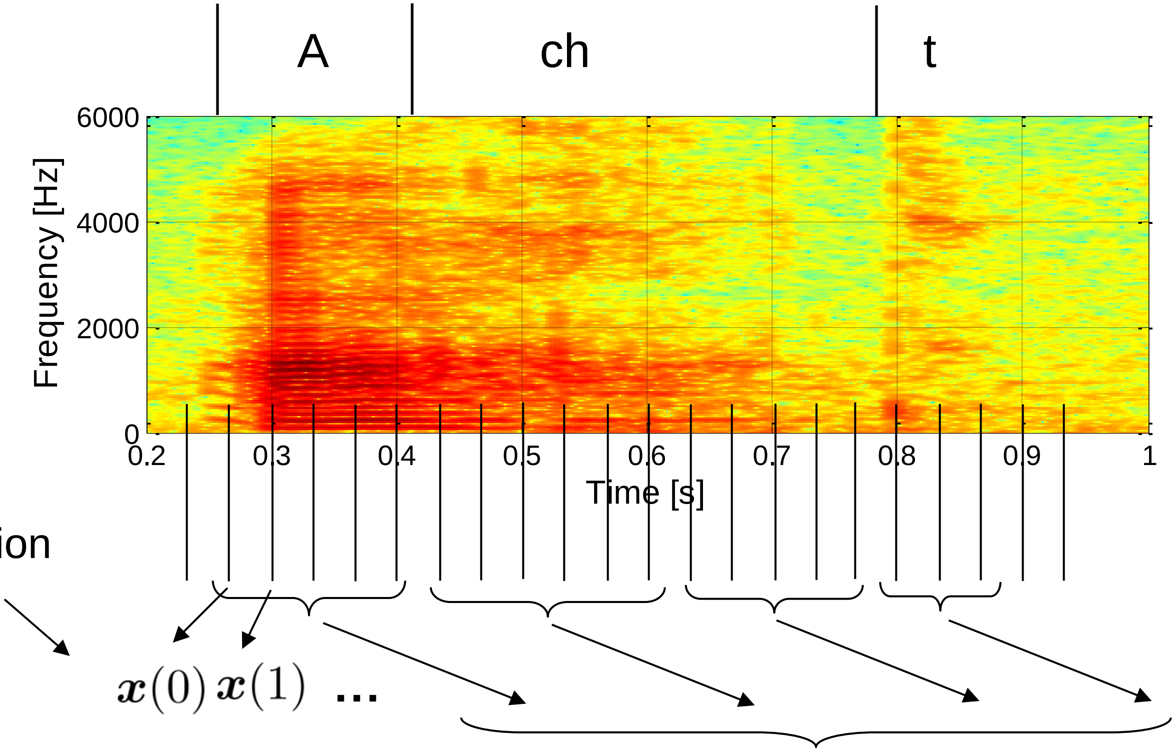
- Principle of Speech recognition
- HMM: General definition
- The three basic problems of HMMs
  - Evaluation problem
  - Decoding problem
  - Model parameter estimation problem
- Evaluation problem:
  - Trellis structures
  - Forward algorithm
- Decoding problem:
  - Viterbi algorithm

Principle of speech recognition

- Generals:
  - Speech is a sequence of phones (“Laute”).
  - Phones with the same meaning can be group to phonemes (“Lautgruppen”).
  - For speech recognition it is important to differentiate between phonemes.
- Procedure:
  - Use the speech signal and calculate an equidistant sequence of feature vectors, typically MFCC vectors.

Principle of speech recognition

- Procedure:

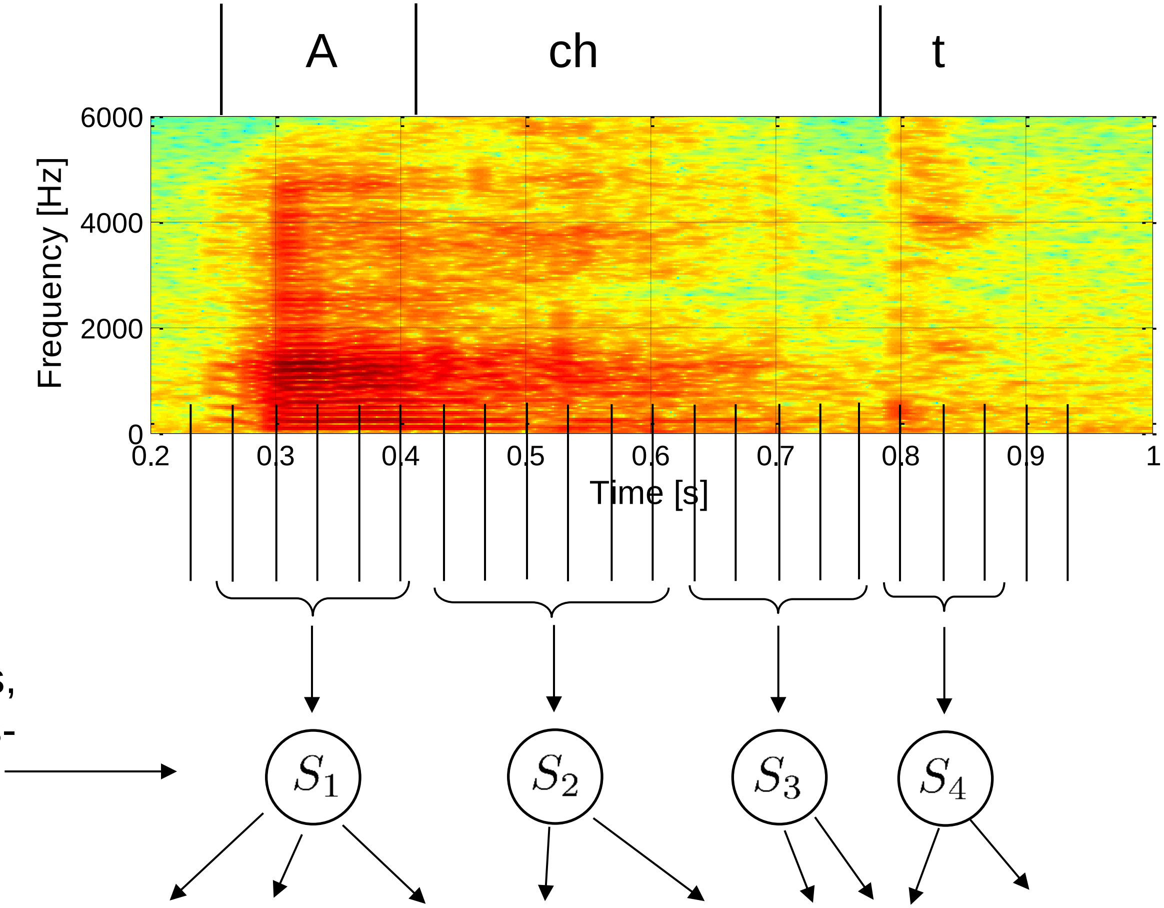


Blockwise calculation of feature vectors

Different probability models for the generation of the feature vectors according to the corresponding phoneme

Principle of speech recognition

- Procedure:



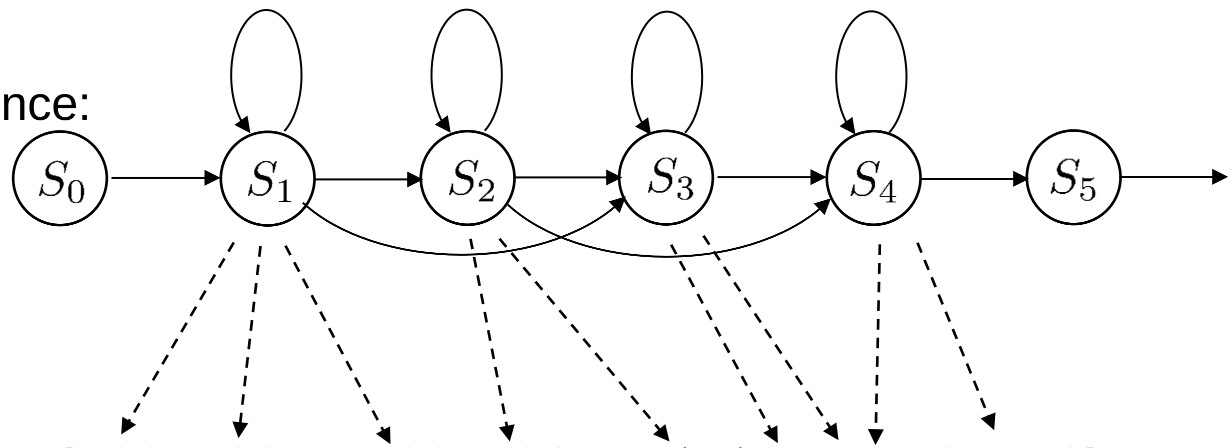
Different states with different probability models, i.e., for each state a corres-ponding pdf is modeled.

Observed feature vector sequence

$$\boldsymbol{X} = [\boldsymbol{x}(0)\ \ \boldsymbol{x}(1)\ \ldots\ \boldsymbol{x}(4)\ \ \boldsymbol{x}(5)\ \ldots \boldsymbol{x}(10)\ \ \ldots\ \ \boldsymbol{x}(T-1)]$$

HMM: The general definition

- States and feature sequence:



$$\boldsymbol{X} = [\boldsymbol{x}(0)\ \ \boldsymbol{x}(1) \ldots \boldsymbol{x}(4)\ \ \boldsymbol{x}(5) \ldots \boldsymbol{x}(10)\ \ \ldots\ \ \boldsymbol{x}(T-1)]$$

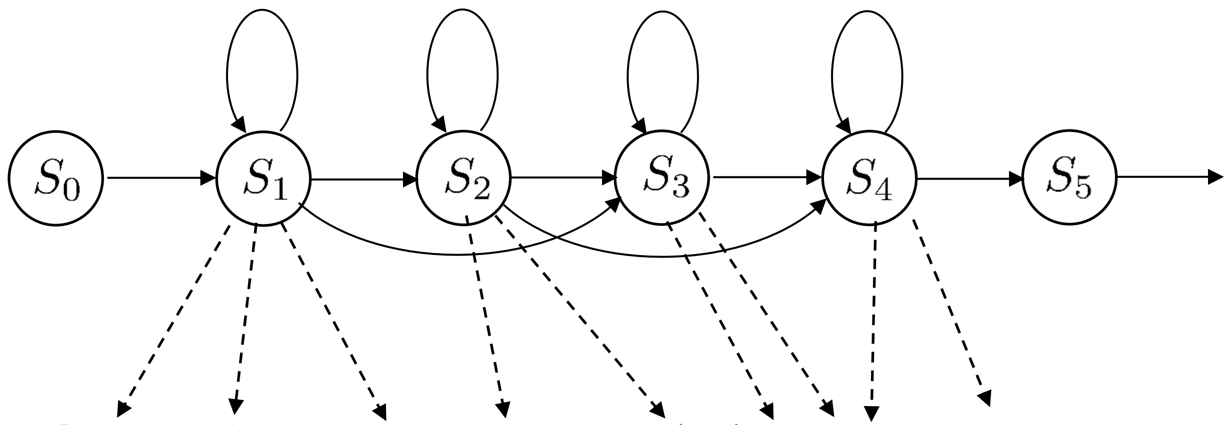
- **HMMs: The hidden model components**:
  - The hidden part of the HMM is modeled by *N* **states**:

$$S_0,\ S_1, \cdots,\ S_{N-1}$$

The states are not accessible. The transition between the states is specified by **transition probabilities**.

HMM: The general definition

- States and feature sequence:



$$\boldsymbol{X} = [\boldsymbol{x}(0) \;\; \boldsymbol{x}(1) \dots \boldsymbol{x}(4) \;\; \boldsymbol{x}(5) \dots \boldsymbol{x}(10) \;\; \dots \;\; \boldsymbol{x}(T-1)]$$
$$\boldsymbol{q} = [q(0) \;\; q(1) \dots q(4) \;\; q(5) \dots q(10) \;\; \dots \;\; q(T-1)]$$
$$= [\,S_1 \;\; S_1 \dots S_1 \;\; S_2 \dots S_2 \;\; \dots \;\; S_4\,]$$

- The hidden states generate a random processes leading to the observation sequence : $\boldsymbol{X} = [\boldsymbol{x}(0), \boldsymbol{x}(1), \dots, \boldsymbol{x}(T-1)]$

- The sequence of the hidden states is defined as $\boldsymbol{q}$, where the elements $q(n)$ are the hidden states.

$$\boldsymbol{q} = [q(0), q(1), \dots, q(T-1)]^{\mathrm{T}} \quad \text{with: } q(n) \in \{S_0, S_1, \dots, S_{N-1}\}$$

HMM: The general definition

- HMMs: The hidden model components:
  - ***The observation probabilities*** *(the probability – pdf – of each state)*
    The probabilities of the observation vectors only depend on the current state:

$$p\left(\boldsymbol{x}(n) \middle| q(n) = S_j,\, q(n-1) = S_i,\, \ldots,\, q(0) = S_k,\, \boldsymbol{x}(n-1),\, \ldots,\, \boldsymbol{x}(0)\right)$$
$$= p\left(\boldsymbol{x}(n) \middle| q(n) = S_j\right)$$

$$\boxed{b_j(\boldsymbol{x}(n)) = p\left(\boldsymbol{x}(n) \middle| q(n) = S_j\right)}$$
Observation probabilities

  - ***The transition probabilities***
    The transition between the states is described by probabilities. They only depend on the current state of origin, not on other previous states:

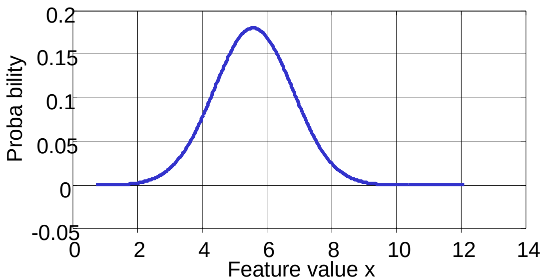
$$p\left(q(n) = S_j \middle| q(n-1) = S_i,\, \ldots,\, q(0) = S_k\right)$$
$$= p\left(q(n) = S_j \middle| q(n-1) = S_i\right)$$

$$\boxed{a_{i,j} = p\left(q(n) = S_j \middle| q(n-1) = S_i\right)}$$
Transition probabilities

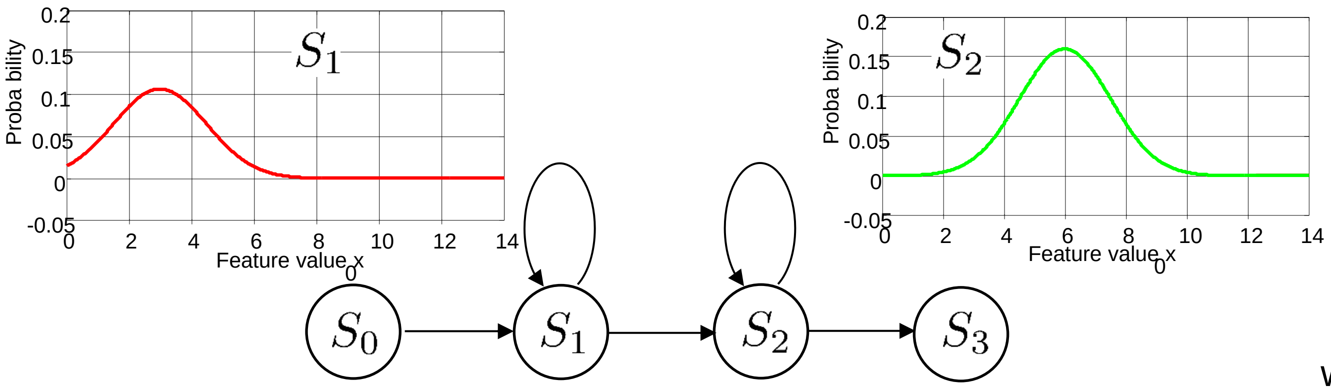
Models for stationary and instationary systems:

- Stationary system: One fixed probability modeling the feature values:

$$p(\boldsymbol{X}|H_i) = \prod_{n=0}^{N-1} p(\boldsymbol{x}(n)|H_i)$$



- Instationary system: Several probabilities modeling the features values (observation probabilities), plus transition probabilities in between => HMM:



$$p(\boldsymbol{X}|\lambda) = \sum_{\boldsymbol{q}_i \in \boldsymbol{Q}} a_{S_0,q_i(0)}\, b_{q_i(0)}(\boldsymbol{x}(0))\; a_{q_i(0),q_i(1)}\, b_{q_i(1)}(\boldsymbol{x}(1)) \;\ldots$$

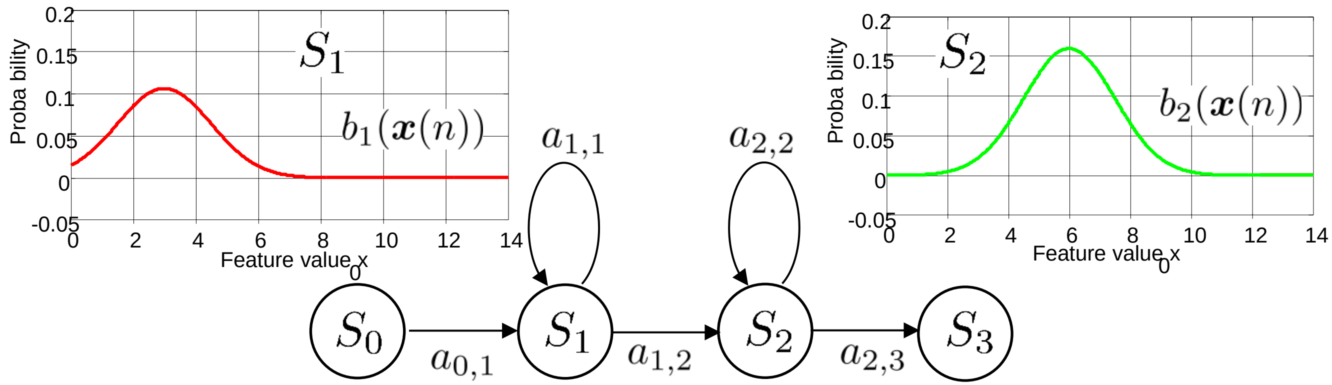
with:

$\lambda \in [a, b]$

$\boldsymbol{Q}$: all paths $\boldsymbol{q}_i$

Summation of the different path probabilities

- Instationary system: HMM model:



Many paths leading from start to end state:
=> Model probability is summation of all path probabilities

Path 1: $S_1 \quad S_1 \quad S_1 \quad S_2$ $\qquad p_1 = a_{0,1} b_1(x(0)) \; a_{1,1} b_1(x(1)) \; a_{1,1} b_1(x(2)) \; a_{1,2} b_2(x(3))$

Path 2: $S_1 \quad S_1 \quad S_2 \quad S_2$ $\qquad p_2 = a_{0,1} b_1(x(0)) \; a_{1,1} b_1(x(1)) \; a_{1,2} b_2(x(2)) \; a_{2,2} b_2(x(3))$

Path 3: $\cdots$

Corresponding formula:

$$p(\boldsymbol{X}|\lambda) = \sum_{\boldsymbol{q}_i \in \boldsymbol{Q}} a_{S_0, q_i(0)} \, b_{q_i(0)}(\boldsymbol{x}(0)) \; a_{q_i(0), q_i(1)} \, b_{q_i(1)}(\boldsymbol{x}(1)) \; \ldots$$

with:

$\lambda \in [a, b]$

$\boldsymbol{Q}$: all paths $\boldsymbol{q}_i$

HMM: The general definition

- HMMs: The transition between states:

  - The start and end state are denoted as follows:

    S_0 start state

    S_{N-1} end state

  - Both states are not related to observation vectors.

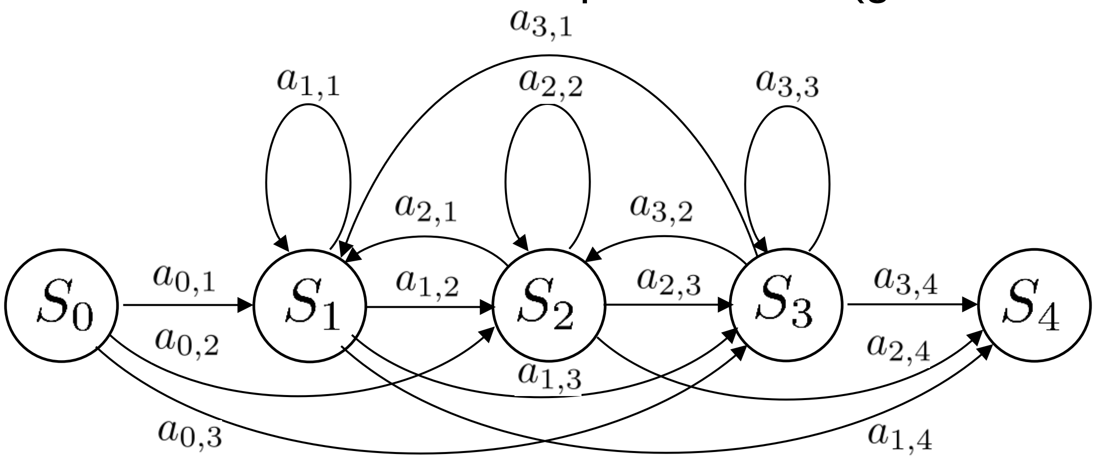
  - No transitions back to the start state are possible: $a_{i,0} = 0$

  - No direct transition from the start to the end state is possible: $a_{0,N-1} = 0$

  - No transitions leaving the end state are possible: $a_{N-1,i} = 0$

HMM: The general definition

- HMMs: States and transition probabilities (general setup):



- The transition matrix:

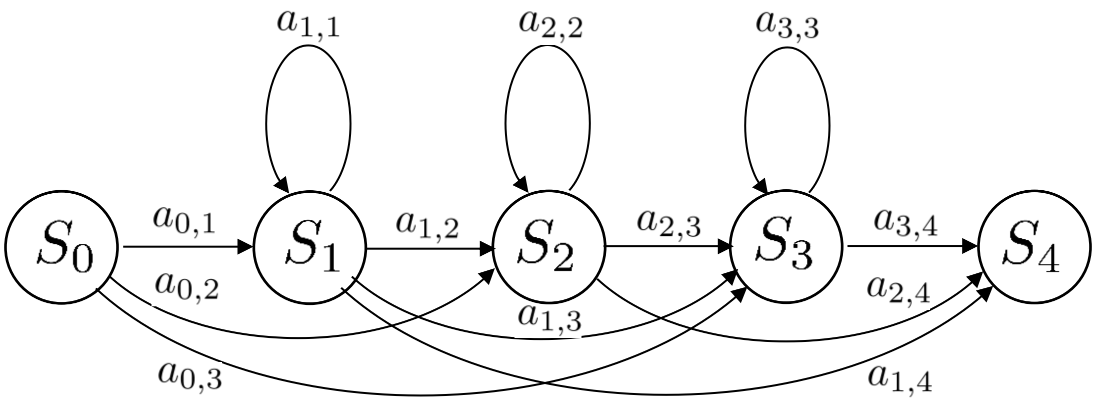
$$\boldsymbol{A} = \left\{ \begin{array}{cccccc} 0 & a_{0,1} & a_{0,2} & \cdots & a_{0,N-2} & 0 \\ 0 & a_{1,1} & a_{1,2} & \cdots & a_{1,N-2} & a_{1,N-1} \\ 0 & a_{2,1} & a_{2,2} & \cdots & a_{2,N-2} & a_{2,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{N-2,1} & a_{N-2,2} & \cdots & a_{N-2,N-2} & a_{N-2,N-1} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right\}$$

with constraints:

$$\begin{aligned} a_{i,0} &= 0 \quad \text{for} \quad i \in \{0, N-1\} \\ a_{N-1,i} &= 0 \quad \text{for} \quad i \in \{0, N-1\} \\ a_{0,N-1} &= 0 \quad \text{for} \quad i \in \{0, N-1\} \\ \\ a_{i,j} &\geq 0 \quad \text{for} \quad i,j \in \{0, N-1\} \\ \sum_{j=0}^{N-1} a_{i,j} &= 1 \quad \text{for} \quad i \in \{0, N-2\} \end{aligned}$$

HMM: The general definition

- HMMs: only transitions from left to right are possible:

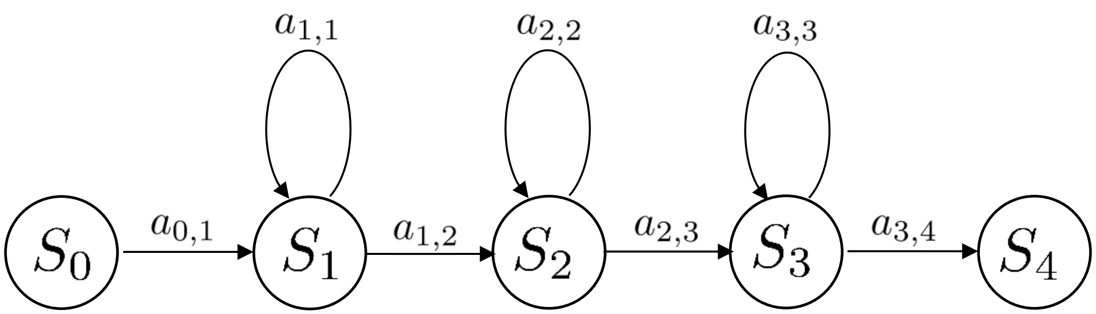


- The corresponding transition matrix:

$$\boldsymbol{A} = \left\{ \begin{array}{ccccc} 0 & a_{0,1} & a_{0,2} & a_{0,3} & 0 \\ 0 & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ 0 & 0 & a_{2,2} & a_{2,3} & a_{2,4} \\ 0 & 0 & 0 & a_{3,3} & a_{3,4} \\ 0 & 0 & 0 & 0 & 0 \end{array} \right\}$$

HMM: The general definition

- HMMs: structure of a linear model, i.e. only transitions to the right neighbor are possible:

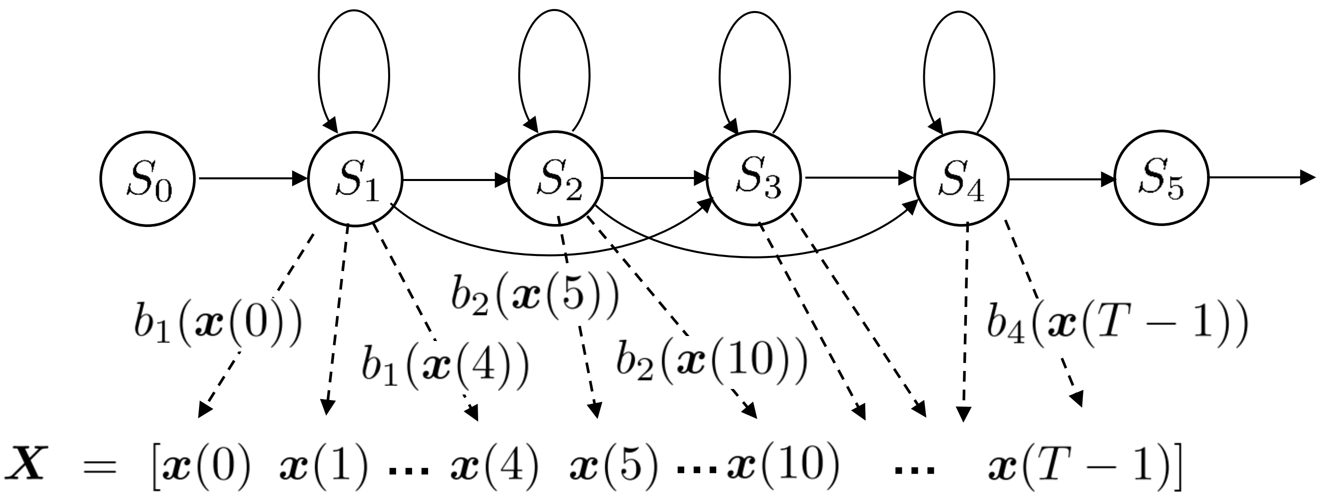


- The corresponding transition matrix:

$$\boldsymbol{A} = \left\{ \begin{array}{ccccc} 0 & a_{0,1} & 0 & 0 & 0 \\ 0 & a_{1,1} & a_{1,2} & 0 & 0 \\ 0 & 0 & a_{2,2} & a_{2,3} & 0 \\ 0 & 0 & 0 & a_{3,3} & a_{3,4} \\ 0 & 0 & 0 & 0 & 0 \end{array} \right\}$$

HMM: The general definition

- The observation probabilities: $b_j(\boldsymbol{x}(n)) = p\left(\boldsymbol{x}(n) \middle| q(n) = S_j\right)$



- Dependent on the states, different probability models are related to the observation vectors.

- The probabilities $b_j(\boldsymbol{x})$ can be defined by **discrete** or **continuous** random processes.

HMM: The general definition

- The observation probabilities can be discrete or continuous random processes: $b_j(\boldsymbol{x}(n)) = p\left(\boldsymbol{x}(n) \middle| q(n) = S_j\right)$

- **1) Modeling by continuous random processes:**
  - Continuous random processes are typically described by Gaussian mixture models:

$$b_j(\boldsymbol{x}) = \sum_{k=0}^{K-1} g_{j,k} \, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{j,k}, \boldsymbol{\Sigma}_{j,k})$$

  - For each state, *K* Gaussian mixtures are used for vectors of length *D*:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D \, |\boldsymbol{\Sigma}|}} \, e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}$$

with: $\boldsymbol{x} = \left[x_0, x_1, \ldots, x_{D-1}\right]^{\mathrm{T}}$

HMM: The general definition

- The observation probabilities can be discrete or continuous random processes: $b_j(\boldsymbol{x}(n)) = p\left(\boldsymbol{x}(n) \middle| q(n) = S_j\right)$

- **2) Modeling by discrete random processes:**
  - Assume $\boldsymbol{x}(n)$ can be one of *K* possible vectors.

$$\boldsymbol{x}(n) \in \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{K-1}\}$$

  - The probability of the *k*-th vector in state *j* is:

$$p\left(\boldsymbol{x}(n) = \boldsymbol{x}_k \middle| q(n) = S_j\right) = b_{j,k}$$

- Resulting in the following probability matrix:

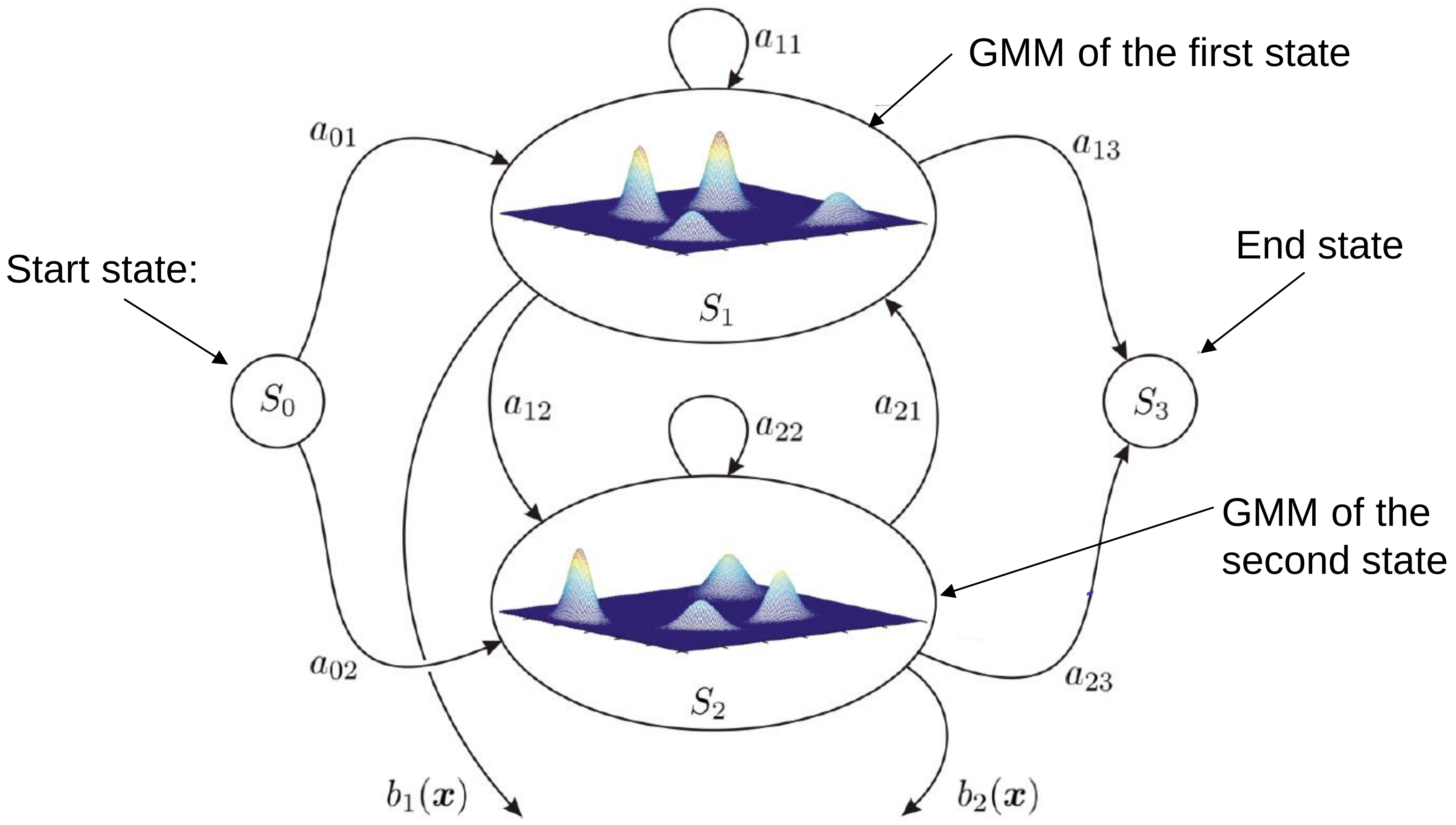
$$\boldsymbol{B} = \left\{ \begin{array}{ccccc} b_{1,0} & b_{1,1} & \cdots & b_{1,K-2} & b_{1,K-1} \\ b_{2,0} & b_{2,1} & \cdots & b_{2,K-2} & b_{2,K-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{N-2,0} & b_{N-2,1} & \cdots & b_{N-2,K-2} & b_{N-2,K-1} \end{array} \right\}$$

with constraints:

$$\begin{aligned} b_{j,k} \geq 0 \quad \text{for} \quad & i \in \{1, N-2\}, \\ & k \in \{0, K-1\} \\ \sum_{k=0}^{K-1} b_{i,k} = 1 \quad \text{for} \quad & i \in \{1, N-2\} \end{aligned}$$

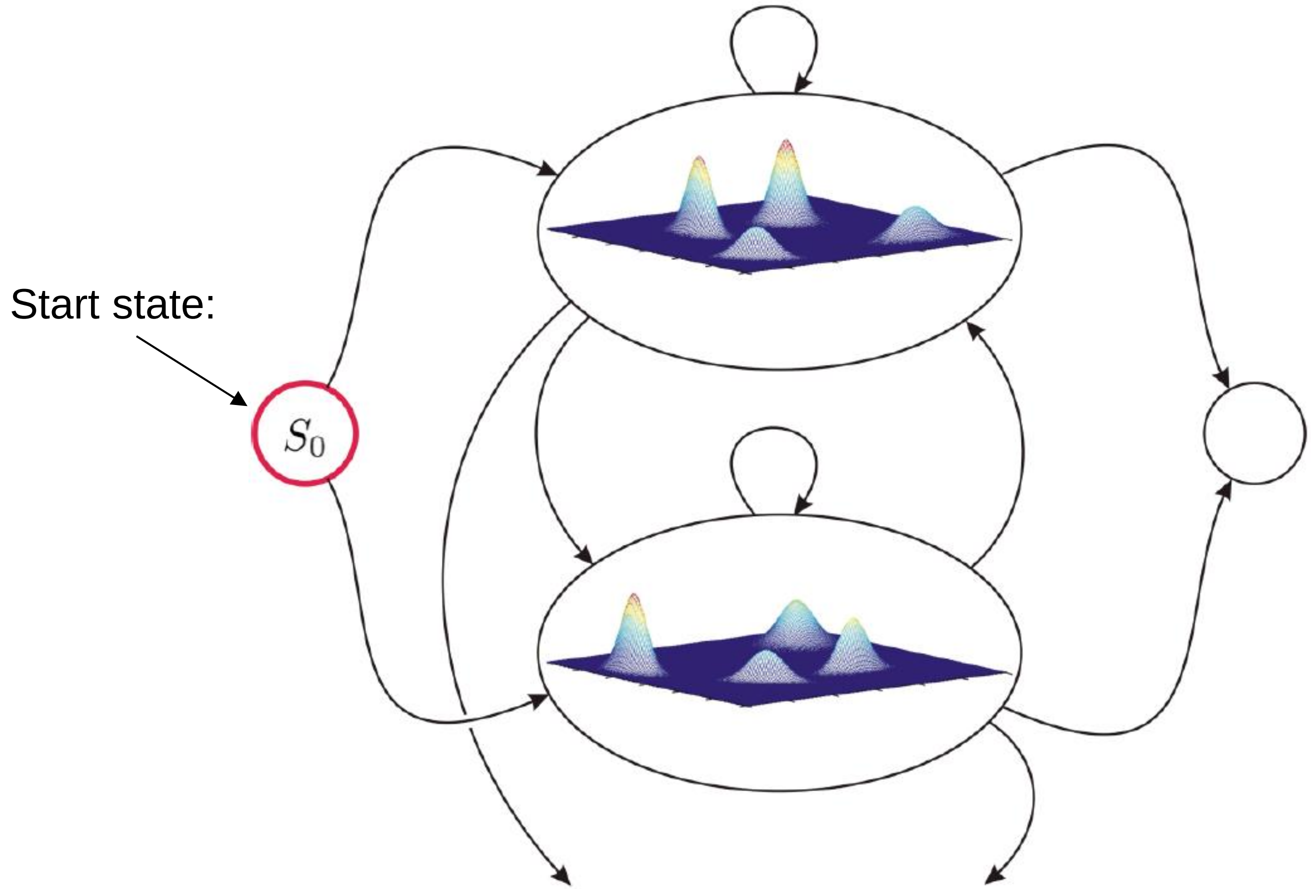
HMM: The general definition for cont. observation pdf's

- Random process based on a Markov model which generates observations:



Example for the generation of observation vectors

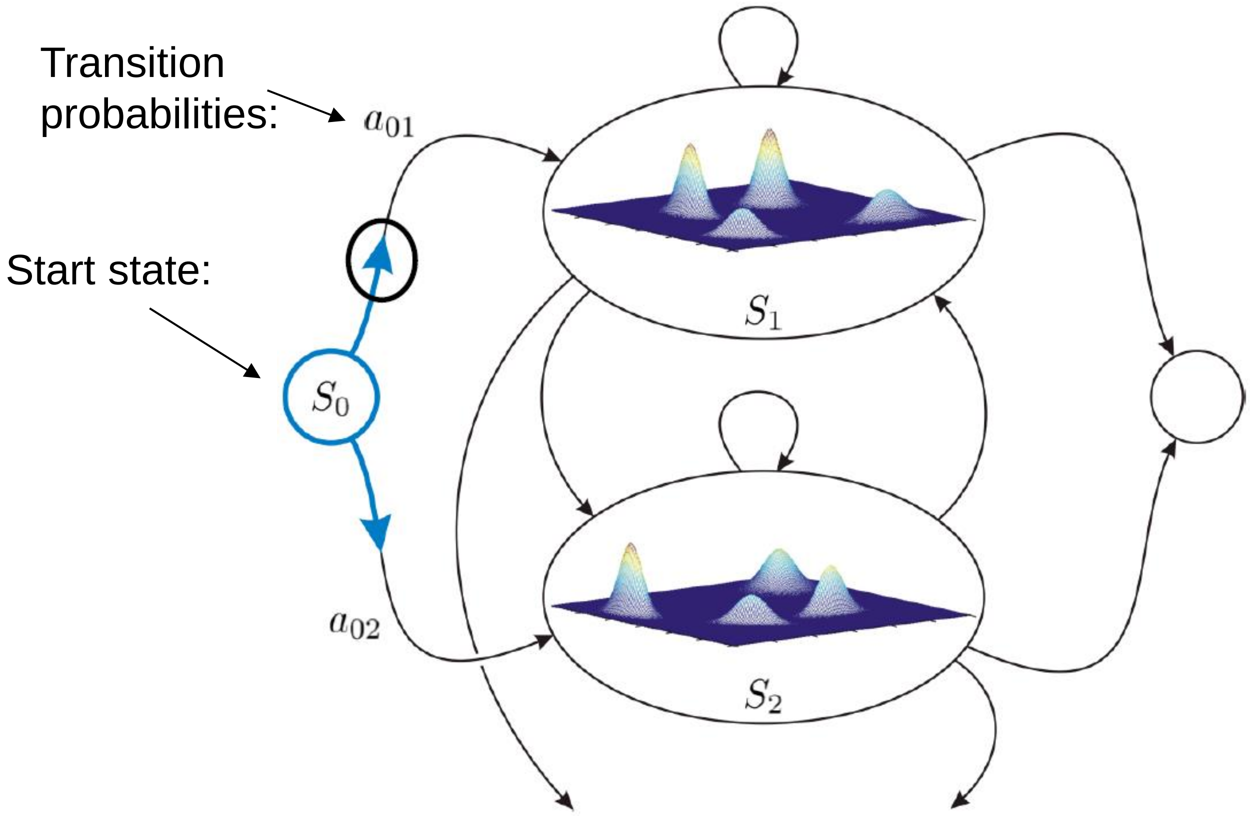
- Start state:



Sequence observed so far: $\boldsymbol{X} = [\ldots$

Example for the generation of observation vectors

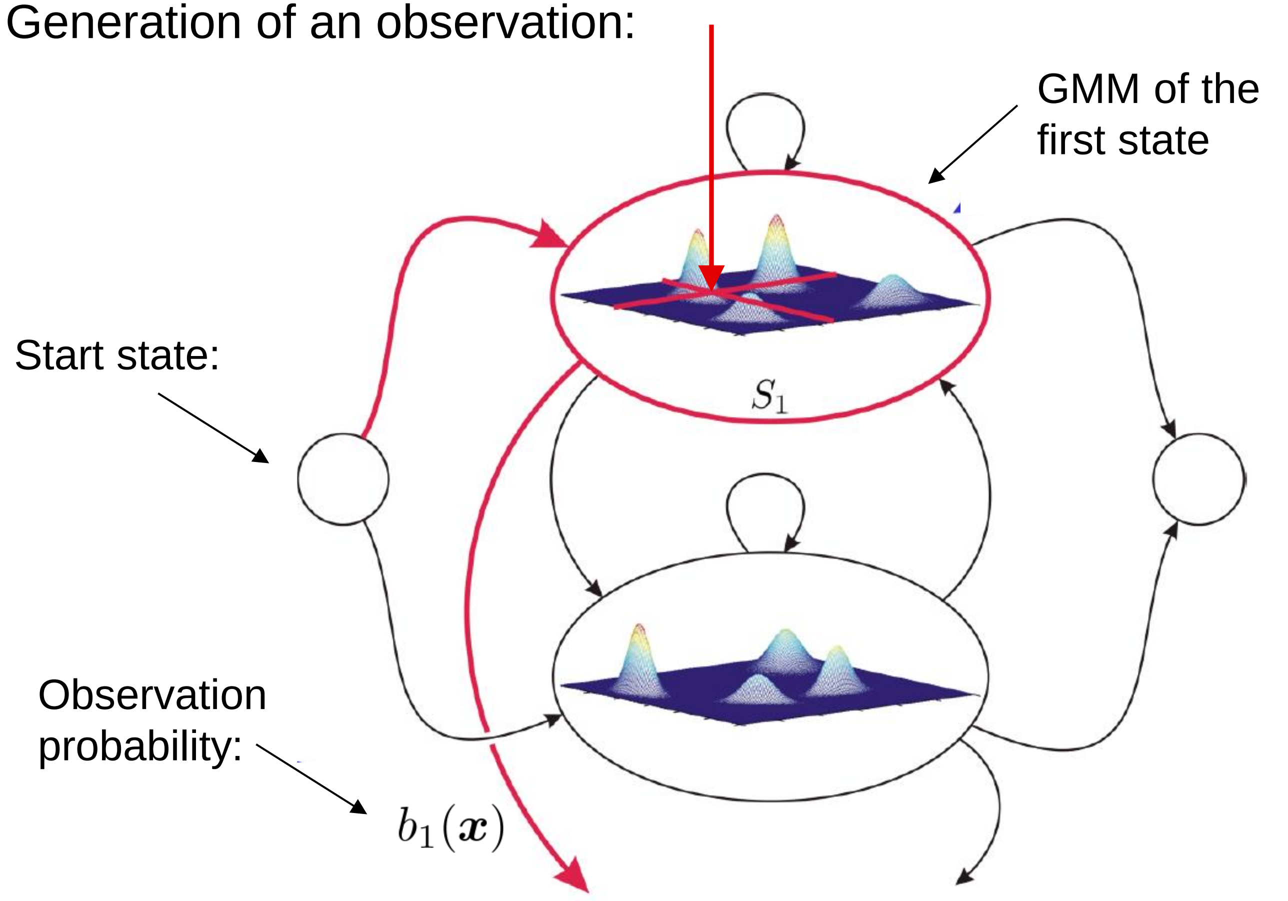
- Determine first state transition:



Sequence observed so far: $\boldsymbol{X} = [\ldots$

Example for the generation of observation vectors

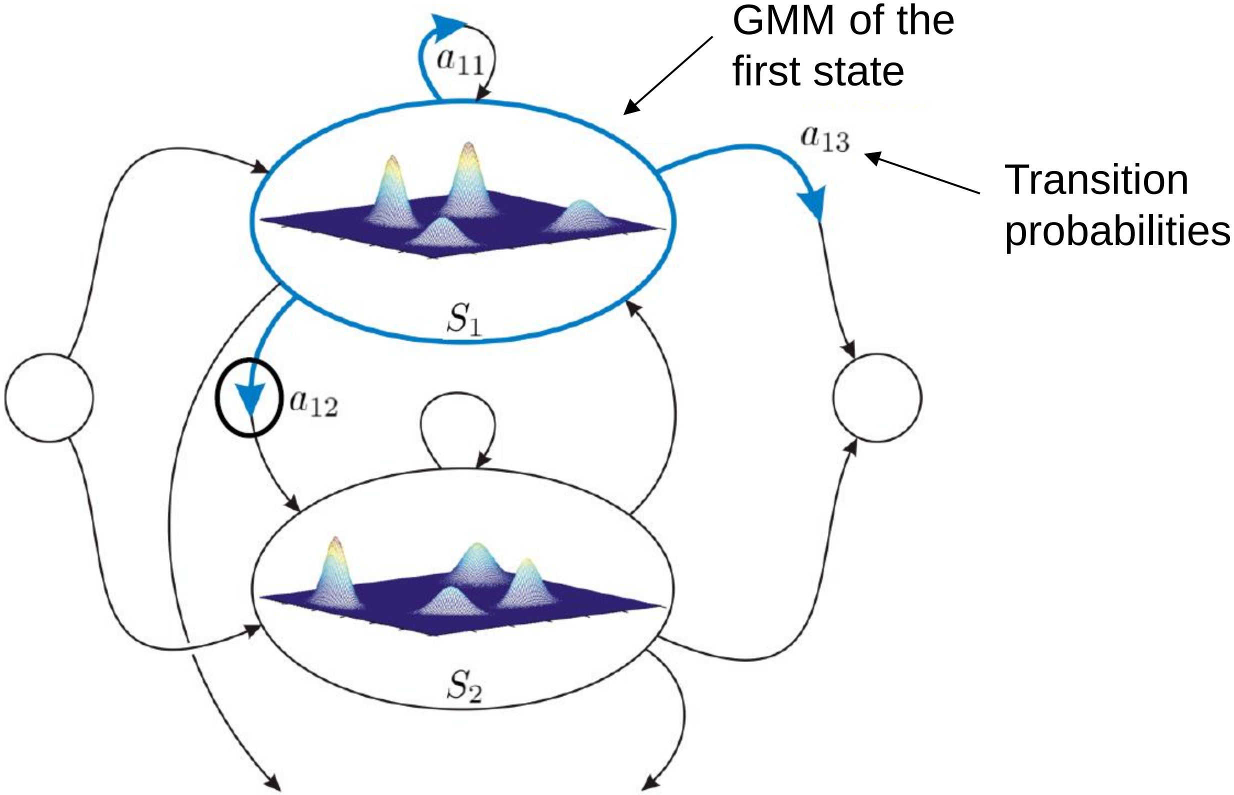
- Generation of an observation:



Sequence observed so far: $\boldsymbol{X} = [\boldsymbol{x}(0), \ldots$

Example for the generation of observation vectors

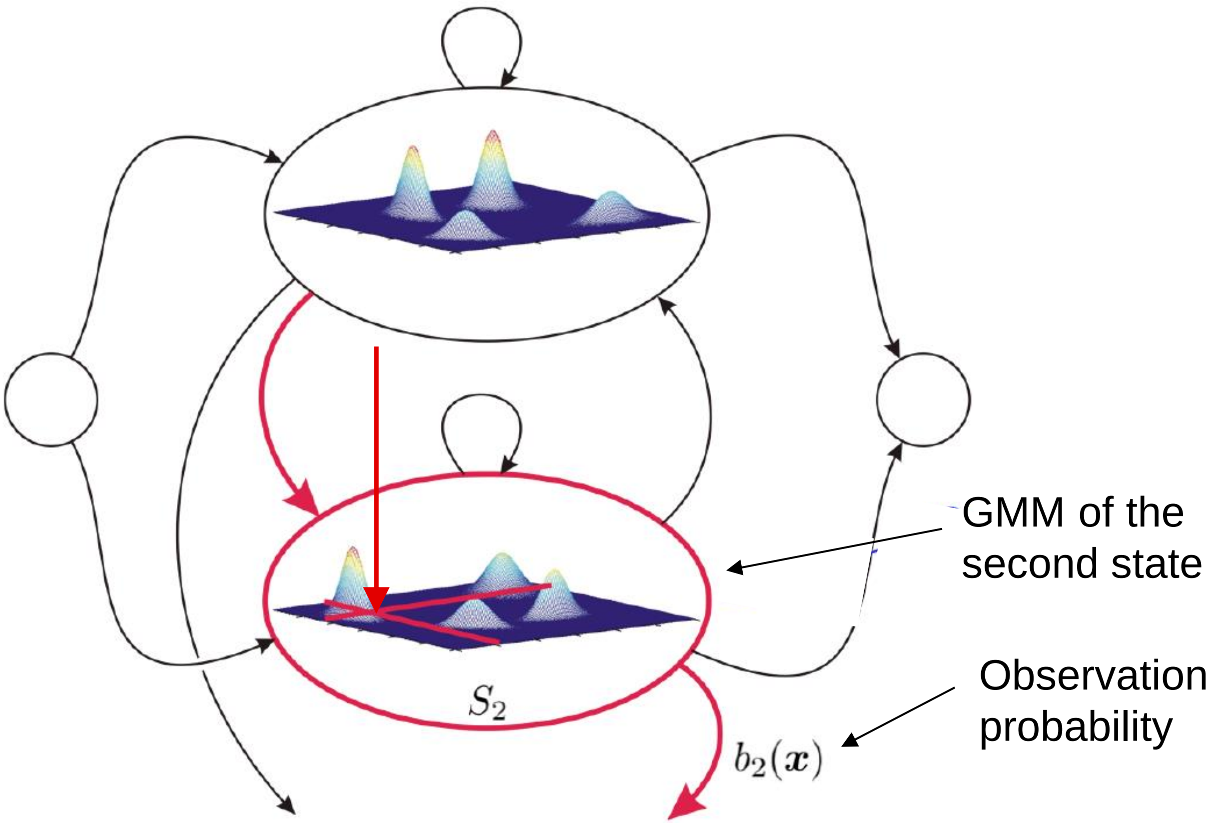
- Determine second state transition:



Sequence observed so far: $\boldsymbol{X} = [\boldsymbol{x}(0), \ldots$

Example for the generation of observation vectors

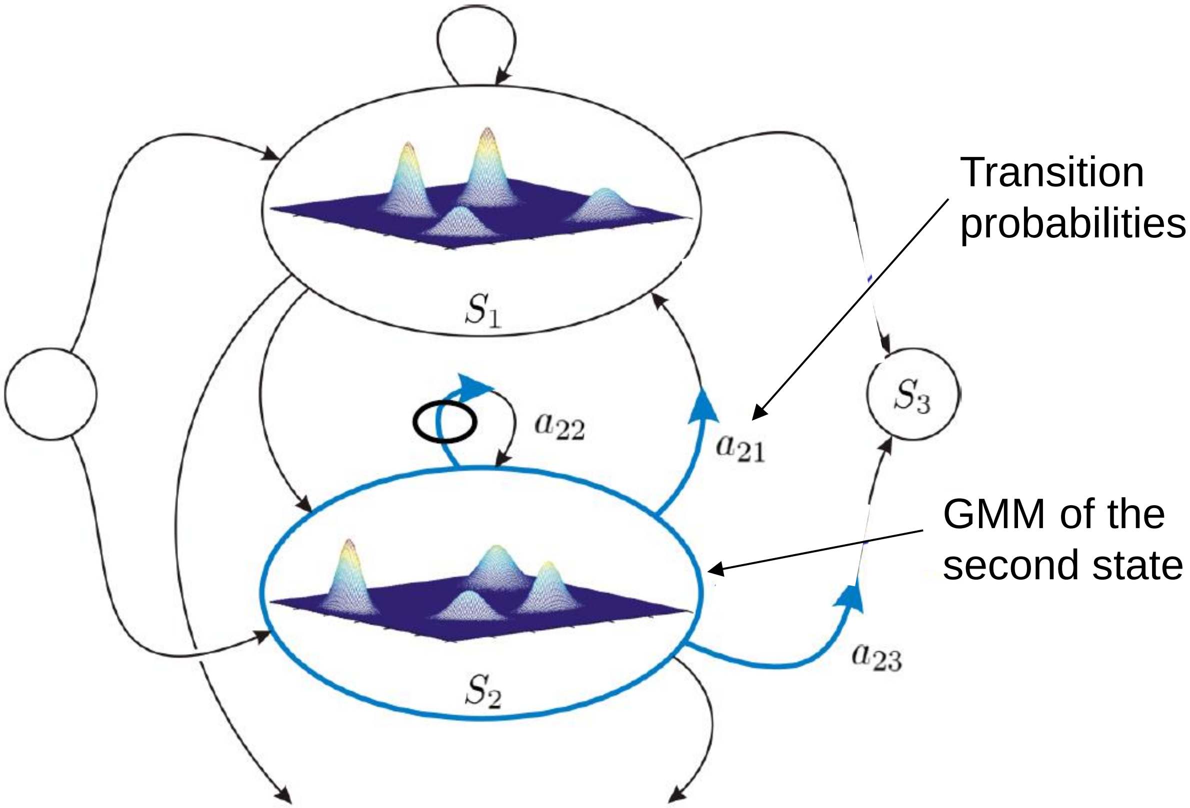
- Generation of a second observation:



Sequence observed so far: $\boldsymbol{X} = [\boldsymbol{x}(0), \boldsymbol{x}(1), \ldots$

Example for the generation of observation vectors

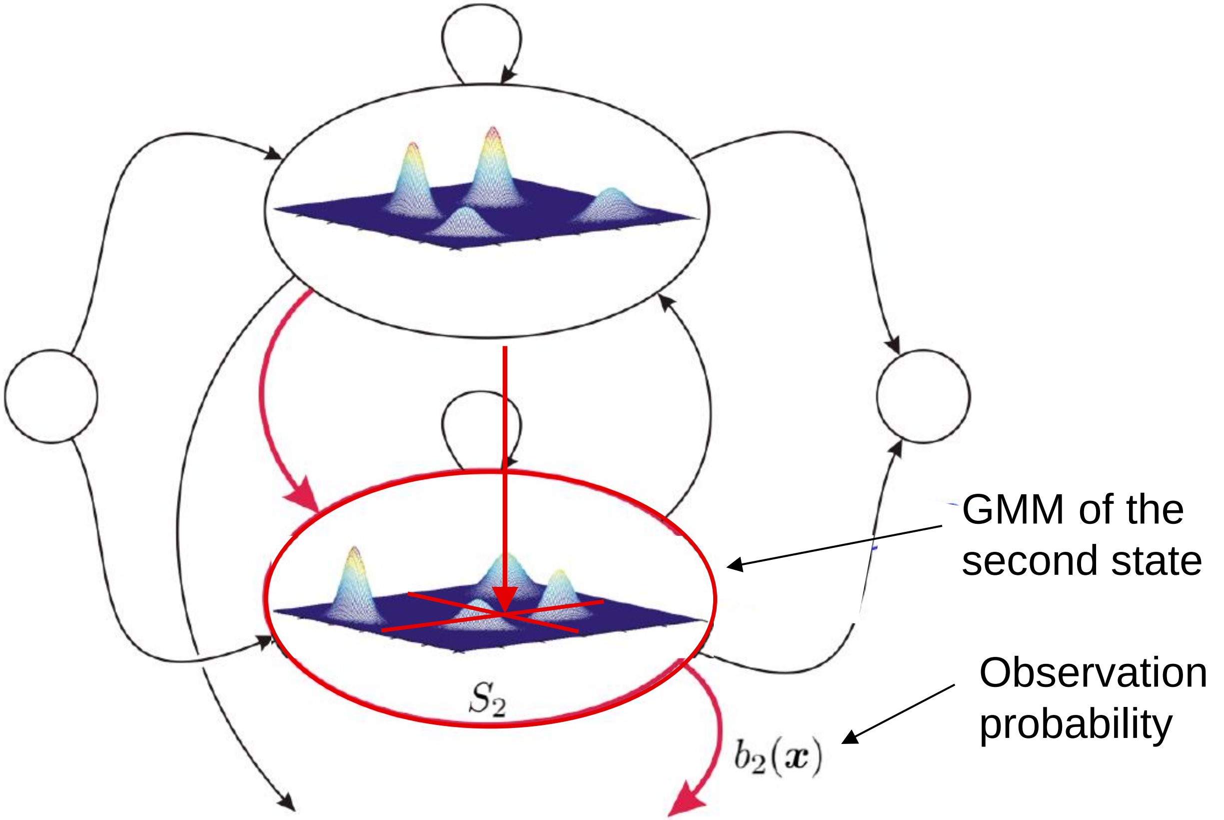
- Determine third state transition:



Sequence observed so far: $\boldsymbol{X} = [\boldsymbol{x}(0), \boldsymbol{x}(1), \ldots$

Example for the generation of observation vectors

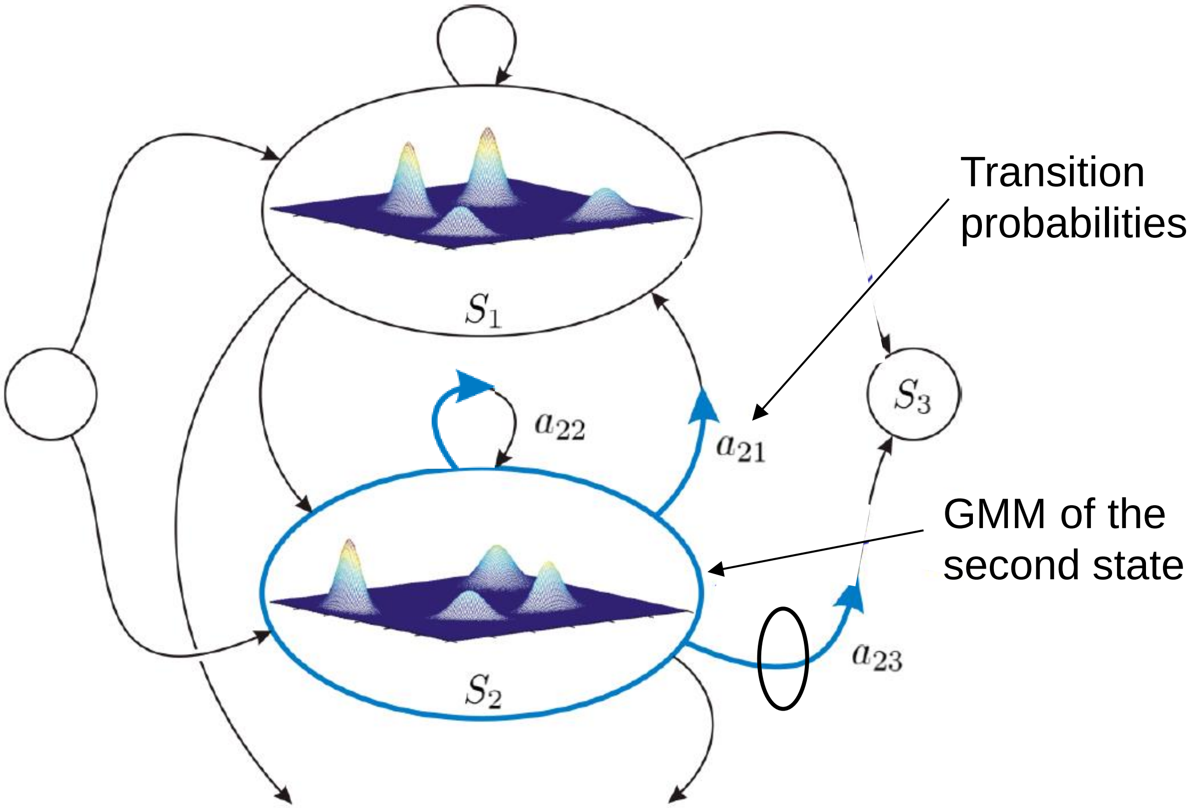
- Generation of a third observation:



Sequence observed so far: $\boldsymbol{X} = [\boldsymbol{x}(0),\, \boldsymbol{x}(1),\, \boldsymbol{x}(2),\, \ldots$

Example for the generation of observation vectors

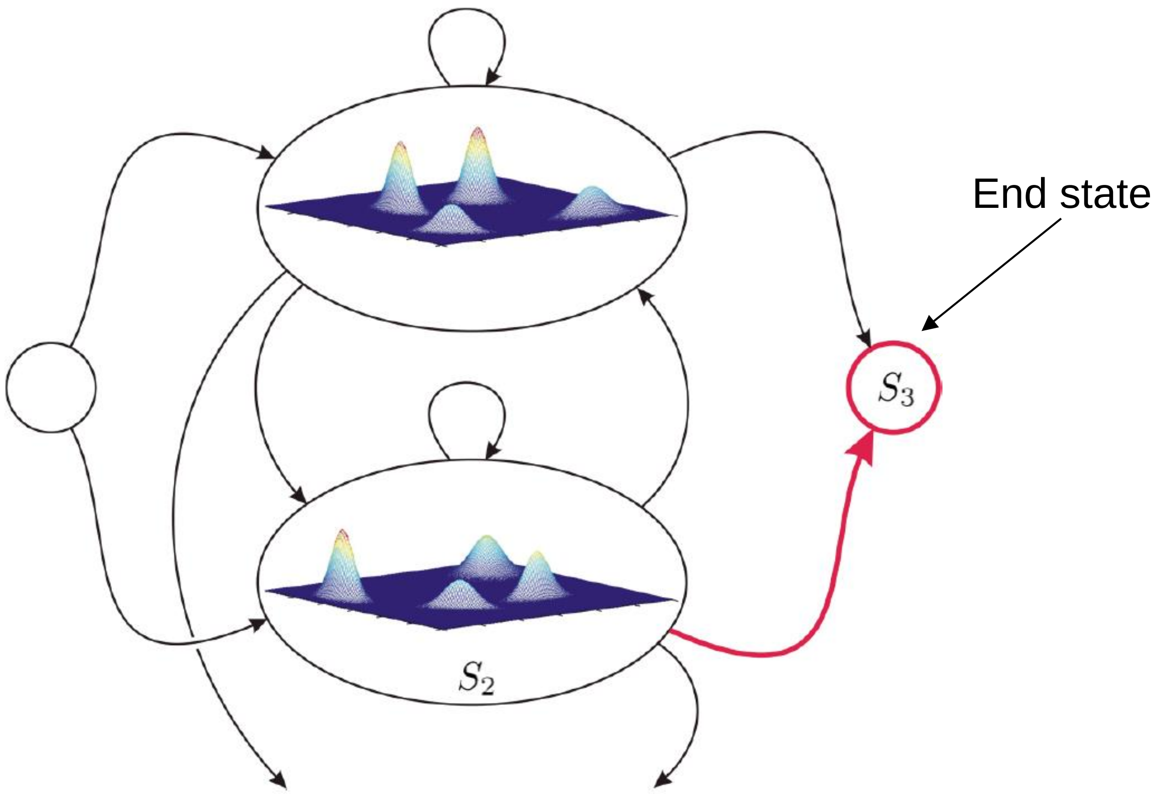
- Determine fourth state transition:



Sequence observed so far: $\boldsymbol{X} = [\boldsymbol{x}(0), \boldsymbol{x}(1), \boldsymbol{x}(2), \ldots$

Example for the generation of observation vectors

- Final state:



Final observed sequence: $\boldsymbol{X} = [\boldsymbol{x}(0), \boldsymbol{x}(1), \boldsymbol{x}(2)]$

The three basic problems of HMMs

- **Evaluation problem:**
  - Estimate the probability $p(\boldsymbol{X}|\lambda)$ that a hidden Markov model has generated an observed sequence $\boldsymbol{X}$.
  - **Example:** The observation sequence is the MFCC sequence of a spoken word. => Estimate the probability that the word model λ fits to the spoken word.
  - The Markov model parameters $a_{i,j}$ and $b_j(\boldsymbol{x}(n))$ are combined by λ.
- **Decoding problem:**
  - Estimate the „correct“ (most probable) hidden state sequence:

$$\hat{\boldsymbol{q}} \;=\; [S_0,\, \hat{q}(0),\, \hat{q}(1),\, \ldots,\, \hat{q}(T-1),\, S_{N-1}]^{\mathrm{T}}$$

given the observed sequence $\boldsymbol{X}$.

- **Model parameter estimation:**
  - Adjustment or training of the hidden Markov models based on training data.

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Evaluation problem:**
  - Estimate the probability $p(\boldsymbol{X}|\lambda)$ that a hidden Markov model has generated an observed sequence $\boldsymbol{X}$.
  - This probability can be determined by summing the observation probabilities of all possible observation sequences:

$$p(\boldsymbol{X}|\lambda) = \sum_{\boldsymbol{q}_i \in \boldsymbol{Q}} p(\boldsymbol{X}, \boldsymbol{q}_i|\lambda)$$

  - The probability can be written as:

Probability for an observed sequence $\boldsymbol{X}$ given the path $\boldsymbol{q}_i$.

Probability for the path $\boldsymbol{q}_i$.

$$p(\boldsymbol{X}|\lambda) = \sum_{\boldsymbol{q}_i \in \boldsymbol{Q}} p(\boldsymbol{X}|\boldsymbol{q}_i, \lambda)\, p(\boldsymbol{q}_i|\lambda)$$

  - The evaluation procedure described in the following slides will determine the two conditional probabilities within the sum separately.

- **Evaluation problem:**
  - First the observation probability is determined, assuming the state sequence $\boldsymbol{q}_i$ to be known. One profits from the HMM property that the observation $\boldsymbol{x}(n)$ only depends on the current state:

$$
\begin{aligned}
p(\boldsymbol{X}|\boldsymbol{q}_i,\lambda) &= \prod_{n=0}^{T-1} p(\boldsymbol{x}(n)|q_i(n),\lambda) \\
&= \prod_{n=0}^{T-1} b_{q_i(n)}(\boldsymbol{x}(n))
\end{aligned}
$$

  - The probability of the sequence $\boldsymbol{q}_i$ can be noted as following:

$$
\begin{aligned}
p(\boldsymbol{q}_i|\lambda) &= p([S_0,\, q_i(0),\, q_i(1),\, \ldots,\, q_i(T-1),\, S_{N-1}]\,|\,\lambda) \\
&= a_{S_0,q_i(0)}\, a_{q_i(0),q_i(1)}\, \cdots\, a_{q_i(T-2),q_i(T-1)}\, a_{q_i(T-1),S_{N-1}}
\end{aligned}
$$

Evaluation problem

- **Evaluation problem:**
  - This results in the following observation probability:

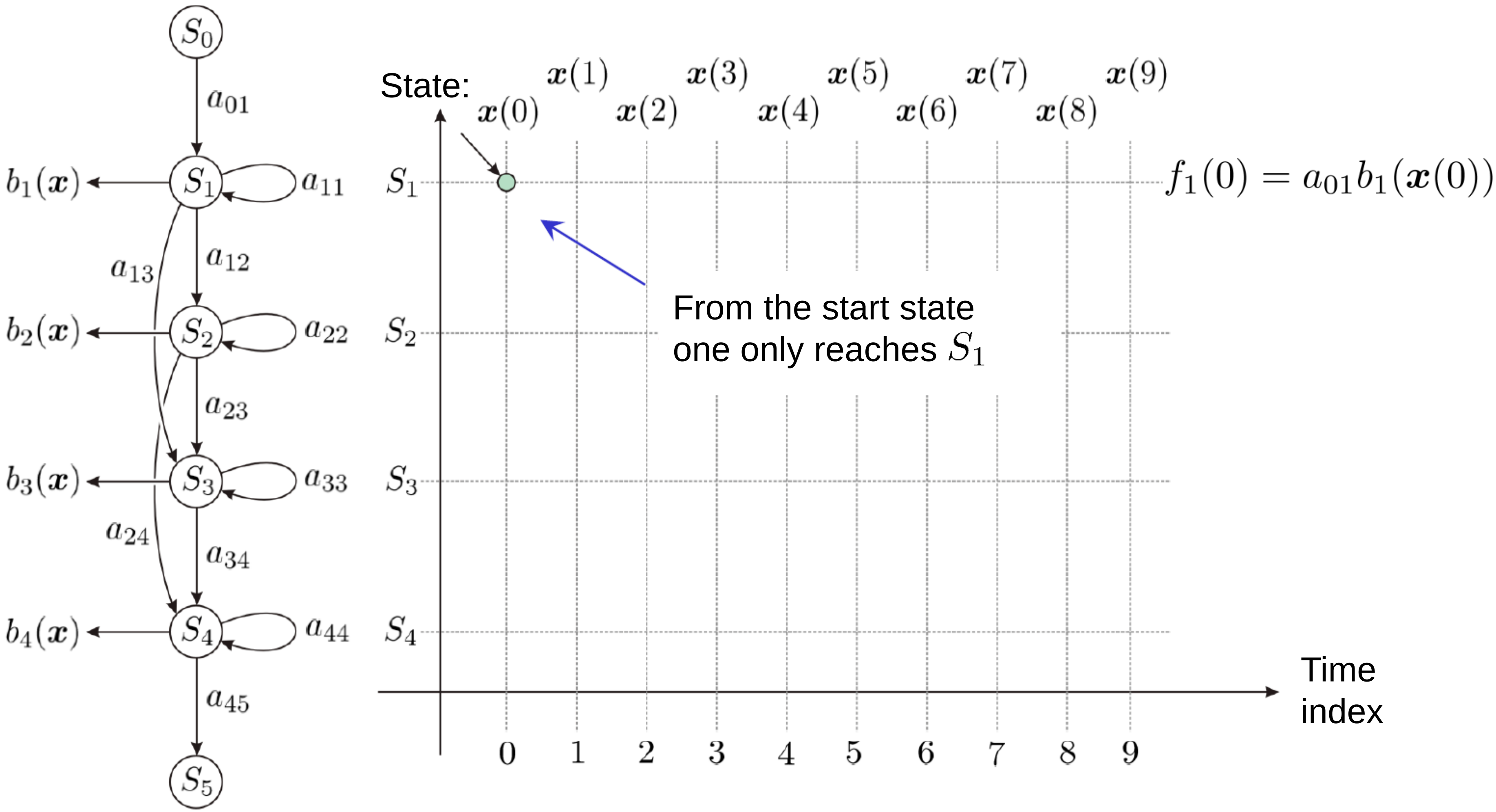
$$p(\boldsymbol{X}|\lambda) = \sum_{\boldsymbol{q}_i \in \boldsymbol{Q}} a_{S_0,q_i(0)}\, b_{q_i(0)}(\boldsymbol{x}(0))\ a_{q_i(0),q_i(1)}\, b_{q_i(1)}(\boldsymbol{x}(1))\ \ldots$$

  - The problem of this procedure is that there are for the general HMM $N-2$ possible states at each time index resulting in $(N-2)^2$ possible paths => impossible to handle.
  - An alternative is the **Forward algorithm**. Here the so-called *forward probability* is calculated:

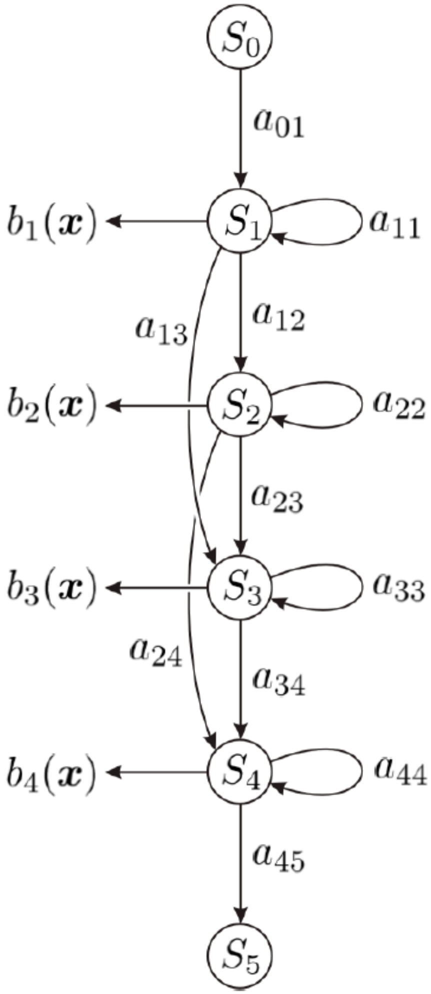
$$f_i(n) = p(\boldsymbol{X}^{(n)},\, q(n) = S_i|\lambda) \qquad \text{with: } \boldsymbol{X}^{(n)} = [\boldsymbol{x}(0),\, \boldsymbol{x}(1),\, \ldots,\, \boldsymbol{x}(n)]$$

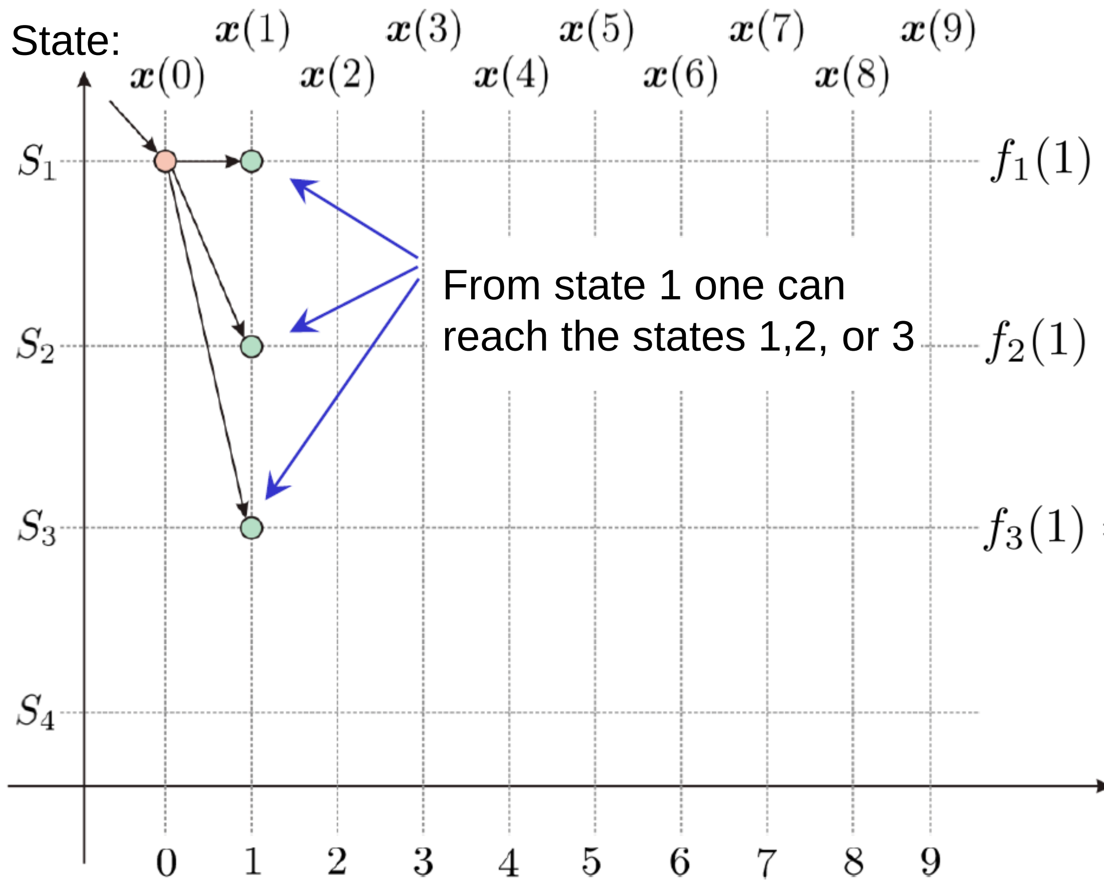
    which defines the probability to be at the time n in the state S_i and having so far observed the sequence $\boldsymbol{X}^{(n)}$.
  - In order to understand its calculation, we will analyse “trellis diagrams“.

Trellis diagram for a specific HMM



Trellis diagram for a specific HMM



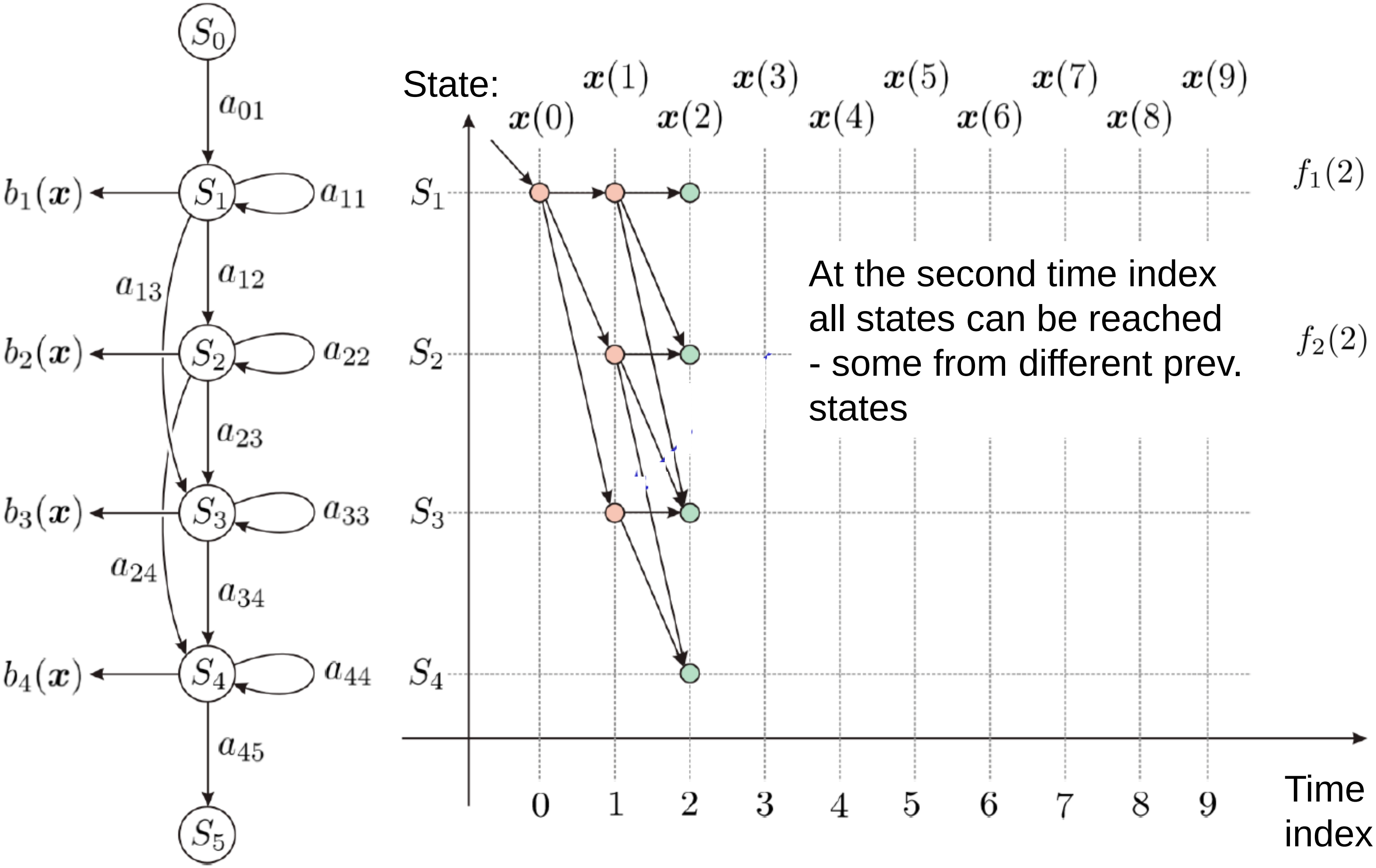


$$f_1(1) = a_{01} b_1(\boldsymbol{x}(0))\, a_{11} b_1(\boldsymbol{x}(1))$$

$$f_2(1) = a_{01} b_1(\boldsymbol{x}(0))\, a_{12} b_2(\boldsymbol{x}(1))$$

$$f_3(1) = a_{01} b_1(\boldsymbol{x}(0))\, a_{13} b_3(\boldsymbol{x}(1))$$

Trellis diagram for a specific HMM



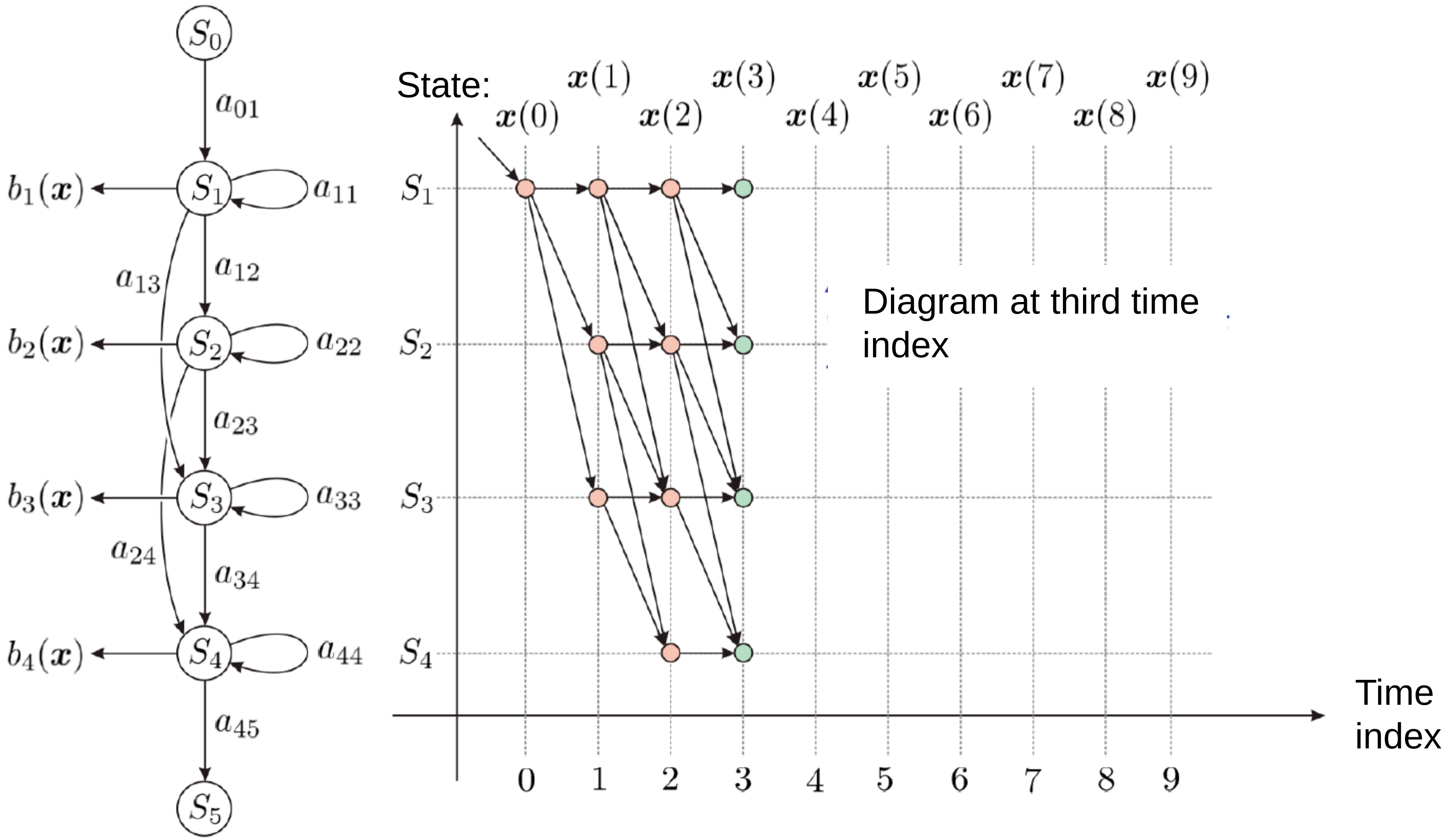
At the second time index all states can be reached - some from different prev. states

$$f_1(2) = a_{01} b_1(\boldsymbol{x}(0))\, a_{11} b_1(\boldsymbol{x}(1))\, a_{11} b_1(\boldsymbol{x}(2))$$

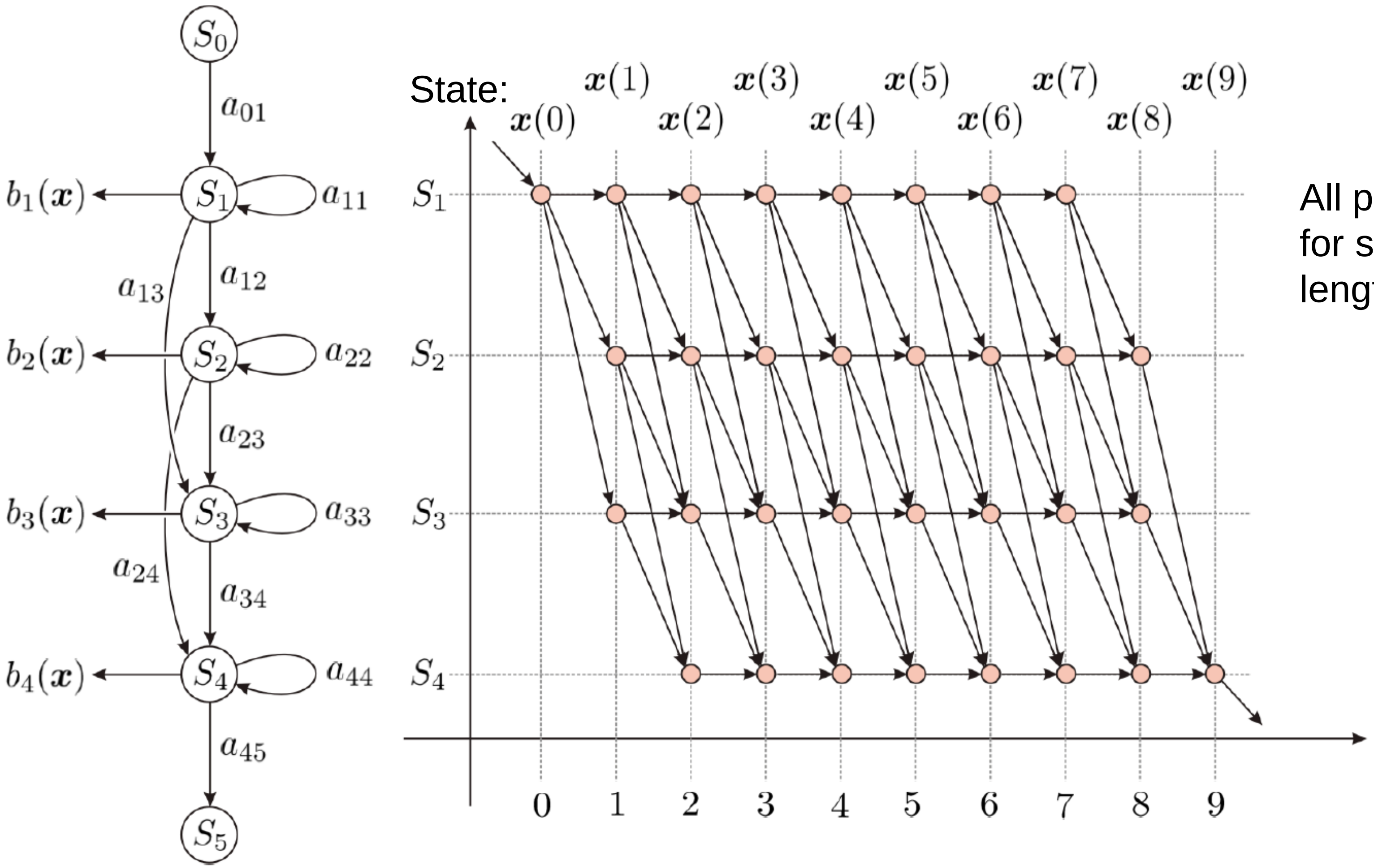
$$f_2(2) = a_{01} b_1(\boldsymbol{x}(0))\, a_{11} b_1(\boldsymbol{x}(1))\, a_{12} b_2(\boldsymbol{x}(2)) + a_{01} b_1(\boldsymbol{x}(0))\, a_{12} b_2(\boldsymbol{x}(1))\, a_{22} b_2(\boldsymbol{x}(2))$$

$$\vdots$$

Trellis diagram for a specific HMM



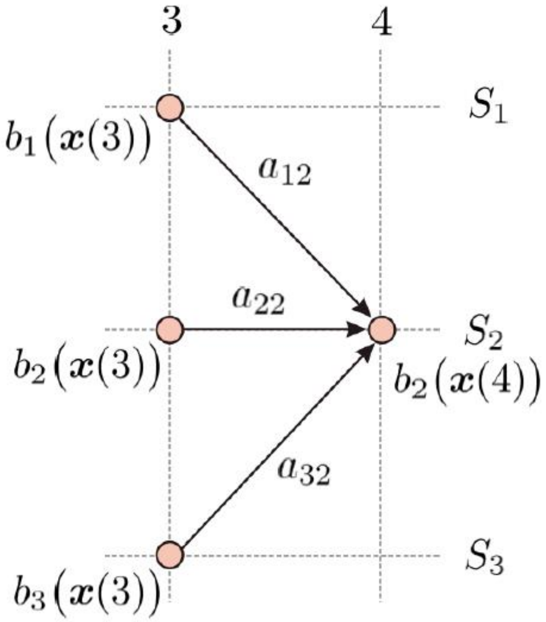
Trellis diagram for a specific HMM



All possible paths for sequences of length 10

Trellis diagram for a specific HMM

- **Signification of nodes and edges:**
  - The transition probabilities are typically noted at the edges.
  - The observation probabilities are noted at the nodes.



The “Forward algorithm”

- **Calculation of the forward probability:**
  - The forward probability can be calculated by summing the probabilities for each path arriving at one state:

$$f_i(n) = p(\boldsymbol{X}^{(n)},\, q(n) = S_i|\lambda) = \sum_{\boldsymbol{q}_j^{(n)} \text{ with } q_j(n)=S_i} p(\boldsymbol{X}^{(n)},\, \boldsymbol{q}_j^{(n)}|\lambda)$$

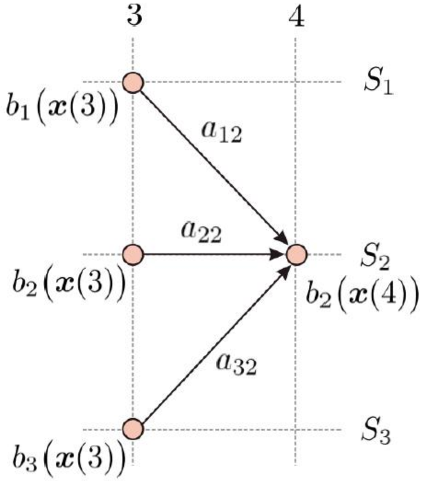
with: $$\boldsymbol{X}^{(n)} = [\boldsymbol{x}(0),\, \boldsymbol{x}(1),\, \ldots,\, \boldsymbol{x}(n)]$$

$$\boldsymbol{q}_i^{(n)} = [q_i(0),\, q_i(1),\, \ldots,\, q_i(n)]^{\mathrm{T}}$$

  - The forward probability can be calculated recursively:

$$f_i(n) = \left[\sum_{j=1}^{N-2} f_j(n-1)\, a_{j,i}\right] b_i(\boldsymbol{x}(n))$$

Example:

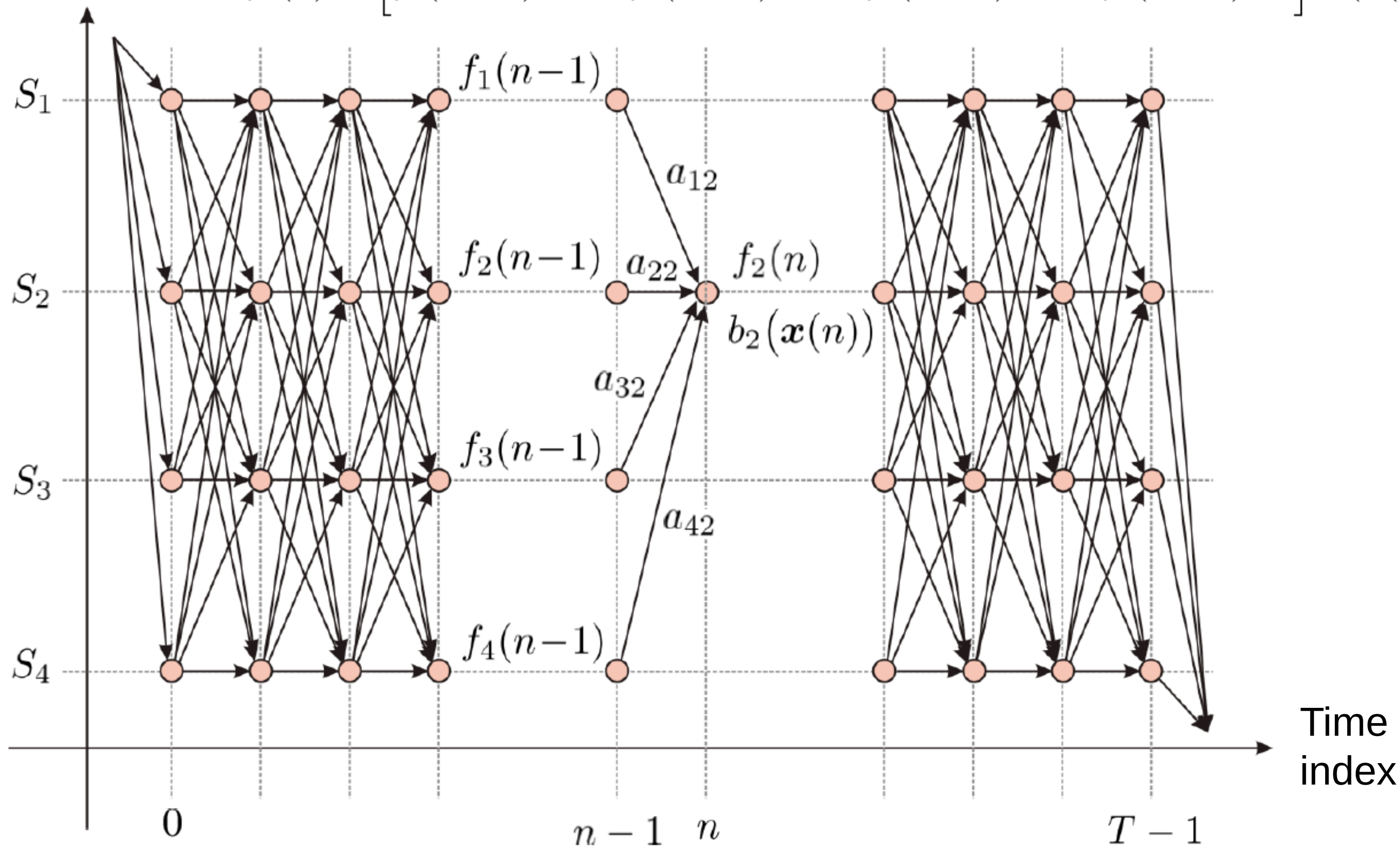


$$f_2(4) = [f_1(3)a_{12} + f_2(3)a_{22} + f_3(3)a_{32}]\, b_2(\boldsymbol{x}(4))$$

The "Forward algorithm"

- **Graph explaining the forward algorithm:**

State: $f_2(n) = \left[f_1(n-1)a_{12} + f_2(n-1)a_{22} + f_3(n-1)a_{32} + f_4(n-1)a_{42}\right] b_2(\boldsymbol{x}(n))$



The “Forward algorithm” - overview

- **Evaluation problem:**
  - Estimate the probability $p(\boldsymbol{X}|\lambda)$ that a hidden Markov model has generated an observed sequence $\boldsymbol{X}$.

- The observation probability is calculated with the **forward algorithm**:
  - Initialization:

$$f_i(0) = a_{0,i}\, b_i(\boldsymbol{x}(0))$$

  - Recursion:

$$f_i(n) = \left[\sum_{j=1}^{N-2} f_j(n-1)\, a_{j,i}\right] b_i(\boldsymbol{x}(n))$$

  - Termination:

$$p(\boldsymbol{X}|\lambda) = \sum_{j=1}^{N-2} f_j(T-1)\, a_{j,N-1}$$

Result of the evaluation problem:

Probability that the corresponding HMM has generated the observed sequence: $p(\boldsymbol{X}|\lambda)$

$\Rightarrow$ Several HMMs (corresponding to different phonemes / words) are evaluated to detect the most probable one.

The three basic problems of HMMs

- **Evaluation problem:**
  - Estimate the probability $p(\boldsymbol{X}|\lambda)$ that a hidden Markov model has generated an observed sequence $\boldsymbol{X}$.
  - The Markov model parameters $a_{i,j}$ and $b_j(\boldsymbol{x}(n))$ are combined by λ.

Done

- **Decoding problem:**
  - Estimate the „correct“, i.e., most probable, hidden state sequence:

$$\hat{\boldsymbol{q}} = [S_0, \hat{q}(1), \hat{q}(2), \ldots, \hat{q}(T-2), S_{N-1}]^{\mathrm{T}}$$

given the observed sequence $\boldsymbol{X}$.

- **Model parameter estimation:**
  - Adjustment or training of the hidden Markov models based on training data

- **Decoding problem:**
  - Estimate the „correct“, i.e. most probable, hidden state sequence:

    $$\hat{\boldsymbol{q}} \;=\; [S_0,\, \hat{q}(1),\, \hat{q}(2),\, \ldots,\, \hat{q}(T-2),\, S_{N-1}]^{\mathrm{T}}$$

    given the observed sequence $\boldsymbol{X}$.

  - Formally this can be written by:

    $$\hat{\boldsymbol{q}} \;=\; \arg\max_{\boldsymbol{q}_j} \left\{ p(\boldsymbol{q}_j|\boldsymbol{X},\lambda) \right\}$$

    using: $$p(\boldsymbol{q}_j|\boldsymbol{X},\lambda) = \frac{p(\boldsymbol{q}_j,\boldsymbol{X}|\lambda)}{p(\boldsymbol{X}|\lambda)}$$

  - Since $p(\boldsymbol{X}|\lambda)$ does not depend on the state sequence $\boldsymbol{q}$, one obtains:

    $$\hat{\boldsymbol{q}} \;=\; \arg\max_{\boldsymbol{q}_j} \left\{ p(\boldsymbol{q}_j,\boldsymbol{X}|\lambda) \right\}$$

    with: $$p(\boldsymbol{q}_j,\boldsymbol{X}|\lambda) = p(\boldsymbol{X}|\boldsymbol{q}_i,\lambda)\, p(\boldsymbol{q}_i|\lambda)$$

    This task can be solved by comparable methods as used for the evaluation problem.

Decoding problem

- An efficient calculation of the most probable sequence can be performed with the **Viterbi algorithm** searching for the **path with the maximum probability**:

$$v_i(n) = \max_{\boldsymbol{q}_j^{(n)} \text{ with } q_j(n)=S_i} \left\{ p(\boldsymbol{X}^{(n)},\, \boldsymbol{q}_j^{(n)} | \lambda) \right\}$$

Iterative procedure:

$$v_i(n) = \max_{j=1\ldots N-2} \left\{ v_j(n-1)\, a_{j,i} \right\} b_i(\boldsymbol{x}(n))$$

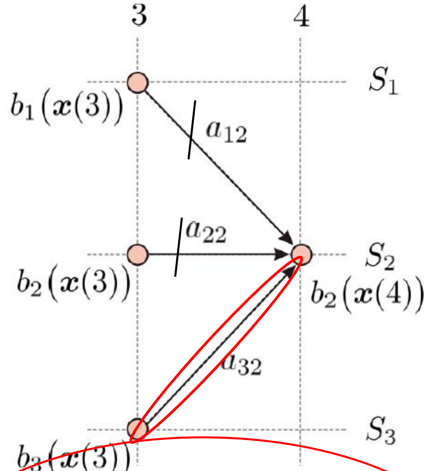
- Instead of summing the probabilities for each path as is done for the **forward algorithm**

$$f_i(n) = \sum_{\boldsymbol{q}_j^{(n)} \text{ with } q_j(n)=S_i} p(\boldsymbol{X}^{(n)},\, \boldsymbol{q}_j^{(n)} | \lambda)$$

$$f_i(n) = \left[\sum_{j=1}^{N-2} f_j(n-1)\, a_{j,i} \right] b_i(\boldsymbol{x}(n))$$

Example:

Third path with max. probability

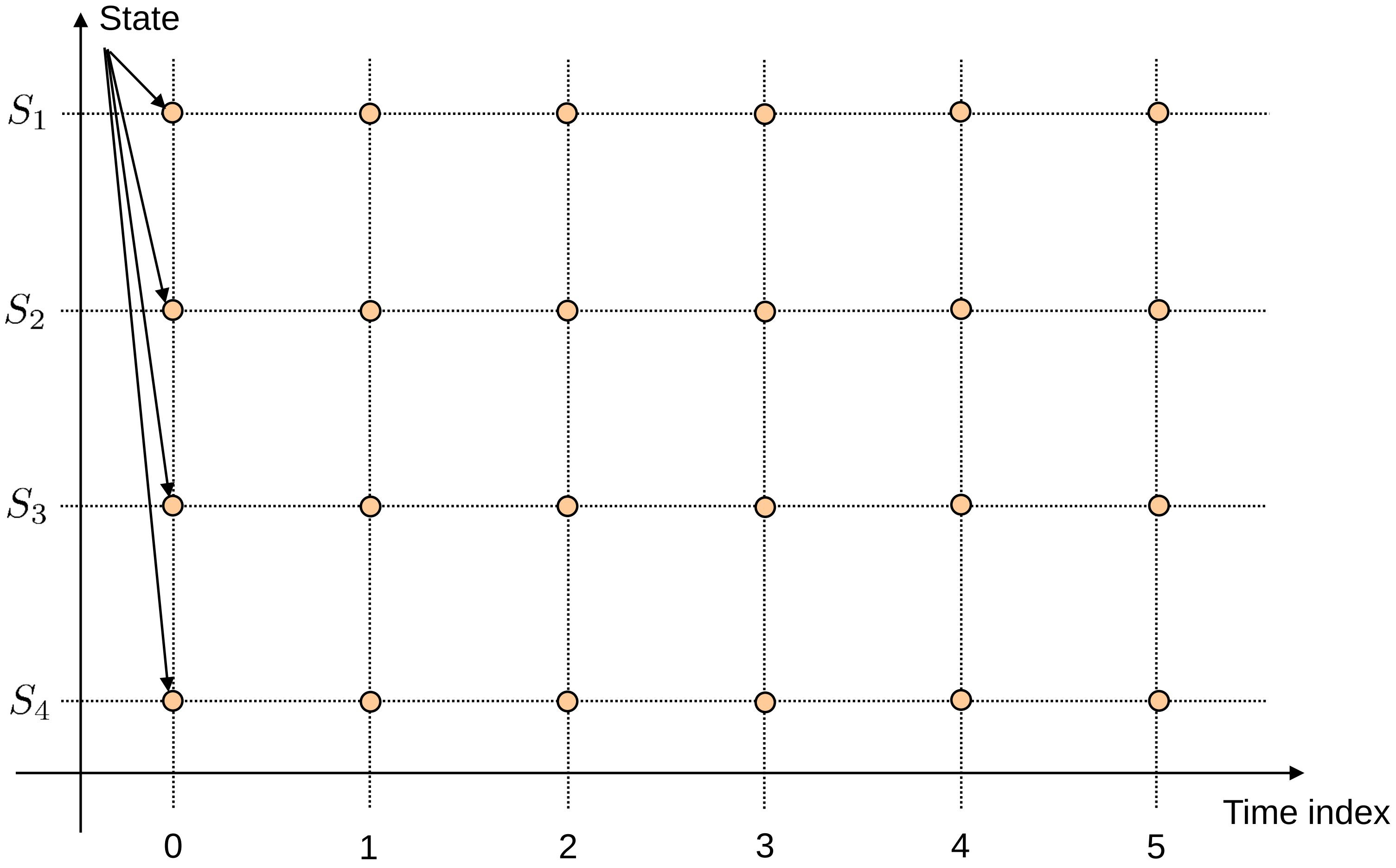


$$v_2(4) = \max[v_1(3)a_{12},\; v_2(3)a_{22},\, v_3(3)a_{32}]\; b_2(\boldsymbol{x}(4))$$

$$v_2(4) = [v_3(3)a_{32}]\, b_2(\boldsymbol{x}(4))$$

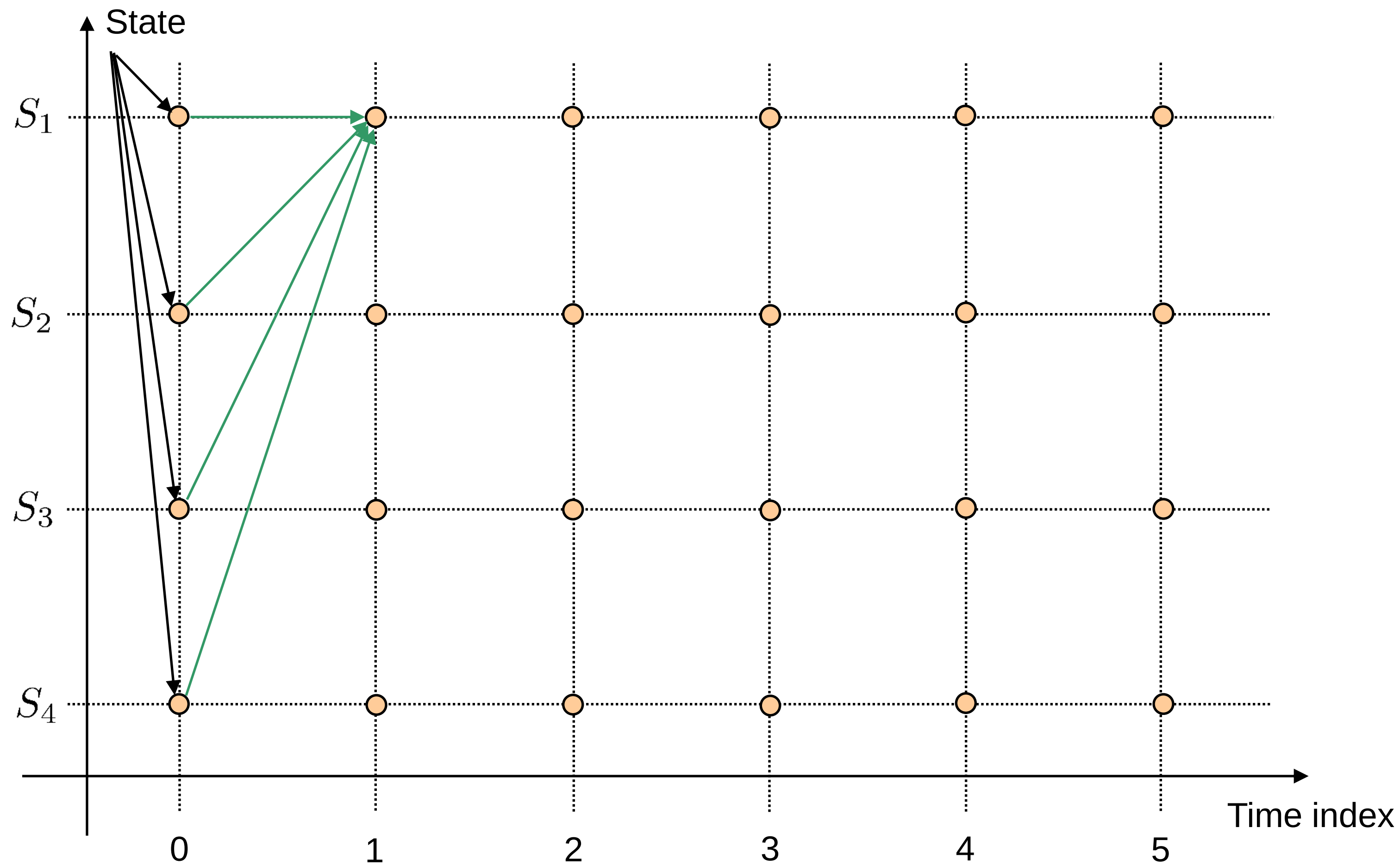
Viterbi algorithm

- Initialization:



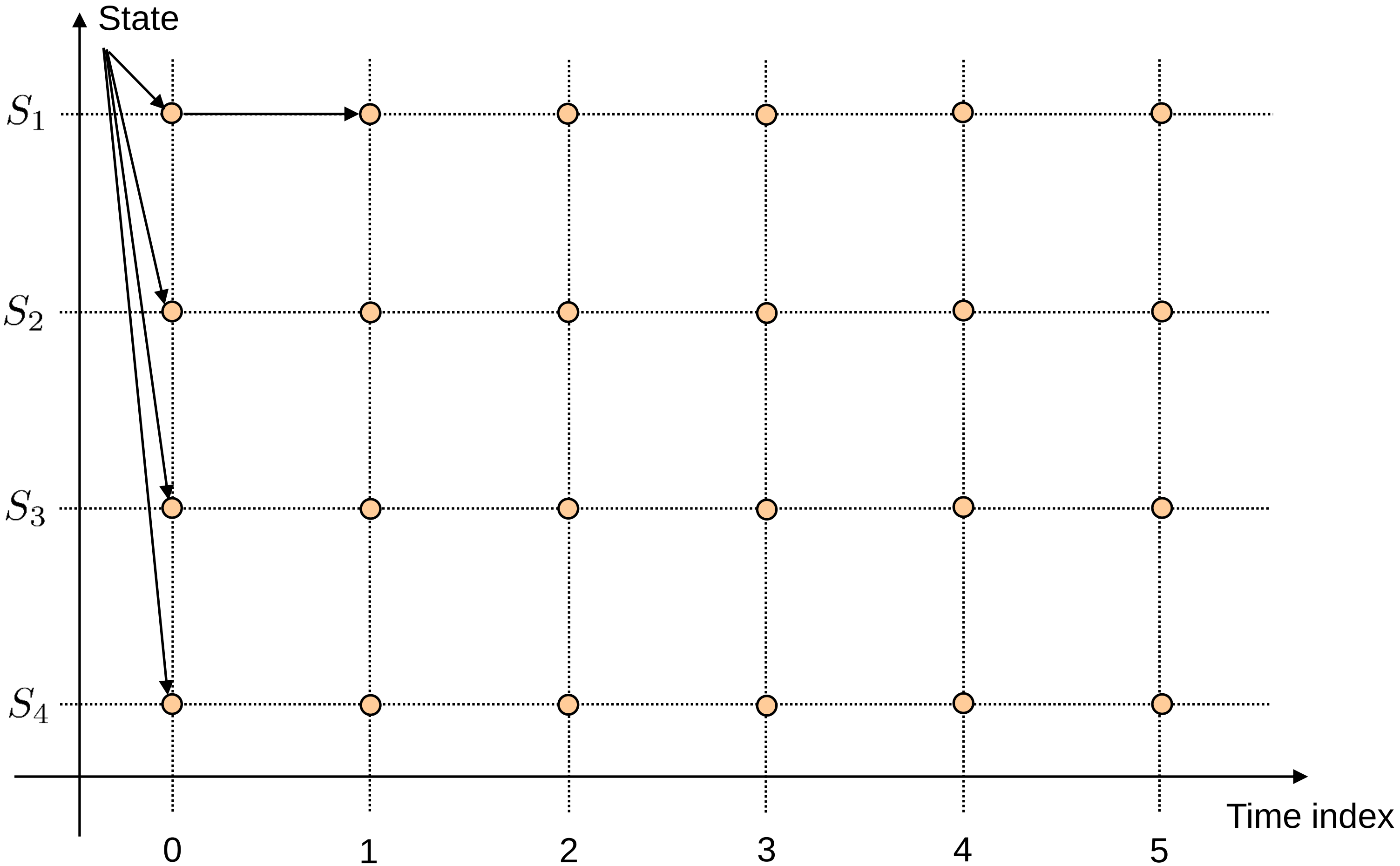
Viterbi algorithm

- Recursion for the first state:



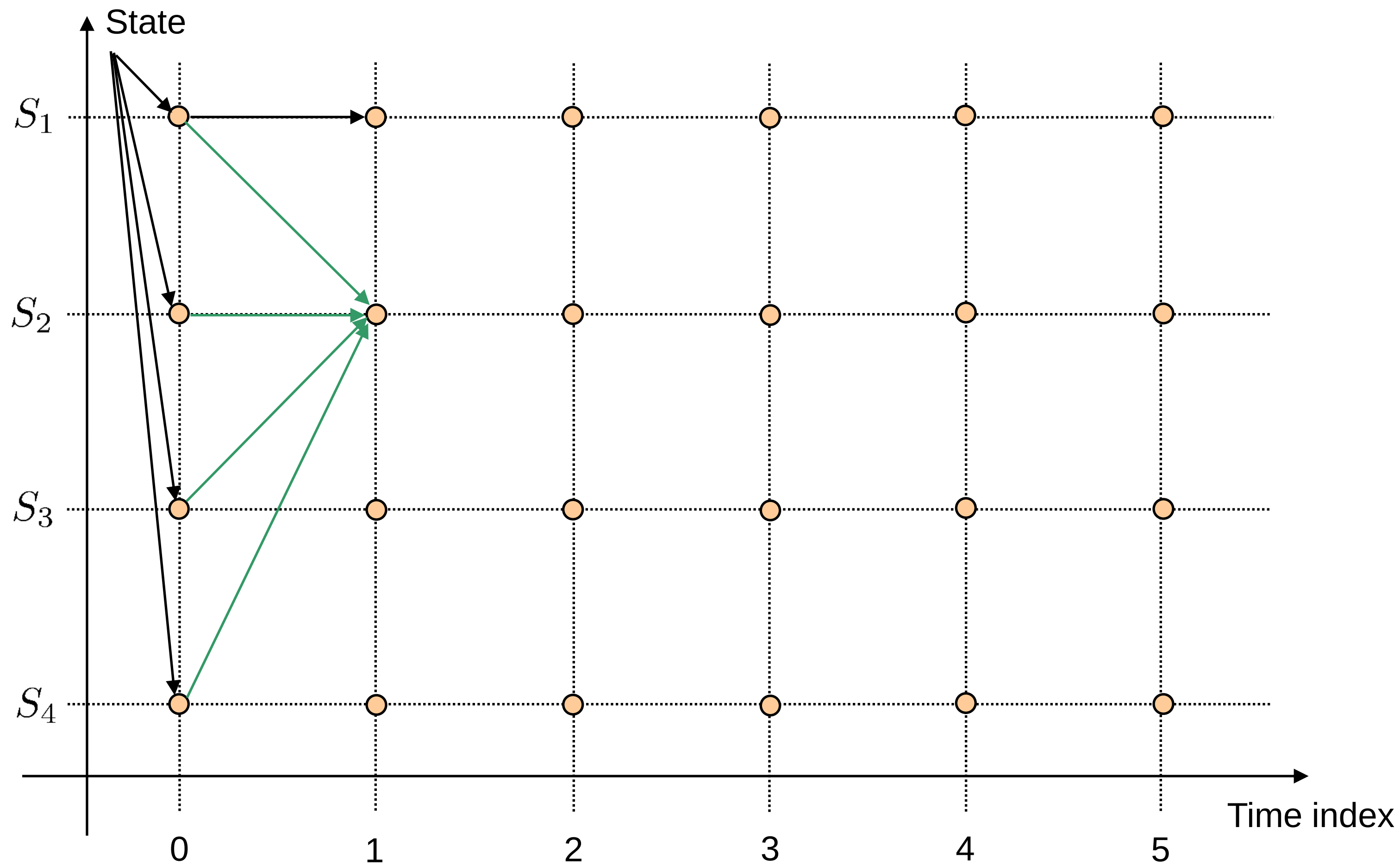
Viterbi algorithm

- Recursion for the first state:



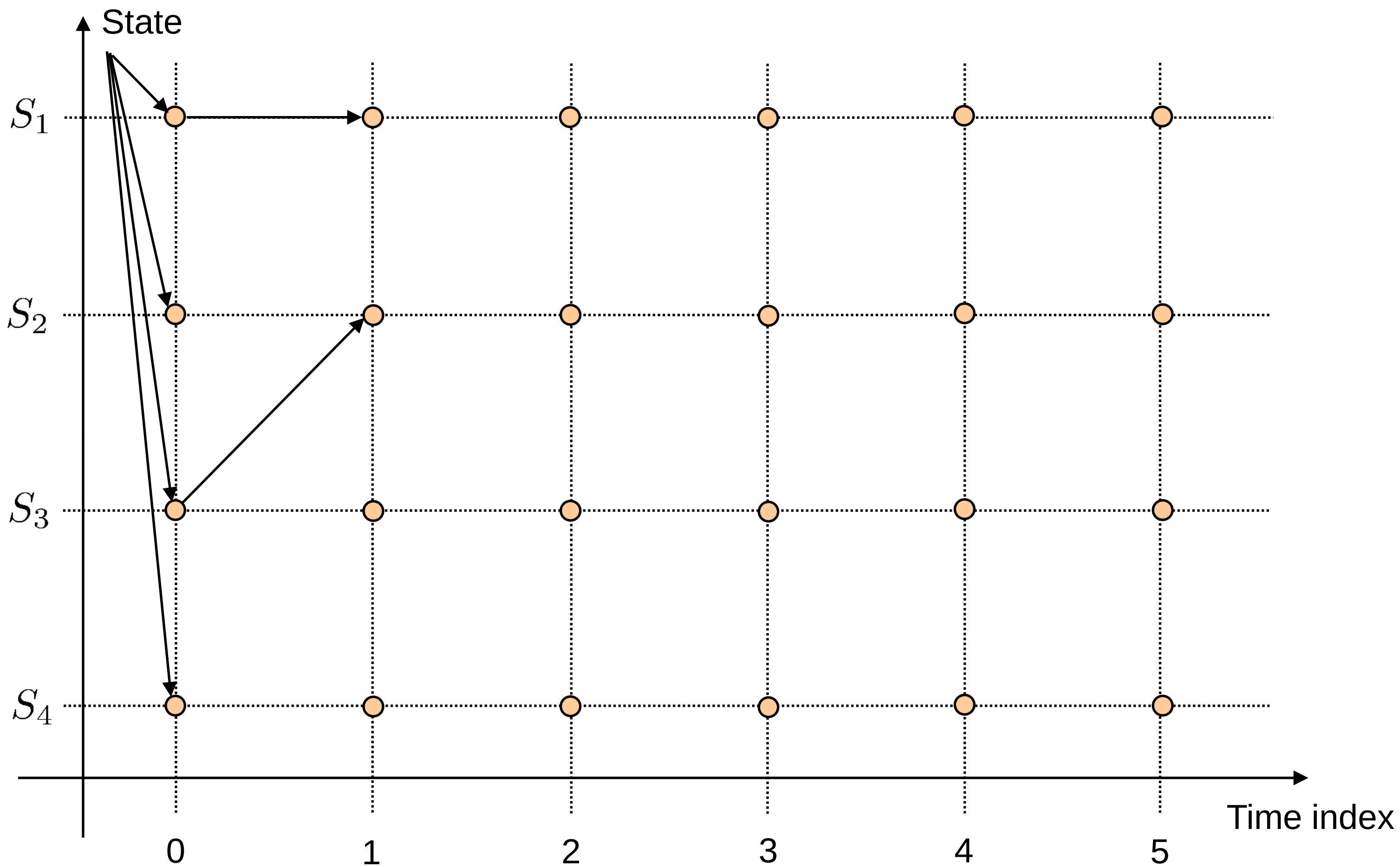
Viterbi algorithm

- Recursion for the second state:



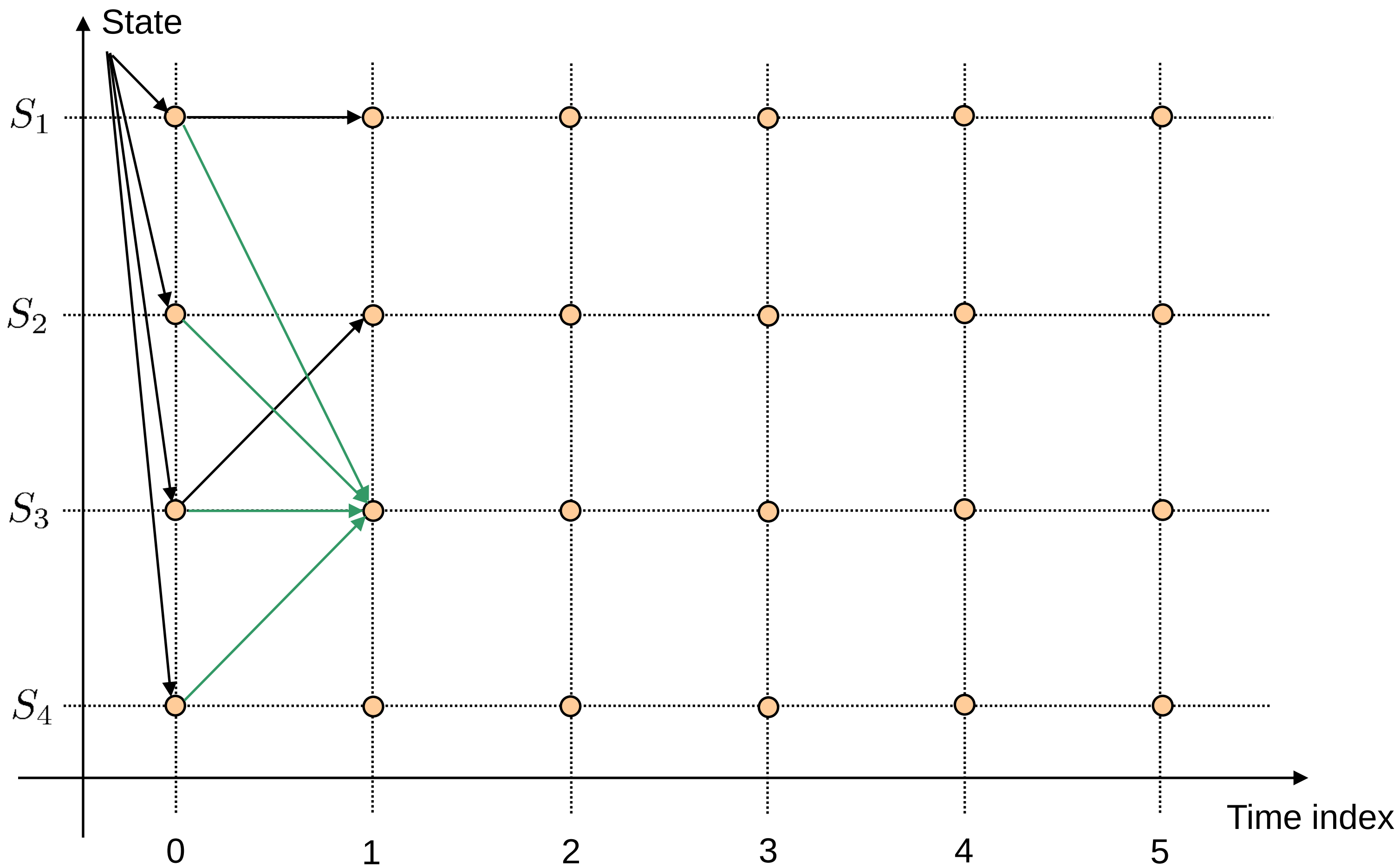
Viterbi algorithm

- Recursion for the second state:



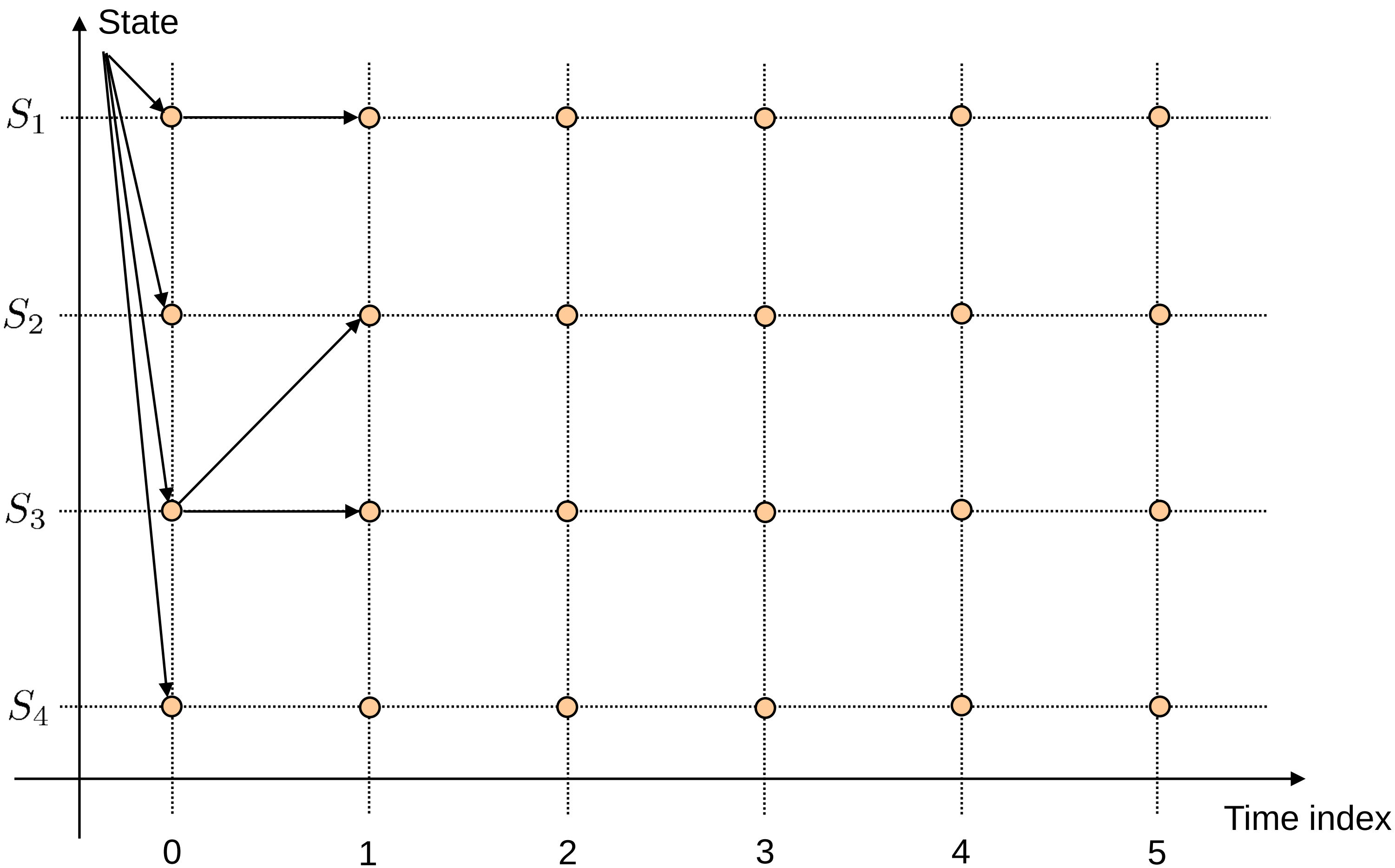
Viterbi algorithm

- Recursion for the third state:



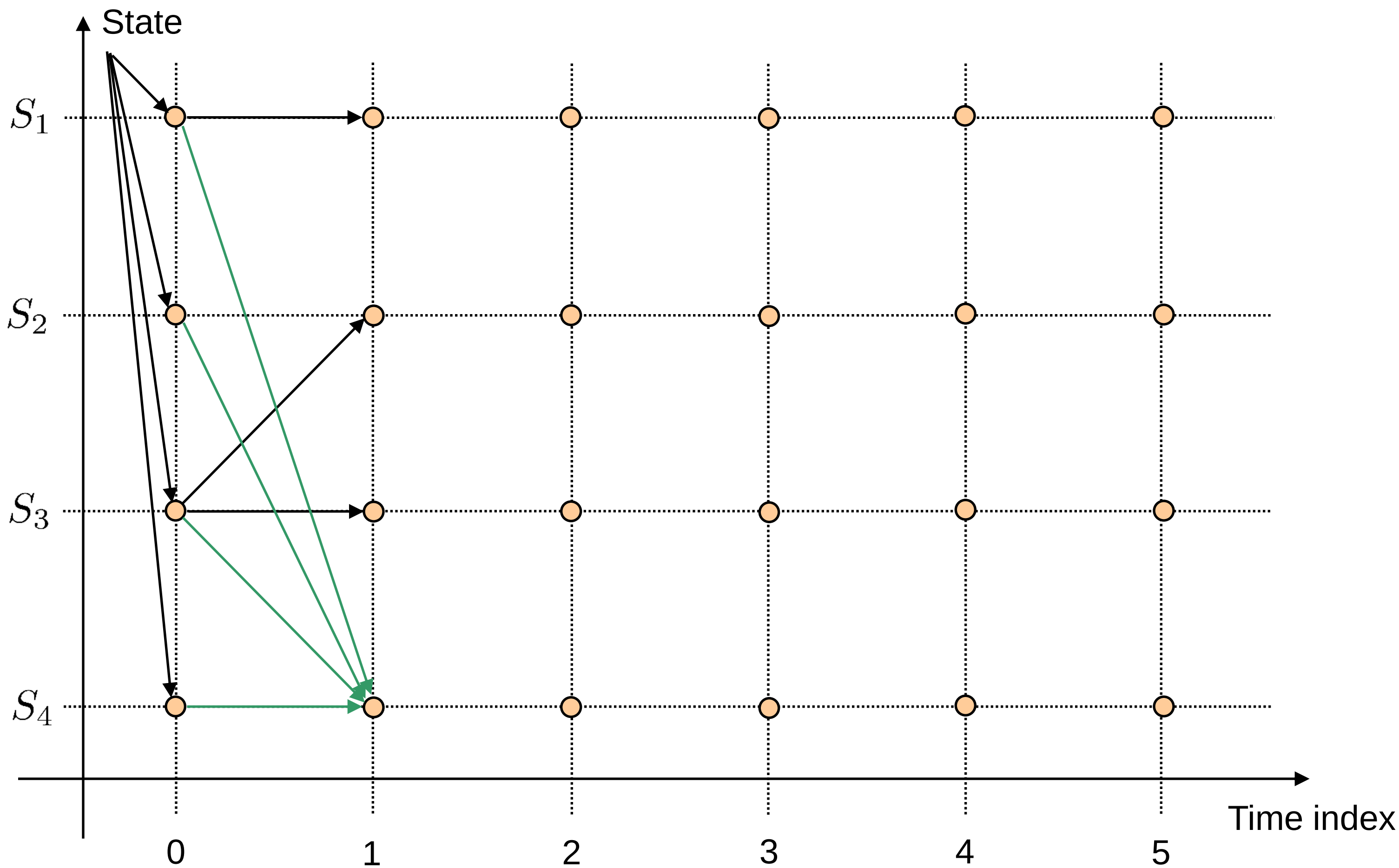
Viterbi algorithm

- Recursion for the third state:



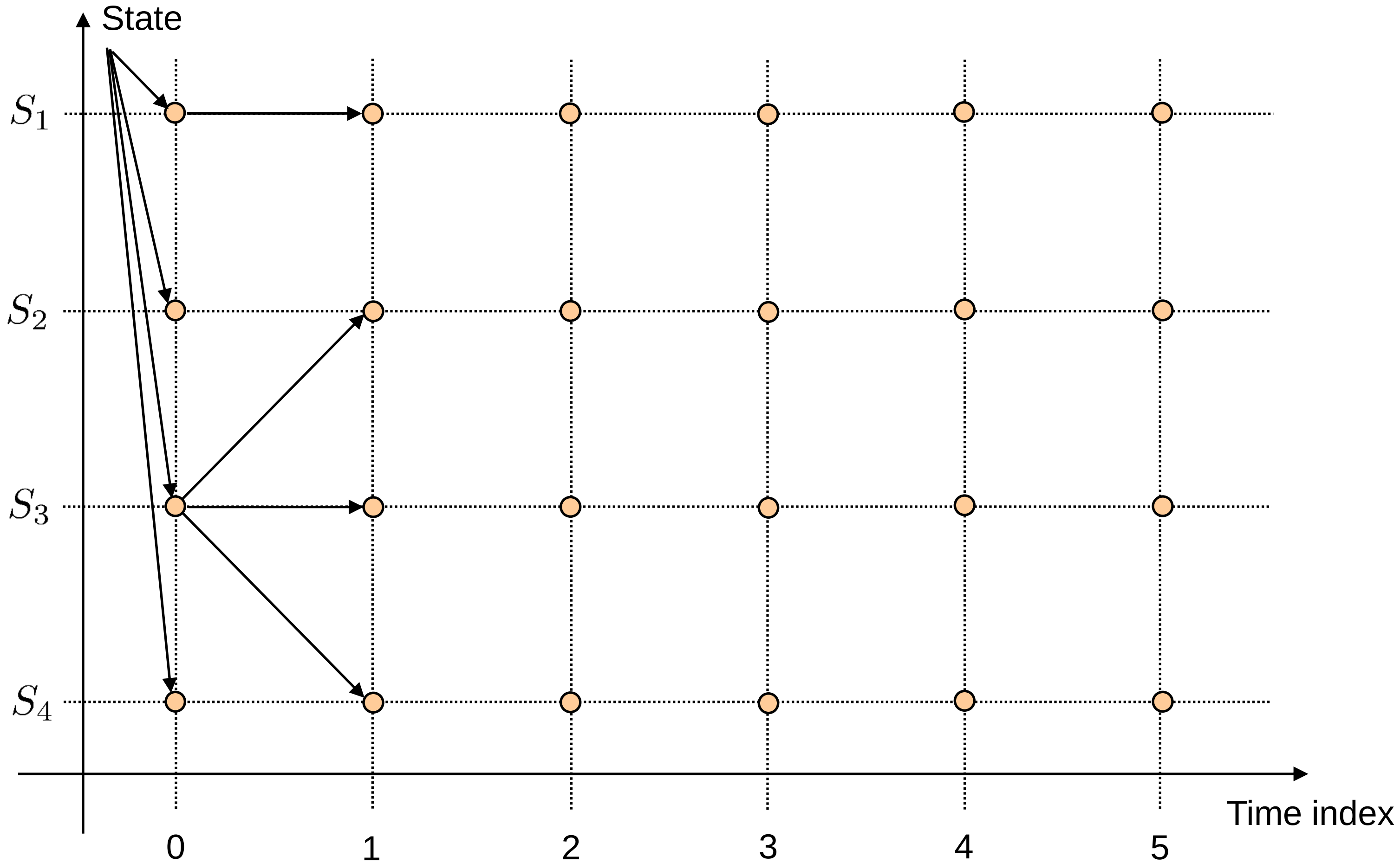
Viterbi algorithm

- Recursion for the fourth state:



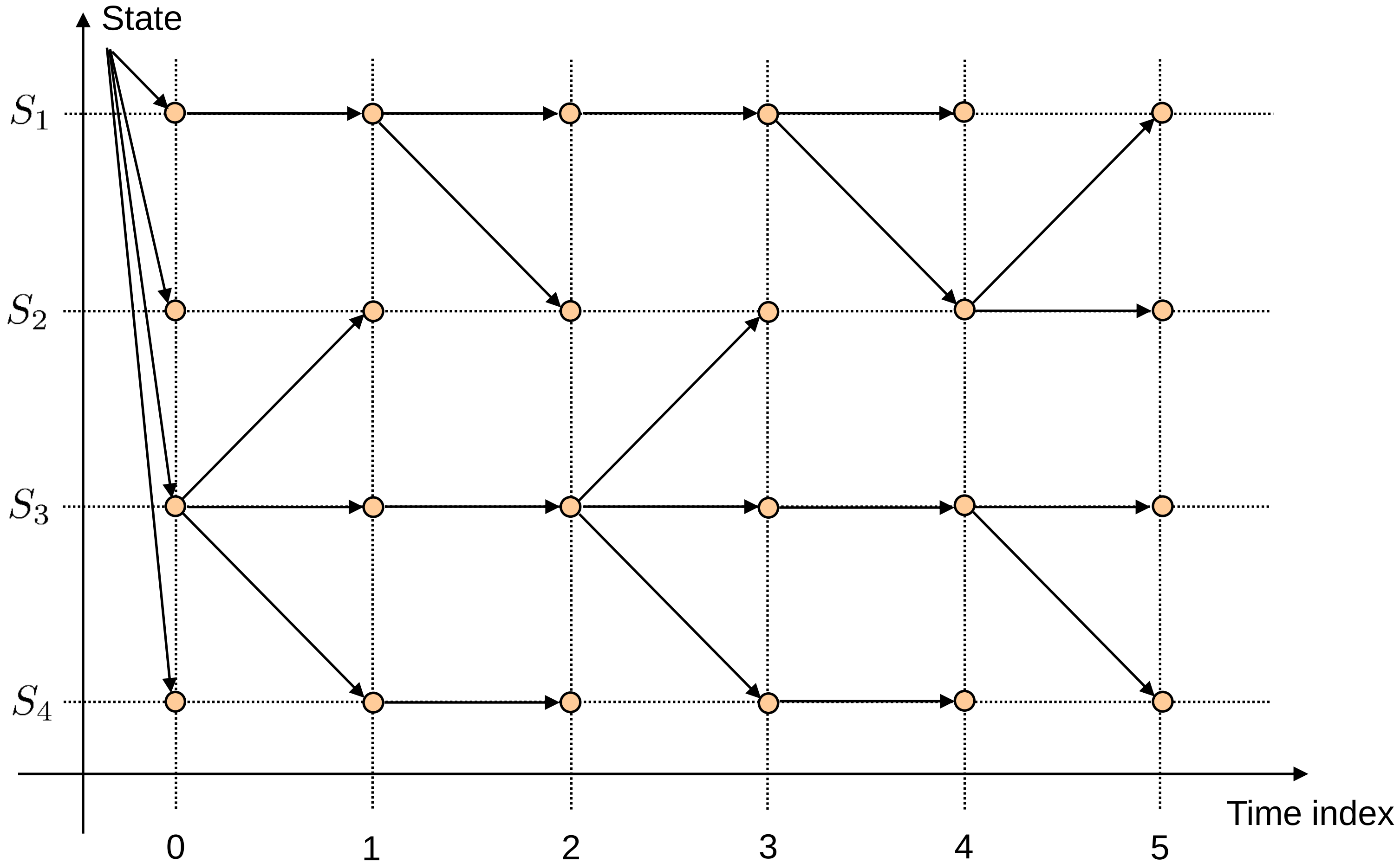
Viterbi algorithm

- Recursion for the fourth state:



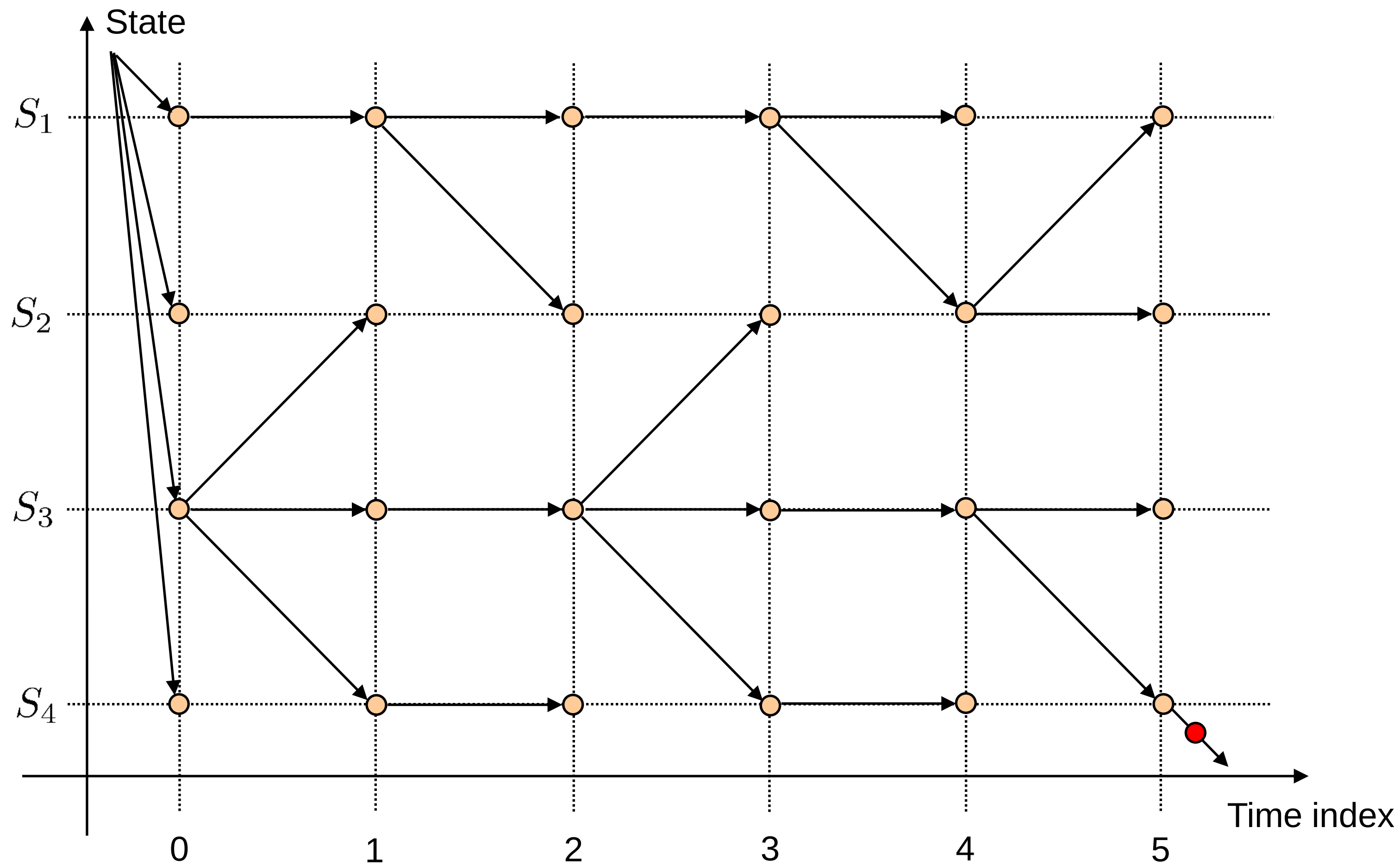
Viterbi algorithm

- Complete recursion:



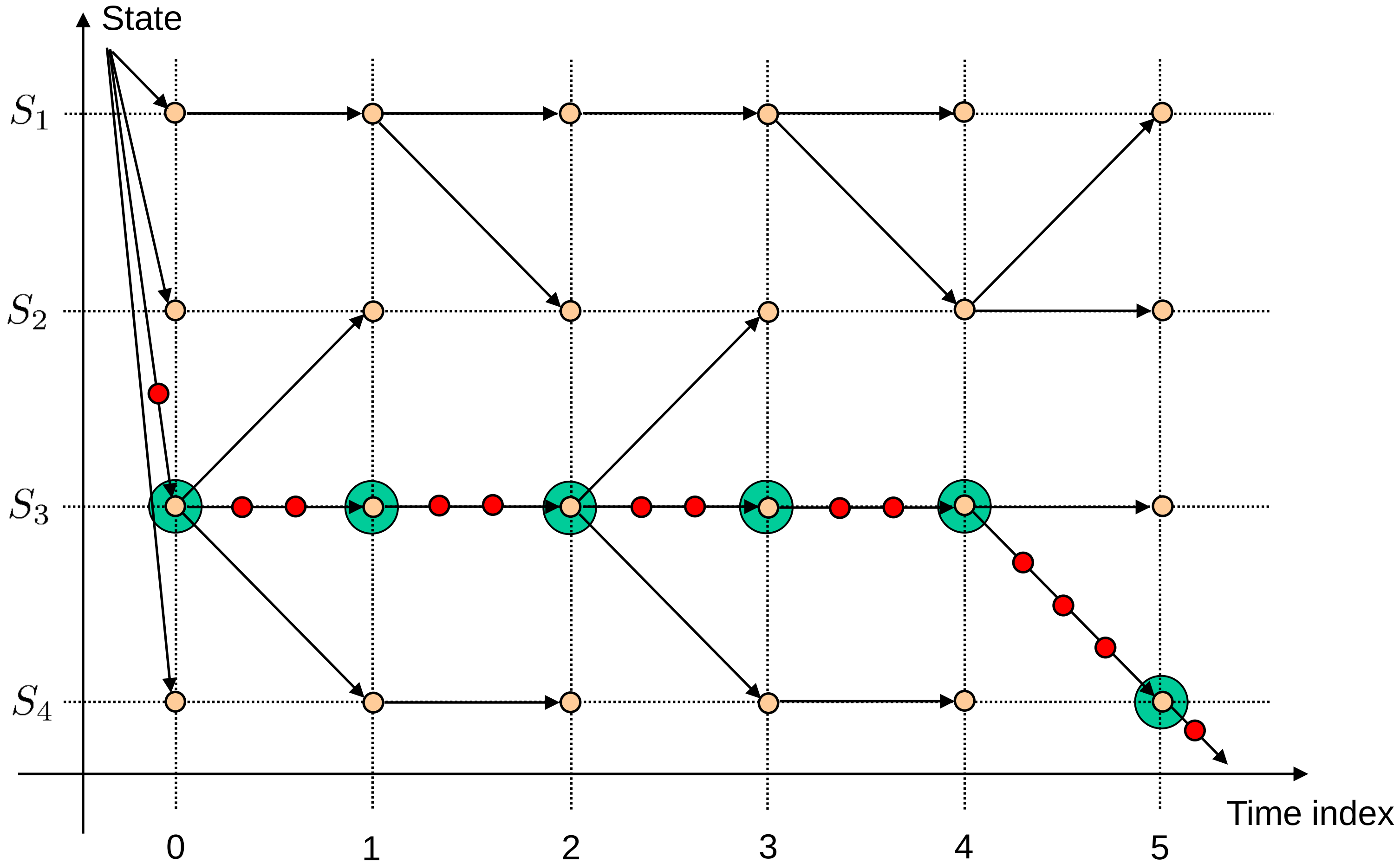
Viterbi algorithm

- Complete recursion + termination:



Viterbi algorithm

- Back-tracking of the optimal sequence:



Viterbi algorithm

- Summary of the Viterbi algorithm:
  - Initialization:

$$v_i(0) = a_{0,i}\, b_i(\boldsymbol{x}(0))$$

  - Recursion:

$$v_i(n) = \max_{j=1\ldots N-2} \left\{ v_j(n-1)\, a_{j,i} \right\} b_i(\boldsymbol{x}(n))$$

$$t_i(n) = \underset{j=1\ldots N-2}{\arg\max} \left\{ v_j(n-1)\, a_{j,i} \right\}$$

  - Termination:

$$v_{N-1}(T) = \max_{j=1\ldots N-2} \left\{ v_j(T-1)\, a_{j,N-1} \right\}$$

$$t_{N-1}(T) = \underset{j=1\ldots N-2}{\arg\max} \left\{ v_j(T-1)\, a_{j,N-1} \right\}$$

  - Back-tracking of the optimal sequence:

$$\hat{q}(n) = \begin{cases} t_{N-1}(T) & \text{if} \quad n = T \\ t_{\hat{q}(n+1)}(n+1) & \text{else} \end{cases}$$

Viterbi algorithm

- Properties of the Viterbi algorithm:

  - Only multiplications are calculated. This allows a processing in the log-domain.
    => Advantages in the scaling of the calculated values. For longer observation sequences these values become very small and have to be re-scaled regularly in the linear domain. => Can be avoided in the log-domain.

$$v_{i,\log}(n) = \max_{j=1\ldots N-2}\left\{v_{j,\log}(n-1) + a_{j,i,\log}\right\} + b_{i,\log}(\boldsymbol{x}(n))$$

  - Sometimes the Viterbi procedure is also used as a simplified version of the forward algorithm.

The three basic problems of HMMs

- **Evaluation problem:**
  - Estimate the probability $p(\boldsymbol{X}|\lambda)$ that a hidden Markov model has generated an observed sequence $\boldsymbol{X}$.
  - The Markov model parameters $a_{i,j}$ and $b_j(\boldsymbol{x}(n))$ are combined by λ.

Done

- **Decoding problem:**
  - Estimate the „correct" hidden state sequence:

$$\hat{\boldsymbol{q}} = [S_0, \hat{q}(1), \hat{q}(2), \ldots, \hat{q}(T-2), S_{N-1}]^{\mathrm{T}}$$

given the observed sequence $\boldsymbol{X}$.

Done

Next week

- **Model parameter estimation:**
  - Adjustment or training of the hidden Markov models based on training data.

Summary

- Principle of Speech recognition
- HMM: General definition
- The three basic problems of HMMs
  - Evaluation problem
  - Decoding problem
  - Model parameter estimation problem
- Evaluation problem:
  - Efficient calculation procedures based on Trellis diagrams and the forward algorithm
- Decoding problem:
  - Efficient calculation based on the Viterbi algorithm

References

Hidden Markov models:

[1] L.R. Rabiner: *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proc. IEEE, vol. 77, no. 2, 1989

[2] B. Pfister, T. Kaufman: *Sprachverarbeitung*, Springer, 2008

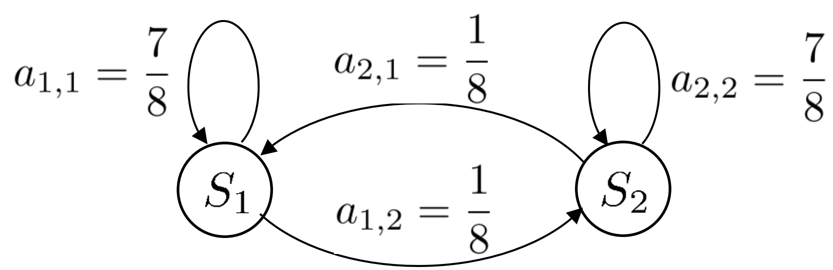
[3] C. M. Bishop: *Pattern Recognition and Machine Learning,* Springer, 2006

[4] L. Rabiner, B.H. Juang: *Fundamentals of Speech Recognition*, Prentice Hall, 1993

[5] B. Gold, N. Morgan: *Speech and Audio Signal Processing*, Wiley, 2000

HMM Task: „two coin experiment“

	Zahl (‚Z‘):	Wappen (‚W‘):
S_1 :	$p(x(n) = Z \vert S_1) = \frac{1}{4}$	$p(x(n) = W \vert S_1) = \frac{3}{4}$
S_2 :	$p(x(n) = Z \vert S_2) = \frac{1}{2}$	$p(x(n) = W \vert S_2) = \frac{1}{2}$



- Observed sequence: $\boldsymbol{X} = [W\,W\,Z\,W]$
- Start in state: S_1 and end in state: S_2
- Task: Calculate the probability that this HMM has generated the observed sequence.
- Compare it to a single „normal coin“ $p(x(n) = Z) = p(x(n) = W) = \frac{1}{2}$