Pattern Recognition

Part 9: Hidden Markov Models (HMMs)

Gerhard Schmidt

Christian-Albrechts-Universität zu Kiel
Faculty of Engineering
Institute of Electrical and Information Engineering
Digital Signal Processing and System Theory

Contents

- Motivation
- Fundamentals
  - The „hidden“ part of the model
  - The inner family of random processes
- Fundamental problems of Hidden Markov Models
  - Efficient calculation of sequence probabilities
  - Efficient calculation of the most probable sequence
  - Calculation (estimation) of the model parameters

Motivation

Modeling of temporal dependencies

- In the previous approaches (vector quantization, Gaussian mixture models), only the probability distribution of multi-dimensional data vectors was analyzed and used. Their ***temporal progression*** was assumed to be ***uncorrelated***.

- If also the temporal progression of the observed data vectors should be analyzed, the previous models can be extended by a temporal component. This new component will again be derived on a ***statistical background***.

- In hidden Markov models, ***two (or three) statistical components are nested***.

- While for multivariate amplitude distributions, both discrete and continuous probability distributions can be used, the ***temporal modeling*** will be done ***discretely***.

Literature

Hidden Markov Models

- B. Pfister, T. Kaufman: ***Sprachverarbeitung***, Springer, 2008 (in German)
- C. M. Bishop: ***Pattern Recognition and Maschine Learning***, Springer, 2006
- L. Rabiner, B.H. Juang: ***Fundamentals of Speech Recognition***, Prentice Hall, 1993
- B. Gold, N. Morgan: ***Speech and Audio Signal Processing***, Wiley, 2000

Common definitions – Part 1

Hidden part of the model (random process) in the Markov model

- The hidden part of the model is assumed to be a Markov process

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

  with *N* ***states***. These states are ***not observable***. For the ***state transitions*** from one discrete state to another, ***probabilities*** are specified.

- The hidden states govern a second family of random processes, which result in the ***observable sequence of vectors***

  $$\boldsymbol{X} = \big[\boldsymbol{x}(0),\ \boldsymbol{x}(1),\ \ldots,\ \boldsymbol{x}(T-1)\big]\,.$$

- The sequence of hidden states is denoted as

  $$\boldsymbol{q} = \big[q(0),\ q(1),\ \ldots,\ q(T-1)\big]^{\mathrm{T}}$$

  where the elements $q(n)$ each correspond to one of the hidden states, respectively:

  $$q(n) \in \big\{S_0,\ S_1,\ \ldots,\ S_{N-1}\big\}.$$

Common definitions – Part 2

Hidden part of the model (random process) in the Markov model

- As soon as the model gets into a new state, the model generates an ***observation vector***. Its distribution is only ***dependant on the new state*** $q(n)$, but not on previous ones:

$$p\Big(\boldsymbol{x}(n)\big|q(n)=S_j,\, q(n-1)=S_i,\, \ldots,\, \boldsymbol{x}(0),\, \boldsymbol{x}(1),\, \ldots,\, \boldsymbol{x}(n-1)\Big)$$
$$= p\Big(\boldsymbol{x}(n)\big|q(n)=S_j\Big).$$

Emission probability

In the following, this probability is denoted as $b_j(\boldsymbol{x})$,

$$p\Big(\boldsymbol{x}(n)\big|q(n)=S_j\Big) = b_j\big(\boldsymbol{x}(n)\big).$$

- The ***state transitions*** are specified (surprise!) by probabilities. These transition probabilities depend only on the current transition's source and target state, but not on previous states.

Transition probability

$$p\Big(q(n)=S_j\big|\, q(n-1)=S_i,\, q(n-2)=S_k,\, \ldots\Big)$$
$$= p\Big(q(n)=S_j\big|\, q(n-1)=S_i\Big).$$

Common definitions – Part 3

Hidden part of the model (random process) in the Markov model

- The ***transition probabilities*** are abbreviated as follows,

$$p\Big(q(n) = S_j \Big|\, q(n-1) = S_i\Big) = a_{i,j}.$$

- The ***initial and final states*** of a HMM are called

S_0 initial state, and

S_{N-1} final state.

Both states are modeled as "***non-emitting***".
The direct transition from the initial to the final state is forbidden – no observation would be created in this case.
I.e., for the transition probabilities, the following holds:

$$a_{i,0} = 0, \quad a_{N-1,i} = 0, \quad a_{0,N-1} = 0 \quad (\text{for } i \in \{0,\ N-1\}).$$

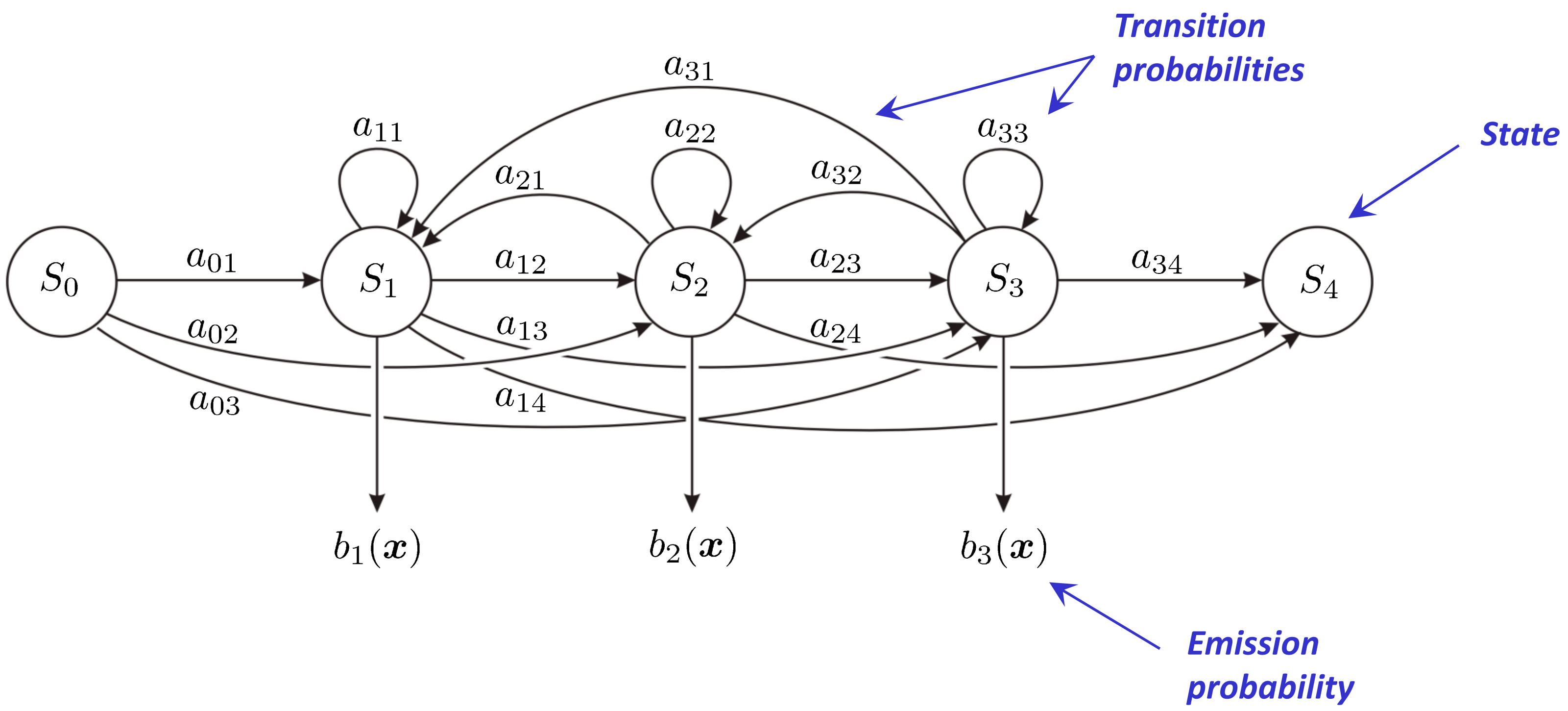
Direct transition from initial to final state

Transitions that leave the final state

Transitions that enter the initial state

Common definitions – Part 4

Hidden part of the model (random process) in the Markov model



Common definitions – Part 5

Hidden part of the model (random process) in the Markov model

- The ***transition probabilities*** of the model are combined in a ***transition matrix***

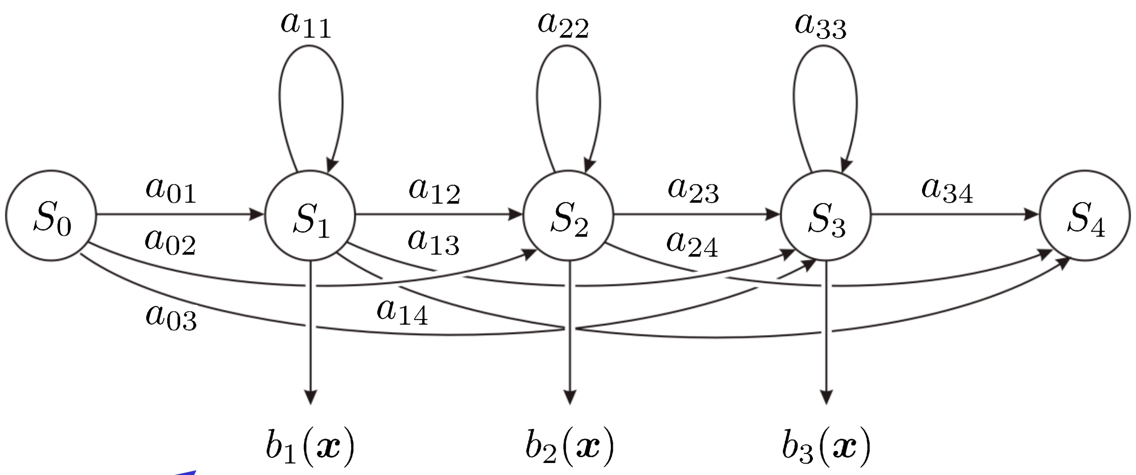
$$\boldsymbol{A} = \begin{bmatrix} 0 & a_{0,1} & a_{0,2} & \cdots & a_{0,N-2} & 0 \\ 0 & a_{1,1} & a_{1,2} & \cdots & a_{1,N-2} & a_{1,N-1} \\ 0 & a_{2,1} & a_{2,2} & \cdots & a_{2,N-2} & a_{2,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{N-2,1} & a_{N-2,2} & \cdots & a_{N-2,N-2} & a_{N-2,N-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

- The constraints are:

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

Types of hidden Markov models – Part 1

Hidden Markov models of the type "left to right"



Structure of a left-to-right Markov model

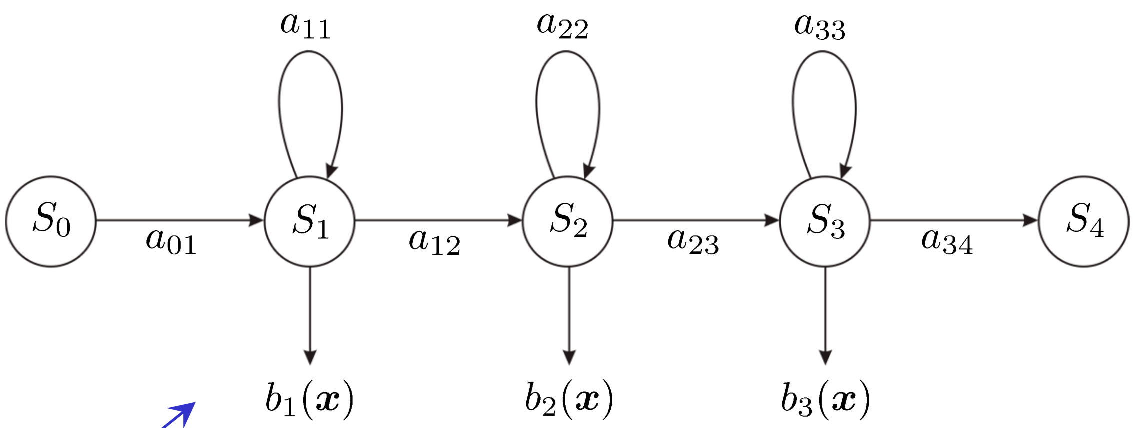
- ***Initial, final and three emitting states are shown.***
- ***Transitions from right to left are not possible.***

Transition matrix

$$\boldsymbol{A} = \begin{bmatrix} 0 & a_{0,1} & a_{0,2} & a_{0,3} & 0 \\ 0 & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ 0 & 0 & a_{2,2} & a_{2,3} & a_{2,4} \\ 0 & 0 & 0 & a_{3,3} & a_{3,4} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Types of hidden Markov models – Part 2

Linear hidden Markov models



Structure of a linear hidden Markov model

- ***Initial, final, and three emitting states are shown.***
- ***Only transitions to the state itself and to right neighbors are possible. Consequently, a sequence of observations must have at least 3 observations.***

Transition matrix

$$\boldsymbol{A} = \begin{bmatrix} 0 & a_{0,1} & 0 & 0 & 0 \\ 0 & a_{1,1} & a_{1,2} & 0 & 0 \\ 0 & 0 & a_{2,2} & a_{2,3} & 0 \\ 0 & 0 & 0 & a_{3,3} & a_{3,4} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Common definitions – Part 6

Generation of observations by a random process

- In order to generate the ***observation vectors***, another random process is assigned to each state. It can be modeled either as ***discrete*** or as ***continuous*** process.
- If the generation of the observations is modeled as N-2 ***discrete processes*** and each process may have K discrete observation states, then the applied probabilities can again be combined in a ***matrix***

$$\boldsymbol{B} = \begin{bmatrix} b_{1,0} & b_{1,1} & b_{1,2} & \dots & b_{1,K-2} & b_{1,K-1} \\ b_{2,0} & b_{2,1} & b_{2,2} & \dots & b_{2,K-2} & b_{2,K-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{N-2,0} & b_{N-2,1} & b_{N-2,2} & \dots & b_{N-2,K-2} & b_{N-2,K-1} \end{bmatrix}.$$

Again, the following constraints hold:

$$b_{i,k} \geq 0, \qquad \text{for } i \in \{1, \dots, N-2\},\ k \in \{0, \dots, K-1\}$$

$$\sum_{k=0}^{K-1} b_{i,k} = 1, \qquad \text{for } i \in \{1, \dots, , N-2\}.$$

Common definitions – Part 7

Generation of observations by a random process

- If the generation of observations is modeled as ***continuous processes*** using ***multivariate Gaussian densities*** (GMMs), then the applied probabilities can be defined as follows,

$$\begin{aligned} b_i(\boldsymbol{x}) &= \sum_{k=0}^{K-1} g_{i,k}\, b_{i,k}(\boldsymbol{x}) \\ &= \sum_{k=0}^{K-1} g_{i,k}\, \mathcal{N}\big(\boldsymbol{x}|\boldsymbol{\mu}_{i,k},\, \boldsymbol{\Sigma}_{i,k}\big)\,, \end{aligned}$$

  assuming that per state K Gaussian distributions are used.

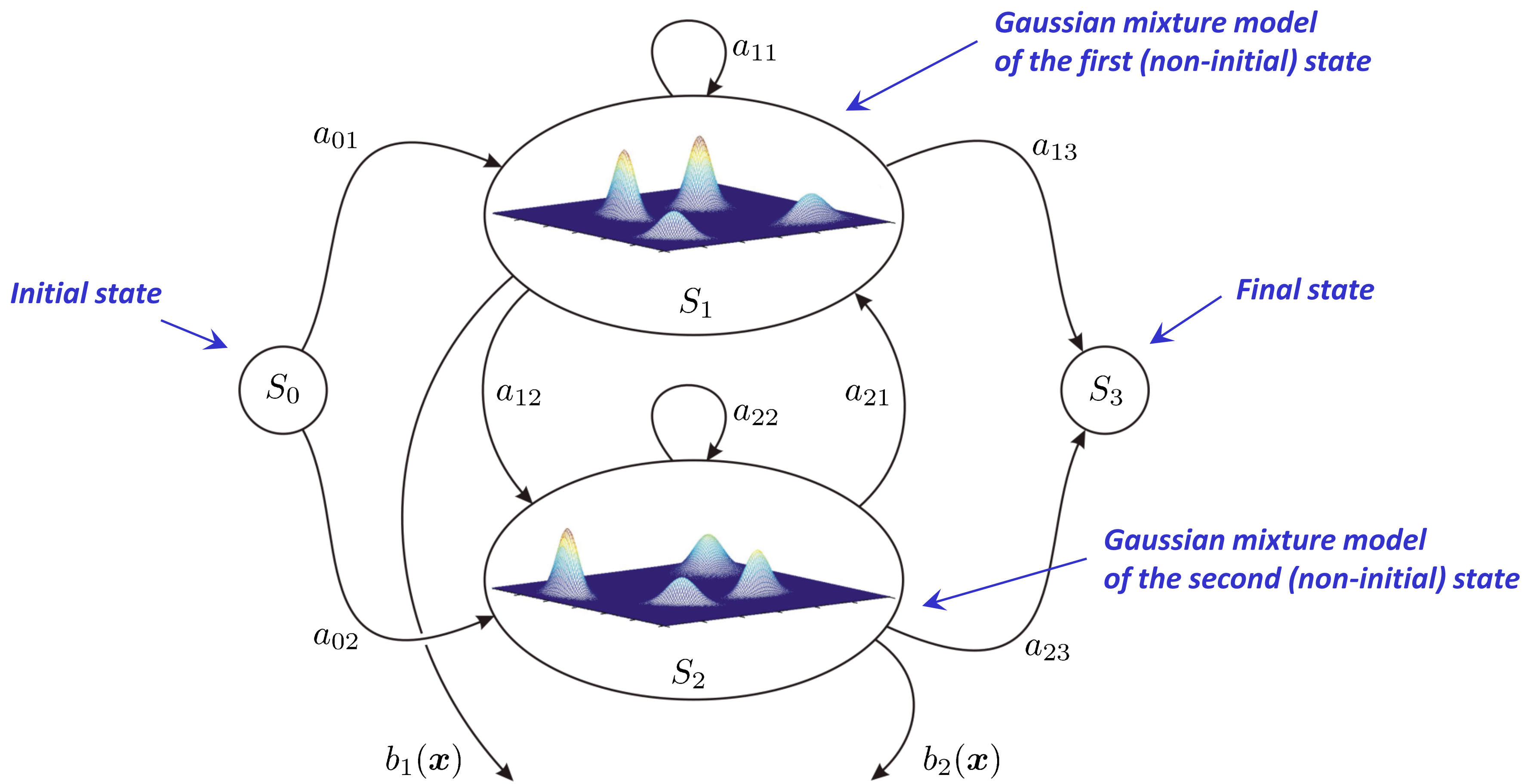
  The Gaussian distributions are defined as in the GMM lecture,

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

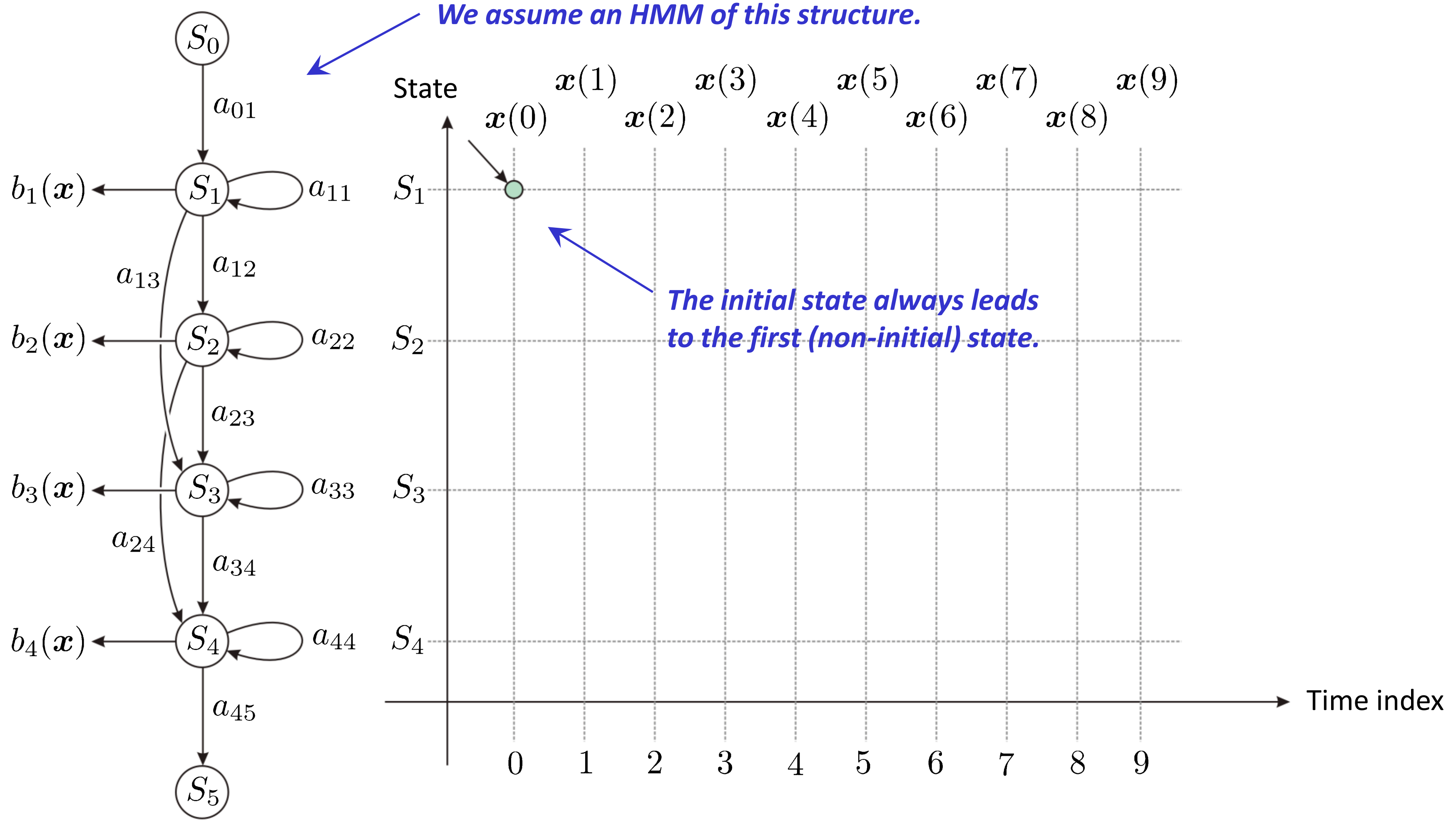
  with $\boldsymbol{x} = \big[x_0,\, x_1,\, \ldots,\, x_{D-1}\big]^{\mathrm{T}}$.

Common definitions – Part 8

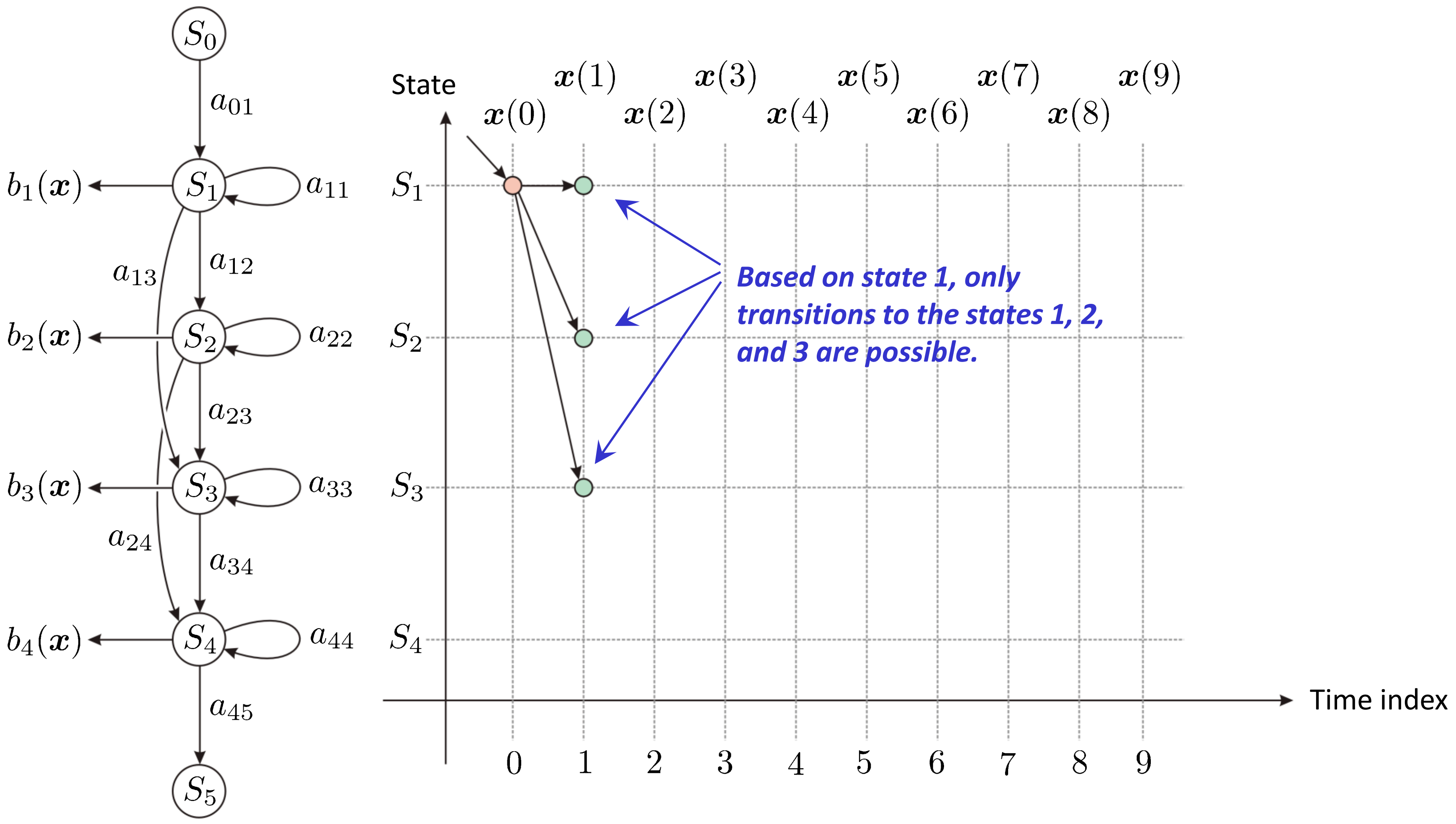
Generation of observations by a random process



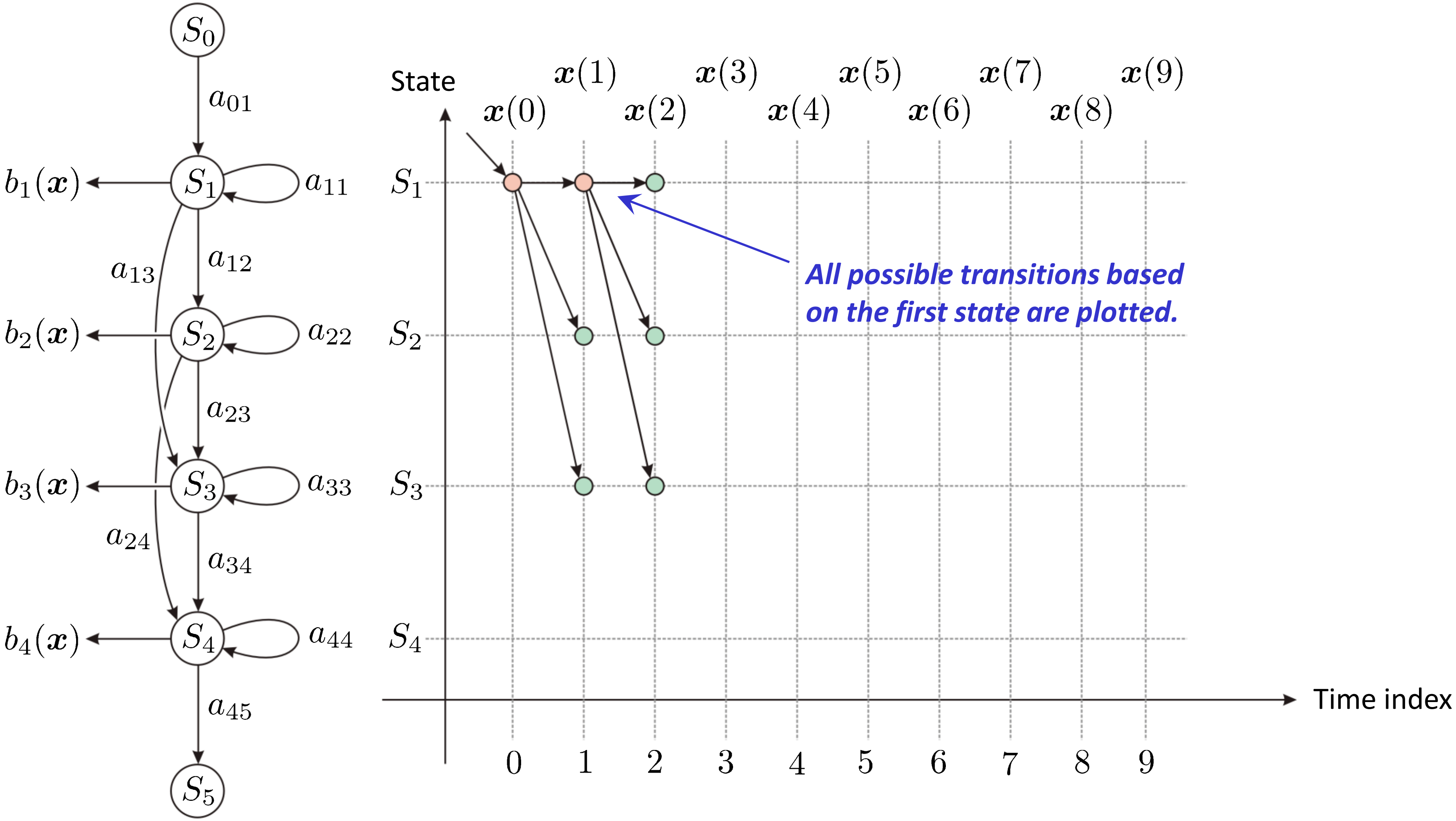
Trellis diagrams – Part 1



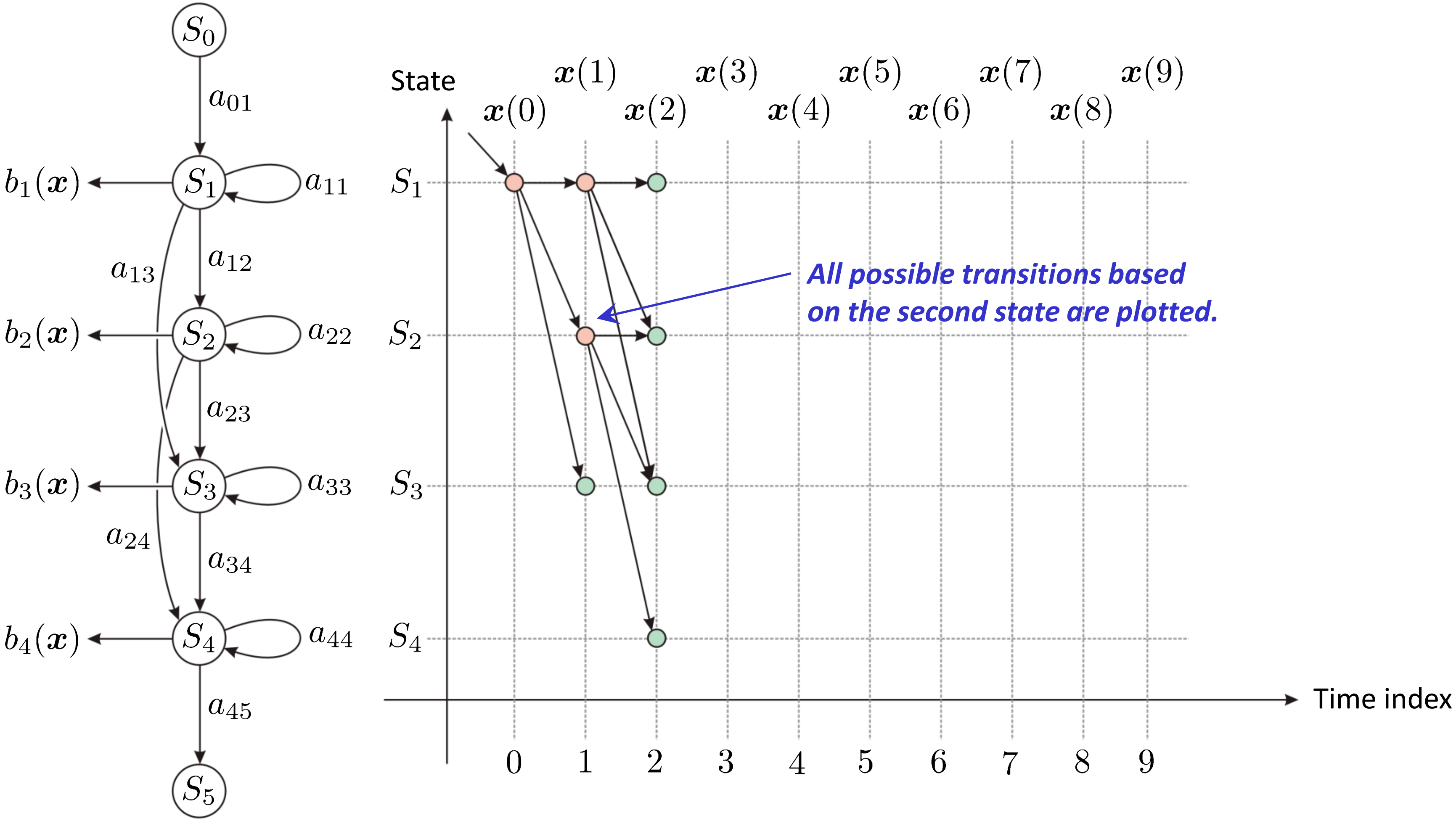
Trellis diagrams – Part 2



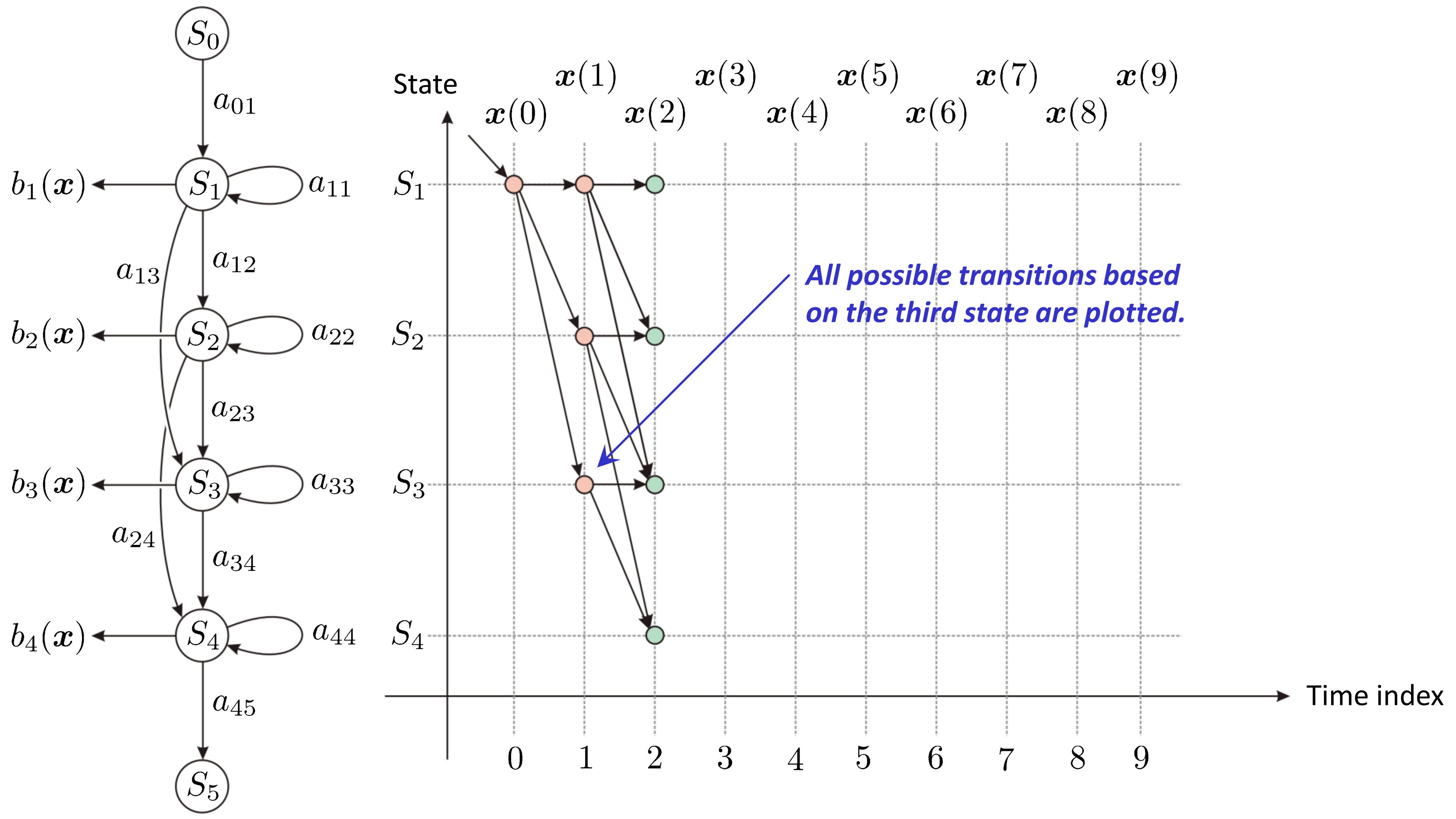
Trellis diagrams – Part 3



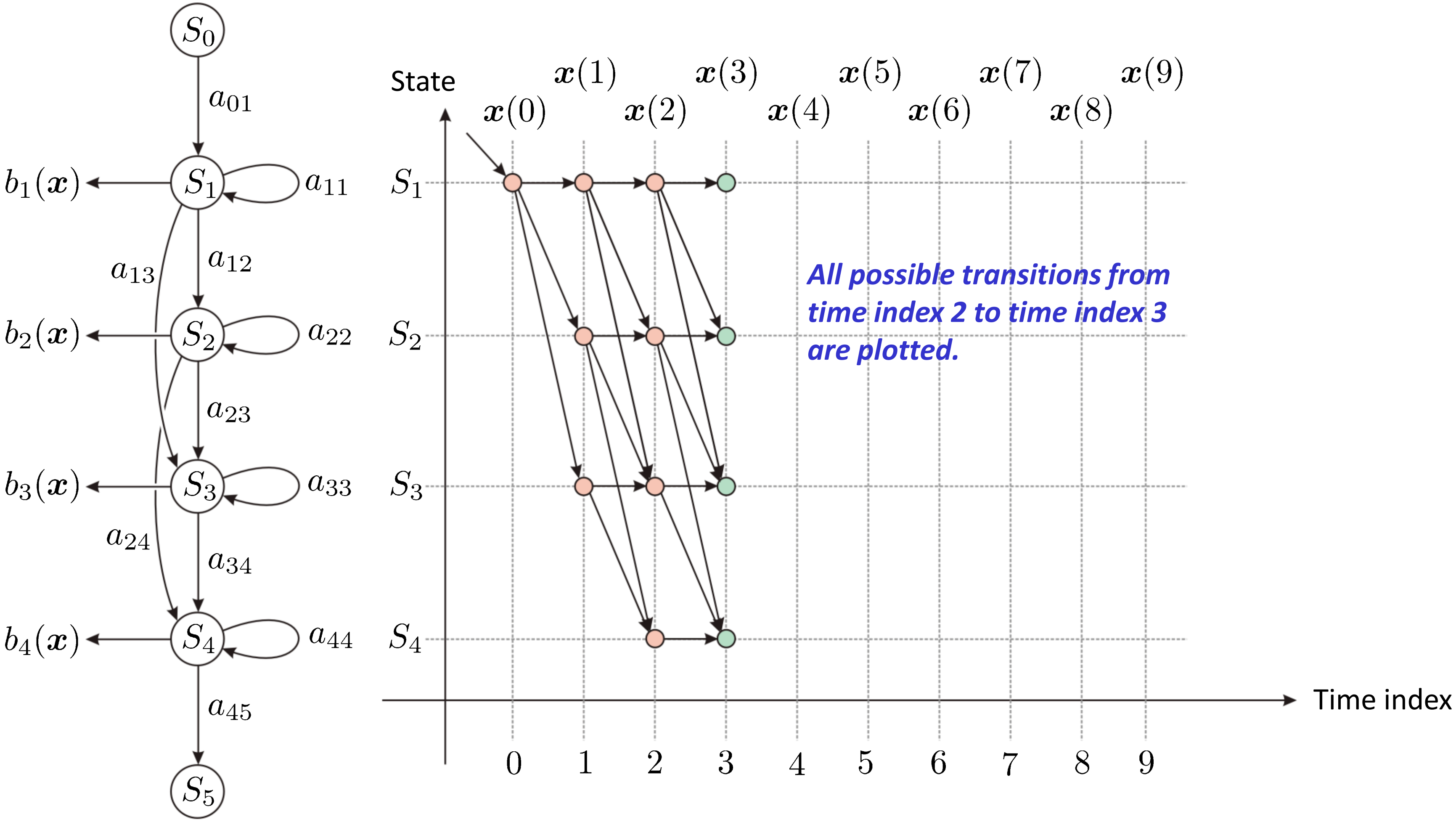
Motivation


S_0
a_{01}
$b_1(\boldsymbol{x})$
S_1
a_{11}
a_{13}
a_{12}
$b_2(\boldsymbol{x})$
S_2
a_{22}
a_{23}
$b_3(\boldsymbol{x})$
S_3
a_{33}
a_{24}
a_{34}
$b_4(\boldsymbol{x})$
S_4
a_{44}
a_{45}
S_5
State
$\boldsymbol{x}(0)$ $\boldsymbol{x}(1)$ $\boldsymbol{x}(2)$ $\boldsymbol{x}(3)$ $\boldsymbol{x}(4)$ $\boldsymbol{x}(5)$ $\boldsymbol{x}(6)$ $\boldsymbol{x}(7)$ $\boldsymbol{x}(8)$ $\boldsymbol{x}(9)$
S_1
S_2
S_3
S_4
All possible transitions based on the second state are plotted.
Time index
0 1 2 3 4 5 6 7 8 9

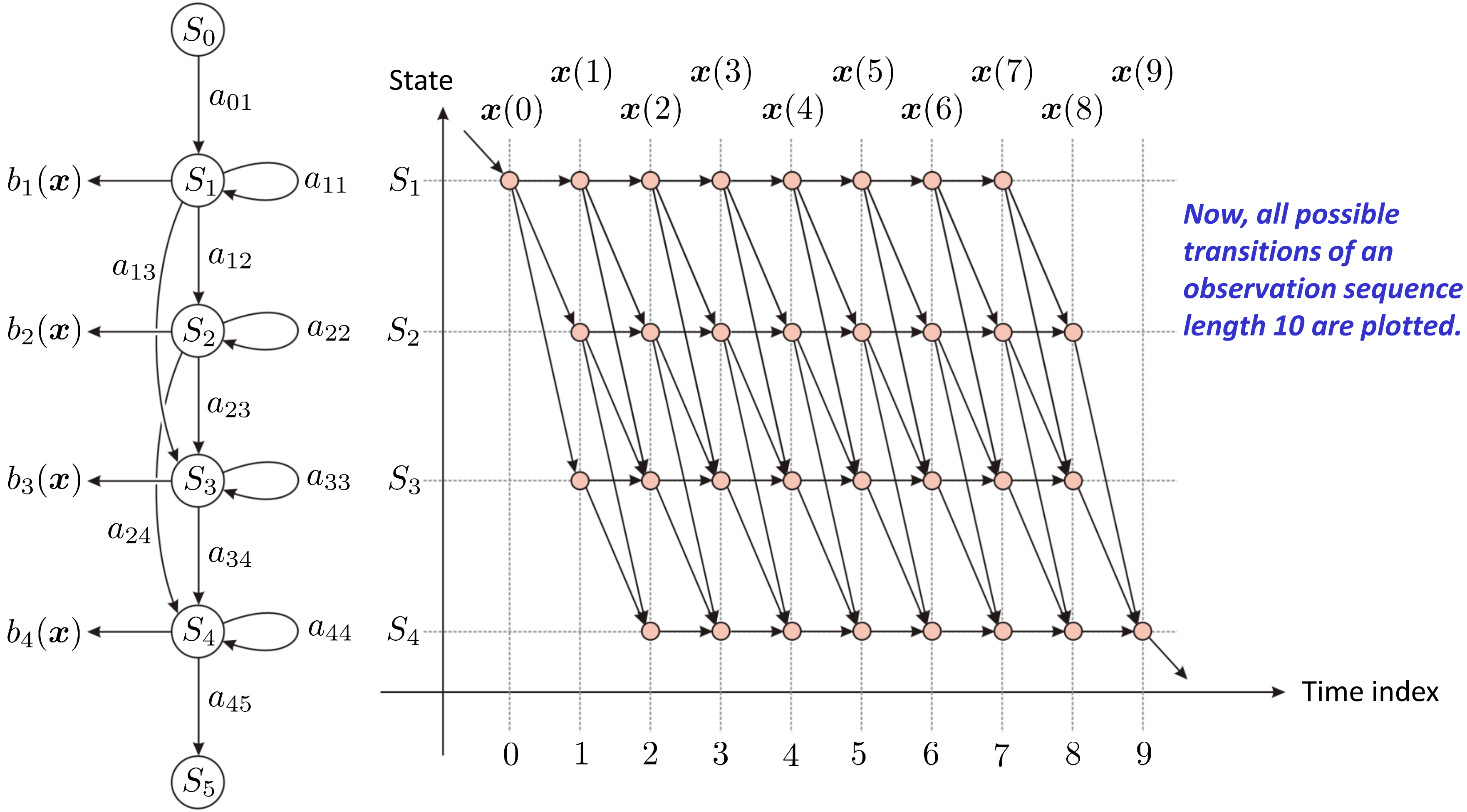
Trellis diagrams – Part 5



Trellis diagrams – Part 6



Trellis diagrams – Part 7

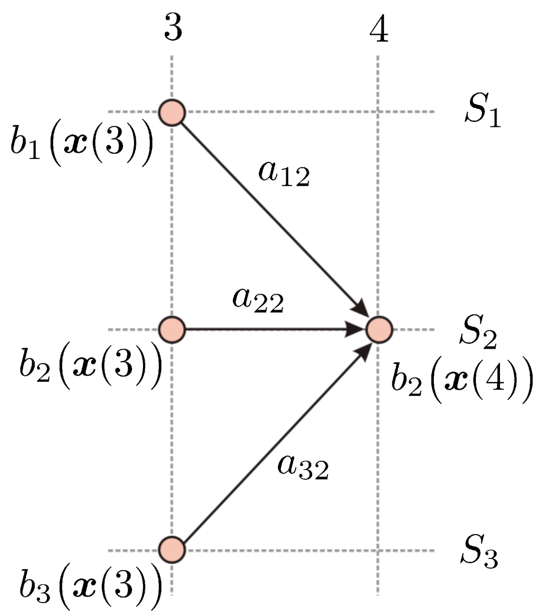


Now, all possible transitions of an observation sequence of length 10 are plotted.

Trellis diagrams – Part 8

Meaning of edges and nodes

- The ***transition probabilities*** are usually denoted at the ***edges***.
- The ***emission probability***, that the observed vector $\boldsymbol{x}(n)$ is produced by the corresponding state, is denoted at the ***nodes***.



Essential problems of hidden Markov models

Evaluation problem

- The probability $p(\boldsymbol{X}|\lambda)$ that the hidden Markov model λ creates the (given) observation sequence $\boldsymbol{X}$ is to be calculated.
- In order to calculate this probability, all possible observation sequences $\boldsymbol{Q}$ have to be taken into account. The direct calculation (summing over all possible observation sequences) would thus be very time consuming.

Decoding problem

- Besides the probability calculated above, also the state sequence

$$\hat{\boldsymbol{q}} = [S_0,\, \hat{q}_0,\, \hat{q}_1,\, ...,\, \hat{q}_{T-1},\, S_{N-1}]^{\mathrm{T}}$$

  that creates the observation sequence $\boldsymbol{X}$ with the highest probability, is of interest.

Estimation problem

- Based on a huge data base, all parameters of the hidden Markov model are to be estimated.

Evaluation problem – Part 1

Evaluation problem

- The probability $p(\boldsymbol{X}|\lambda)$ that the hidden Markov model λ creates the (given) observation sequence $\boldsymbol{X}$ is to be found.
- The wanted probability can be calculated by summing up the conditional production probabilities of all possible observation sequences,

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

- This can be written as follows,

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

- In the following we will try to calculate the two conditional probabilities separately.

Evaluation problem – Part 2

Evaluation problem

- In a first step, the production probability is being calculated, that results from the assumption that the state sequence $\boldsymbol{q}_i$ is known. We use that the probability of an observation $\boldsymbol{x}(n)$ only depends on the actual state of the HMM – but not of previous or subsequent states:

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

- The probability that the sequence $\boldsymbol{q}_i$ has been selected, can be evaluated as follows:

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

Evaluation problem – Part 3

Evaluation problem

- The production probability results in

$$p(\boldsymbol{X}|\lambda) = \sum_{\boldsymbol{q}_i \in \boldsymbol{Q}} a_{0,q_i(0)}\; b_{q_i(0)}(\boldsymbol{x}(0))\; a_{q_i(0),q_i(1)}\; \dots\; b_{q_i(T-1)}(\boldsymbol{x}(T-1))\; a_{q_i(T-1),N-1}.$$

- The problem when directly calculating the production probability is the fact that per time index, there are N-2 possible states. As a result, for the overall sequence, $(N\text{-}2)^T$ possible paths exist, so the number of summands is ***no longer manageable***.
- As a remedy, the so-called ***forward algorithm*** is used. For this purpose the so-called ***forward probability*** is defined in a first step,

$$f_i(n) = p(\boldsymbol{X}^{(n)},\, q(n) = S_i|\lambda).$$

This is the probability that at time index n, the state S_i is active and the "shortened" observation sequence $X^{(n)}$ could be observed up to now.

Evaluation problem – Part 4

Evaluation problem

- The upper indices specify the ***shortened versions of the observation matrix*** and of the state sequence, respectively:

$$\begin{aligned}\boldsymbol{X}^{(n)} &= \left[\boldsymbol{x}(0),\, \boldsymbol{x}(1),\, ...,\, \boldsymbol{x}(n)\right], \\ \boldsymbol{q}_i^{(n)} &= \left[q_i(0),\, q_i(1),\, ...,\, q_i(n)\right]^{\mathrm{T}}.\end{aligned}$$

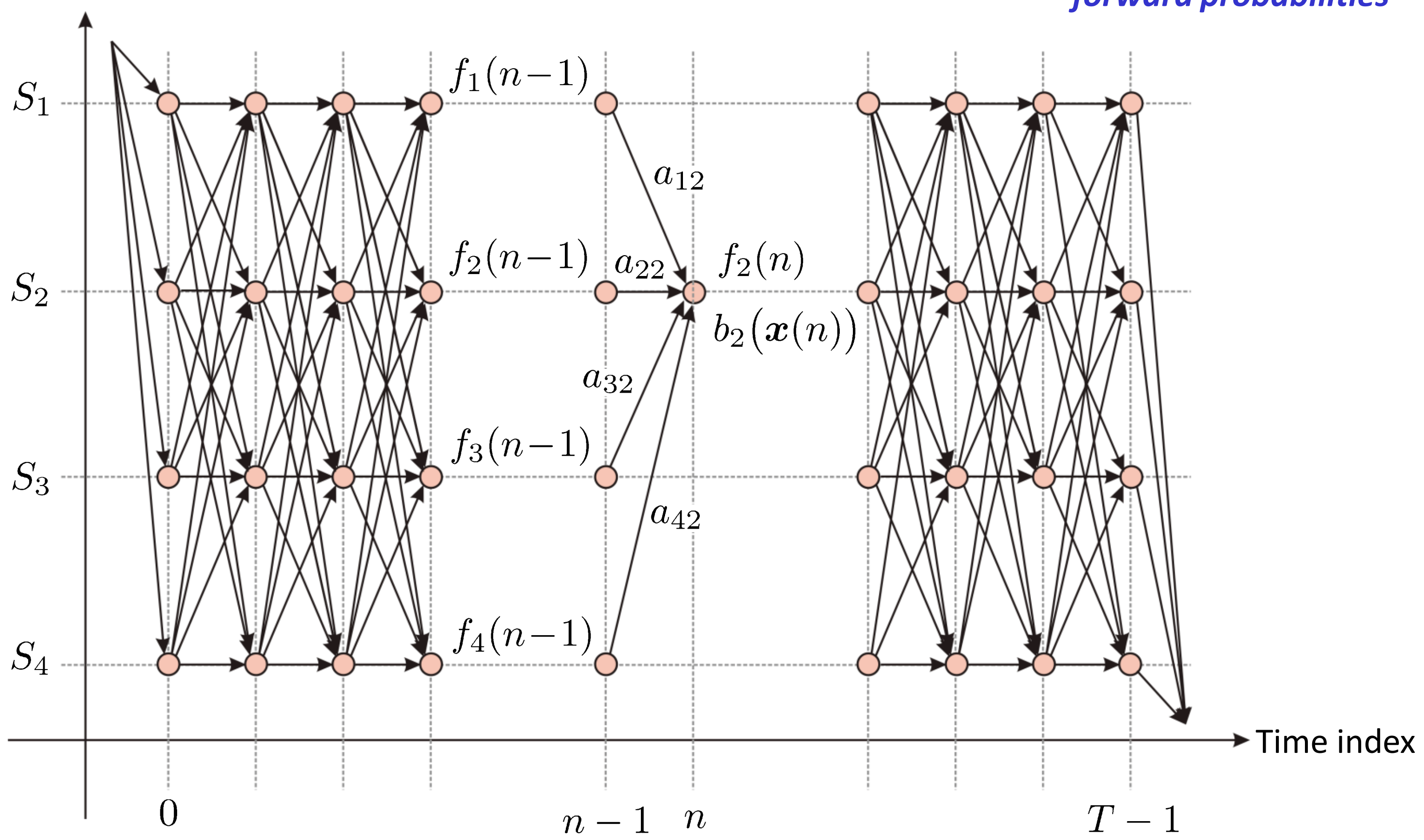
- The forward probability can be determined by summing up all possible shortened observation sequences and being at state S_i at time index n,

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

Evaluation problem – Part 5

Evaluation problem

Illustration of the forward probabilities



Evaluation problem – Part 6

Evaluation problem

- Because of the independence of the previous states, the forward probabilities can be calculated recursively as follows,

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

- The initialization is done as follows,

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

- Hereby, the production probability of the observed sequence can be determined by ***summation of the previous forward probabilities***,

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

- Note that the *computational complexity now just grows linearly with the sequence length* (instead of growing exponentially using direct calculation).

Decoding problem – Part 1

Decoding problem

- Besides the probability that the hidden Markov model λ created the observation vector sequence $\boldsymbol{X}$, some applications require ***the most probable state sequence***. The latter can be defined as follows,

$$\hat{\boldsymbol{q}} = \underset{\boldsymbol{q}_j}{\operatorname{argmax}} \Big\{ p(\boldsymbol{q}_j | \boldsymbol{X}, \lambda) \Big\}.$$

- The conditional probability mentioned above can be permuted,

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

- Because $p(\boldsymbol{X}|\lambda)$ only depends on the (given) observation sequence, also

$$\hat{\boldsymbol{q}} = \underset{\boldsymbol{q}_j}{\operatorname{argmax}} \Big\{ p(\boldsymbol{q}_j, \boldsymbol{X} | \lambda) \Big\}.$$

  can be optimized instead. By this permutation of the cost function, similar quantities as in the previous problem can be considered.

Decoding problem – Part 2

Decoding problem

- The most probable state sequence can be calculated efficiently using the so-called ***Viterbi algorithm***. In analogy to the explanation of the evaluation problem, the joint probability for the shortened observation vector sequence and the optimal shortened state sequence is defined,

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

- The calculation of the ***probability*** can again be computed in a ***recursive*** way,

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

- For each time index and each state, ***the index of the state that induced the maximum probability has to be stored***, so the optimal path can be tracked later on.

$$t_i(n) = \operatorname*{argmax}_{j=1\ldots N-2} \Big\{ v_j(n-1)\, a_{j,i} \Big\}.$$

Summary of the Viterbi algorithm

- Initialization

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

- Recursion (Iteration)

$$\begin{aligned} v_i(n) &= \max_{j=1\ldots N-2} \big\{v_j(n-1)\, a_{j,i}\big\}\, b_i\big(\boldsymbol{x}(n)\big), \\ t_i(n) &= \operatorname*{argmax}_{j=1\ldots N-2} \big\{v_j(n-1)\, a_{j,i}\big\}. \end{aligned}$$

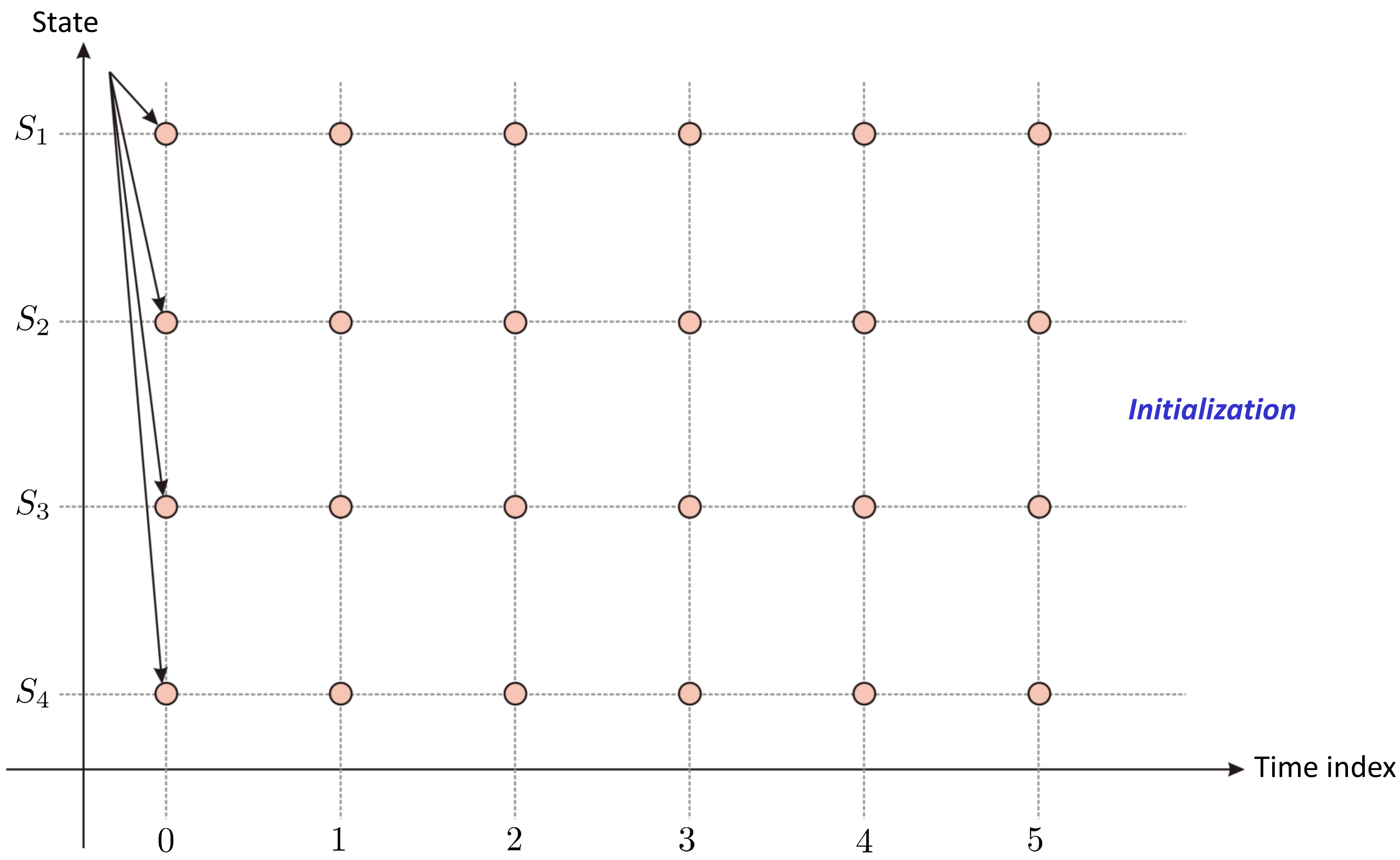
- Termination

$$\begin{aligned} v_{N-1}(T) &= \max_{j=1\ldots N-2} \big\{v_j(T-1)\, a_{j,N-1}\big\}, \\ t_{N-1}(T) &= \operatorname*{argmax}_{j=1\ldots N-2} \big\{v_j(T-1)\, a_{j,N-1}\big\}. \end{aligned}$$

- Backtracking of the optimal state sequence

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

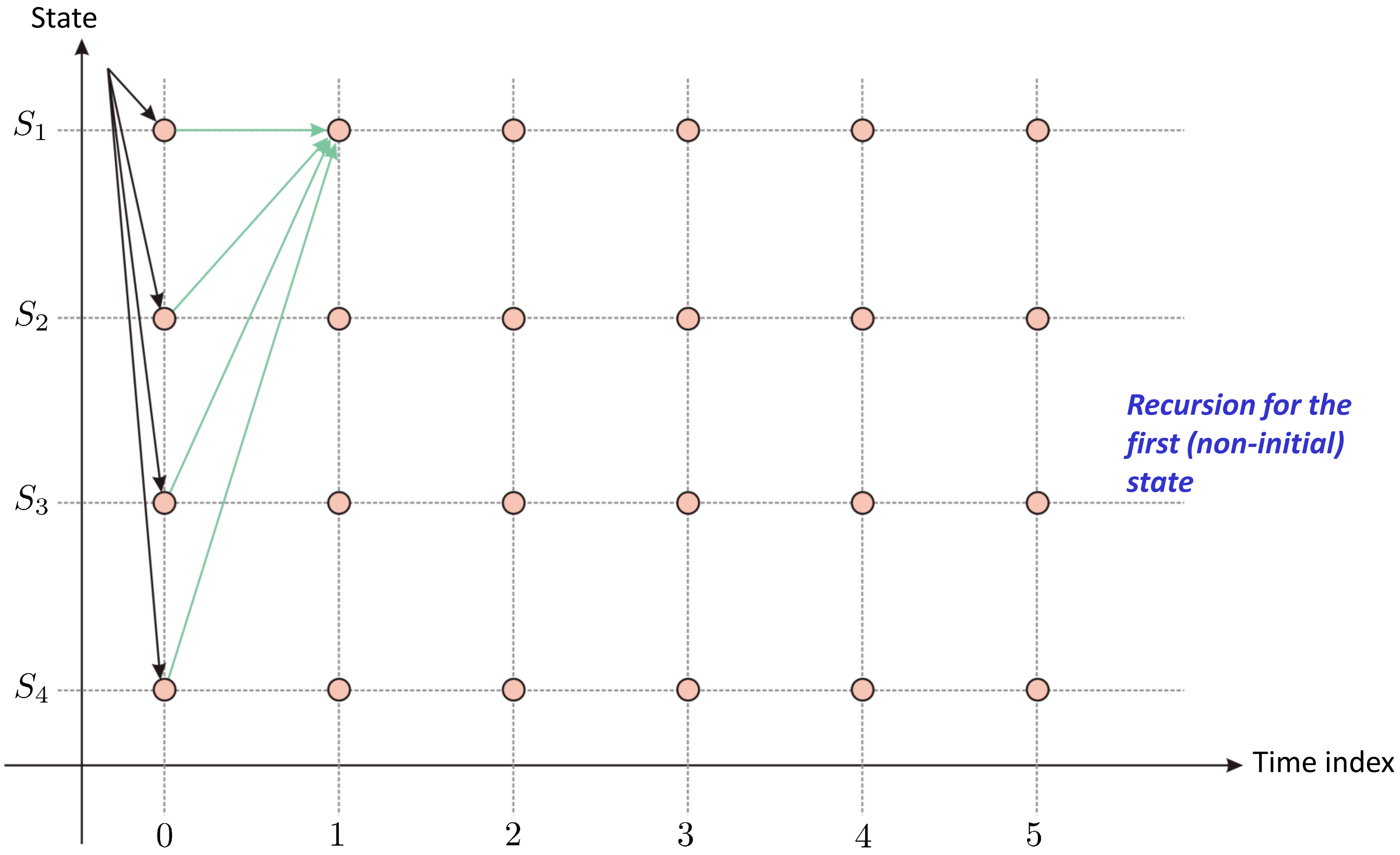
Decoding problem – Part 4



Initialization

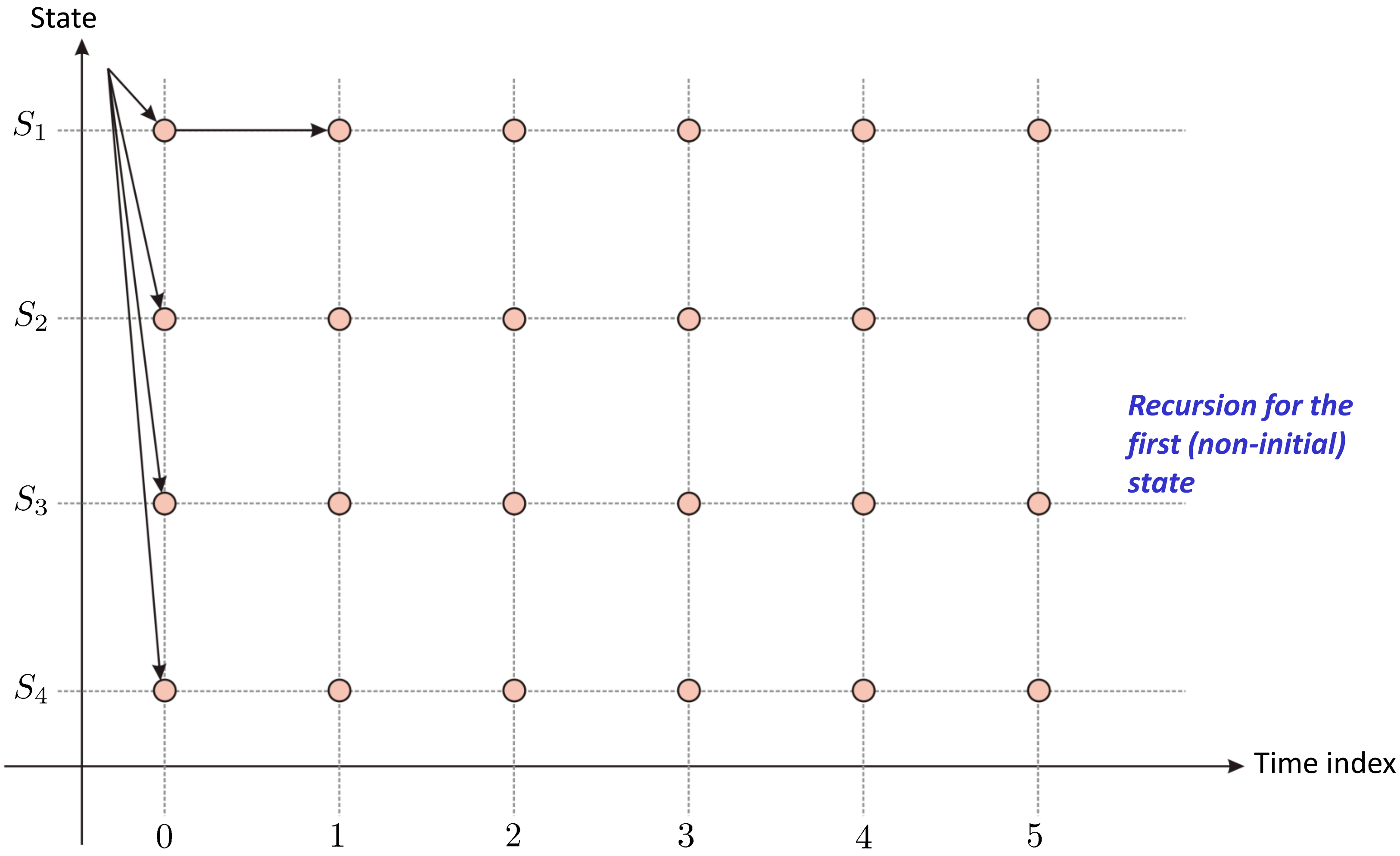
C | A | U
Christian-Albrechts-Universität zu Kiel

Decoding problem – Part 5



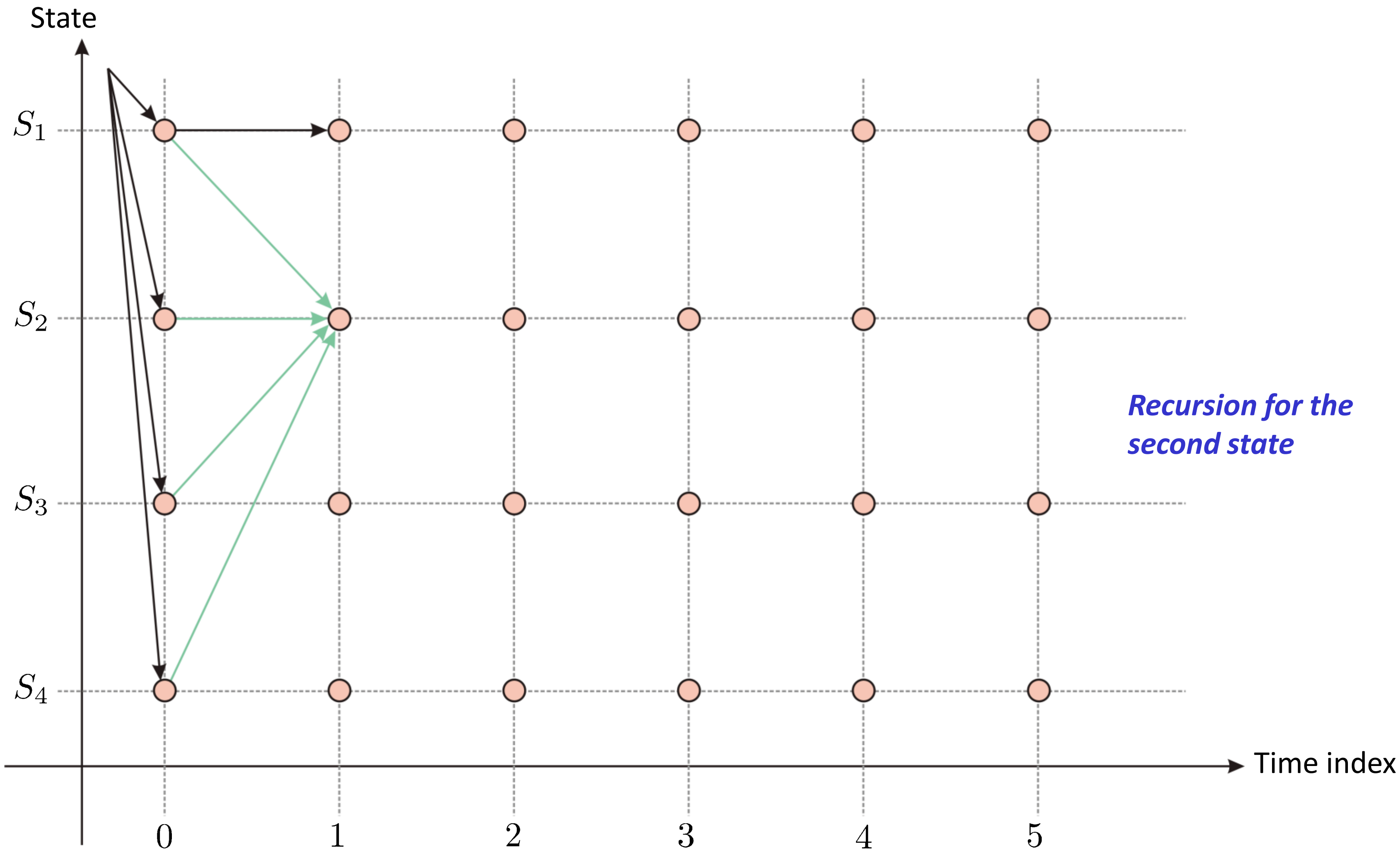
Recursion for the first (non-initial) state

Decoding problem – Part 6



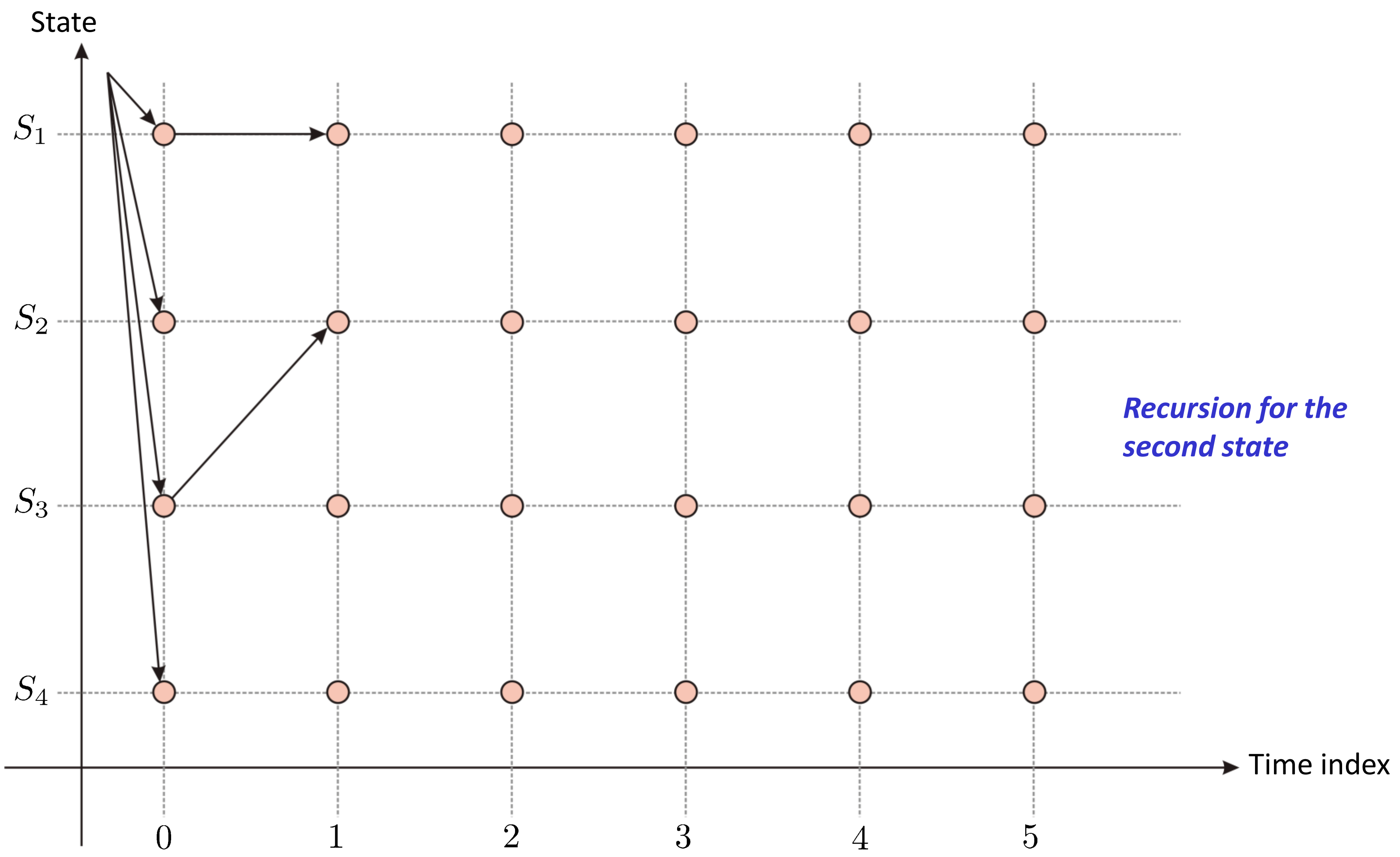
Recursion for the first (non-initial) state

Decoding problem – Part 7



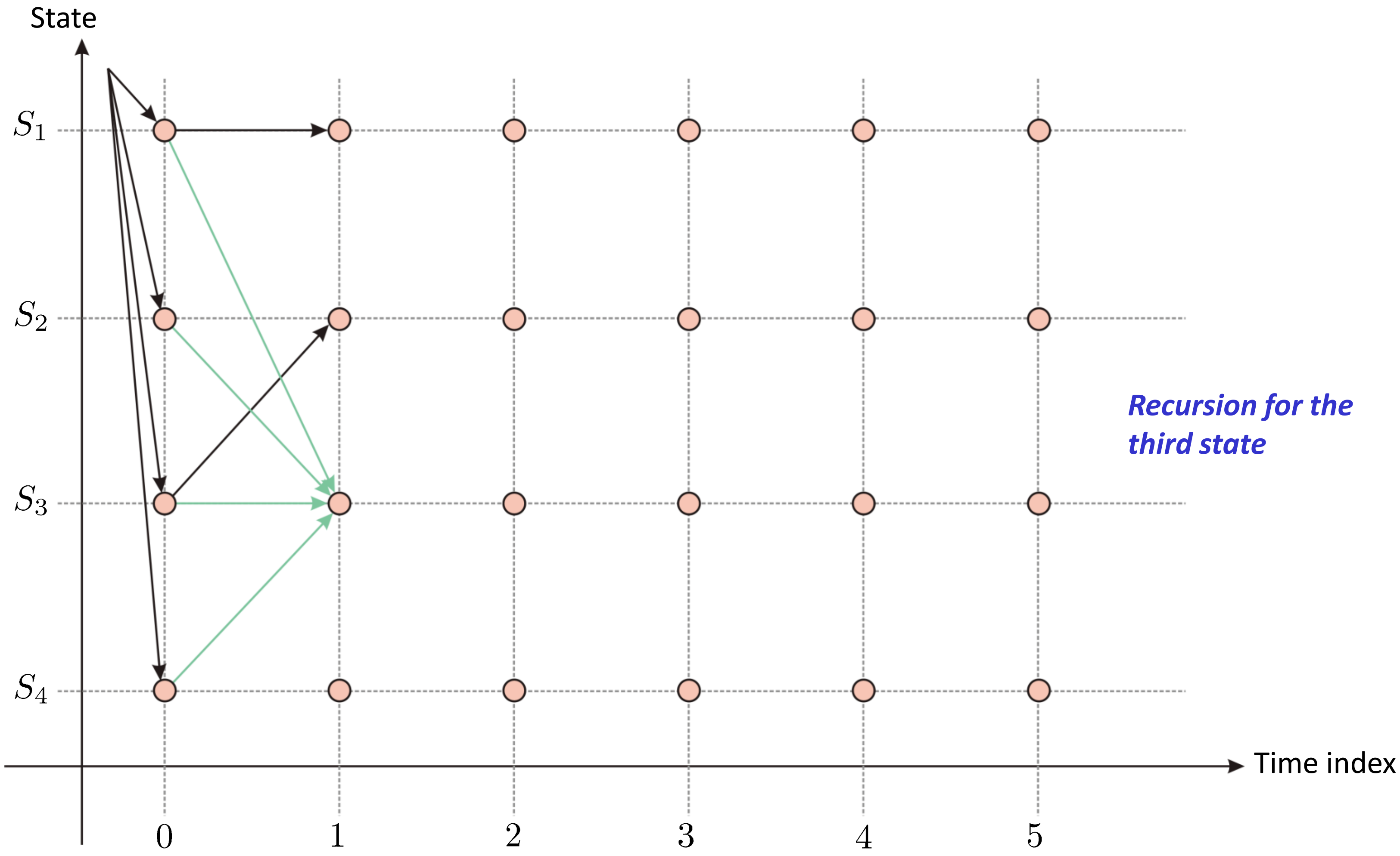
Recursion for the second state

Decoding problem – Part 8



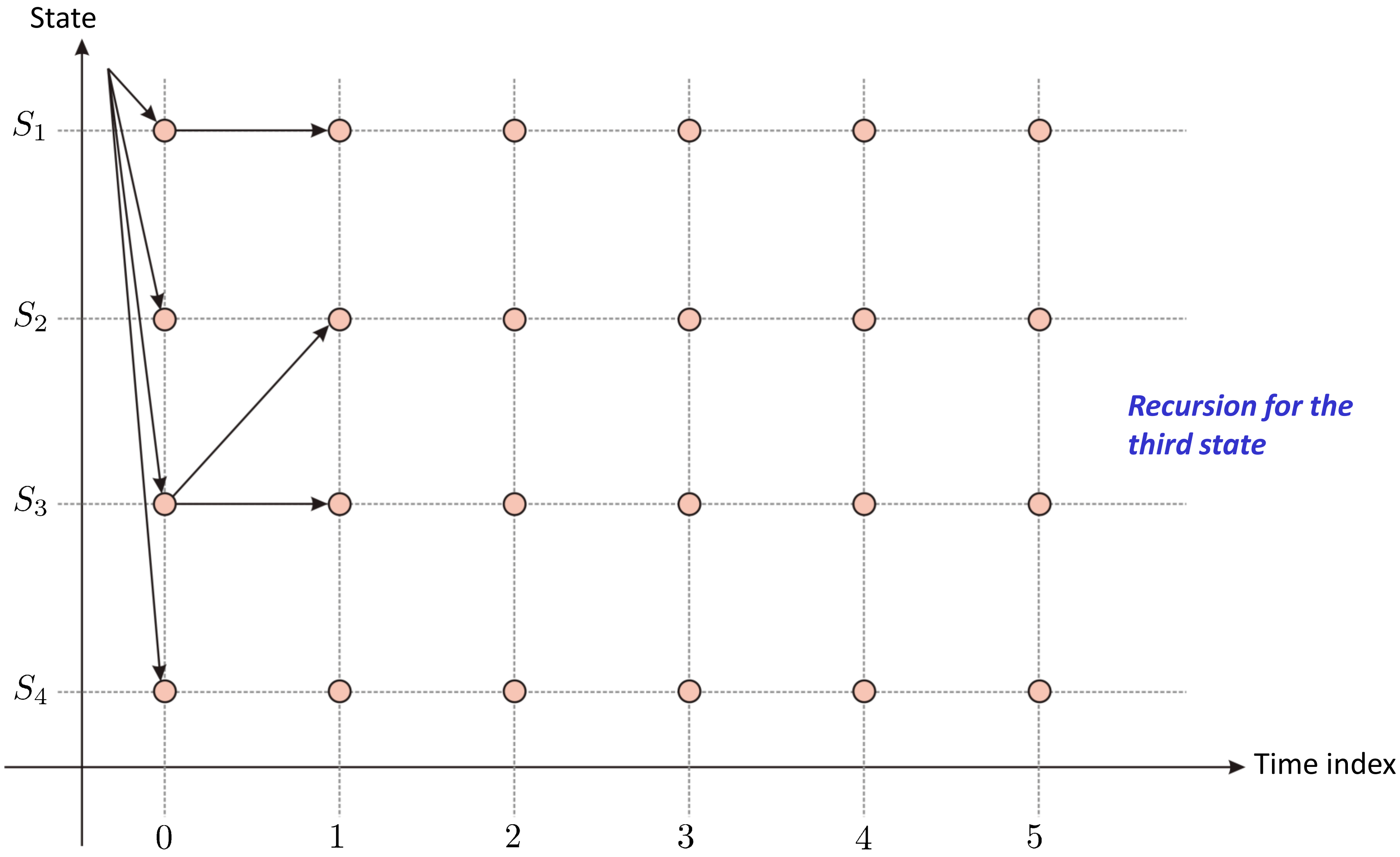
Recursion for the second state

Decoding problem – Part 9



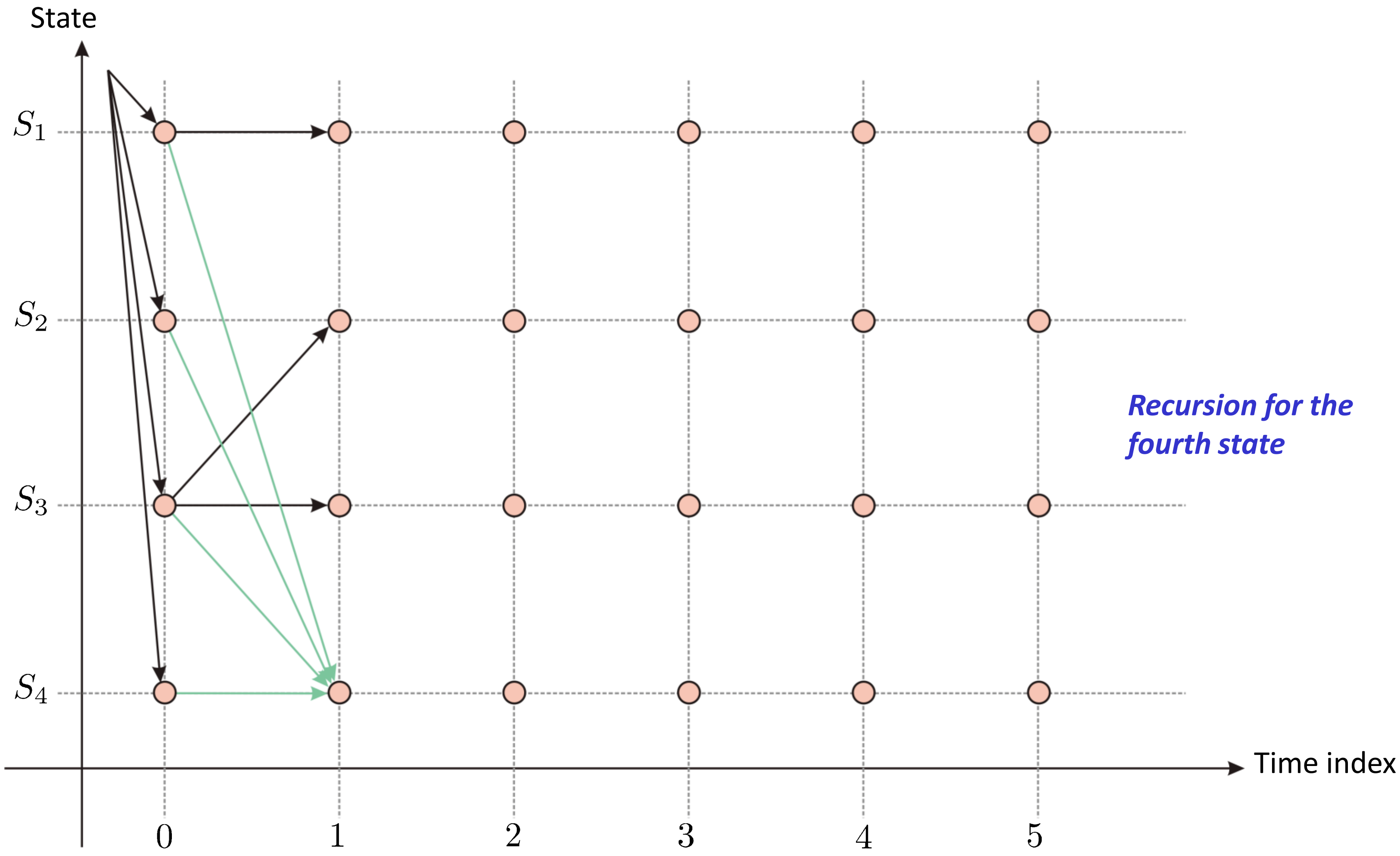
Recursion for the third state

Decoding problem – Part 10



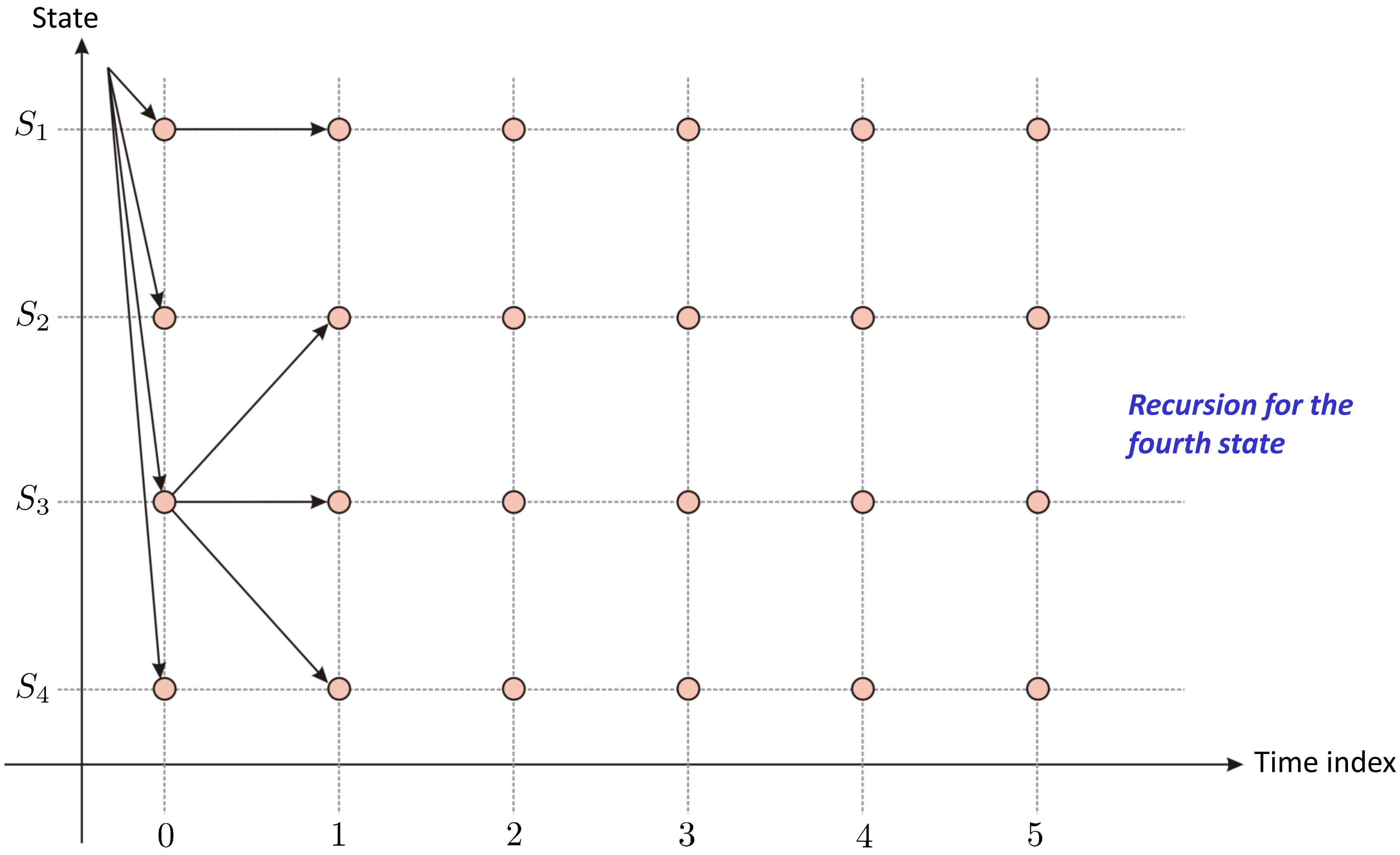
Recursion for the third state

Decoding problem – Part 11



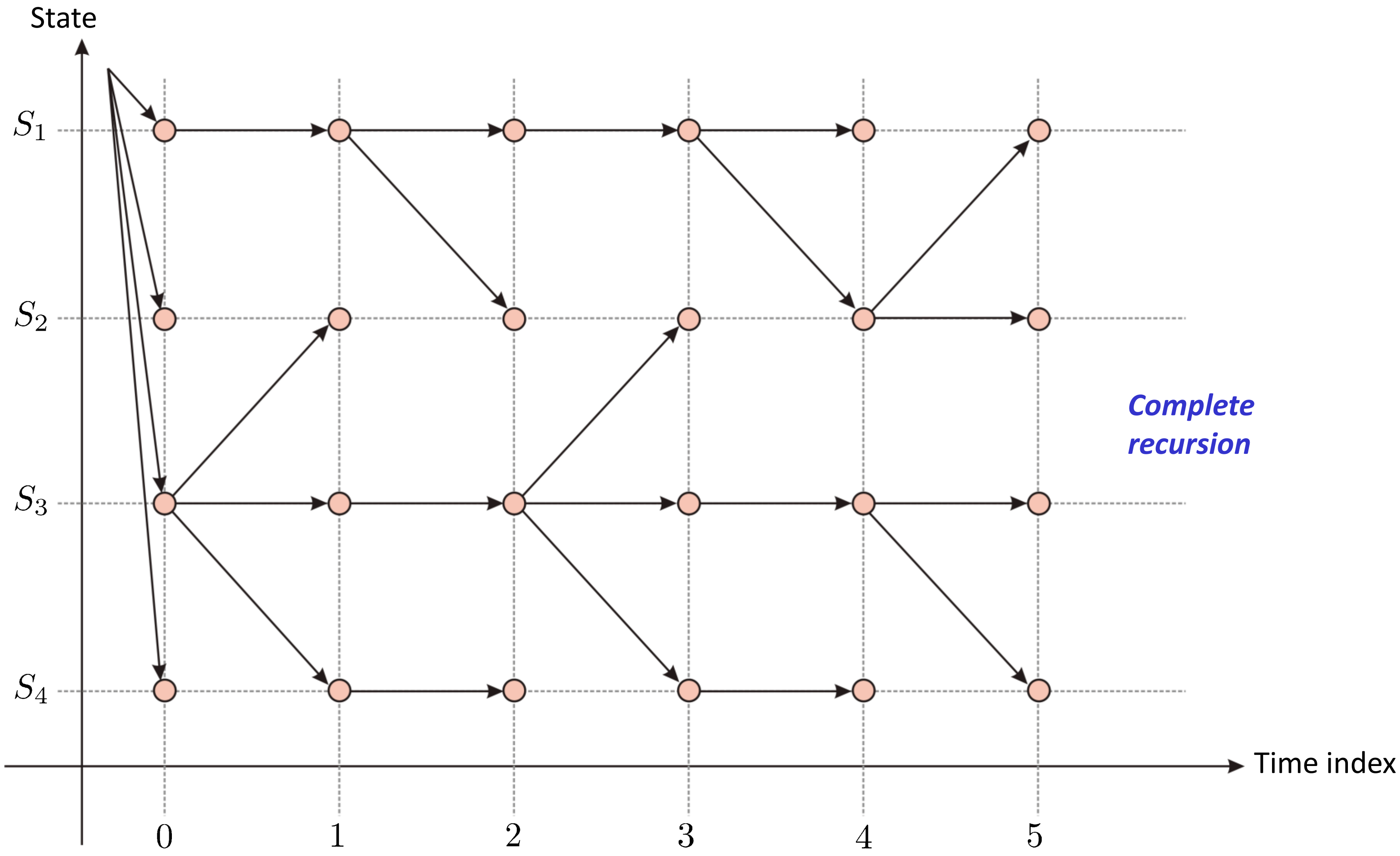
Recursion for the fourth state

Decoding problem – Part 12



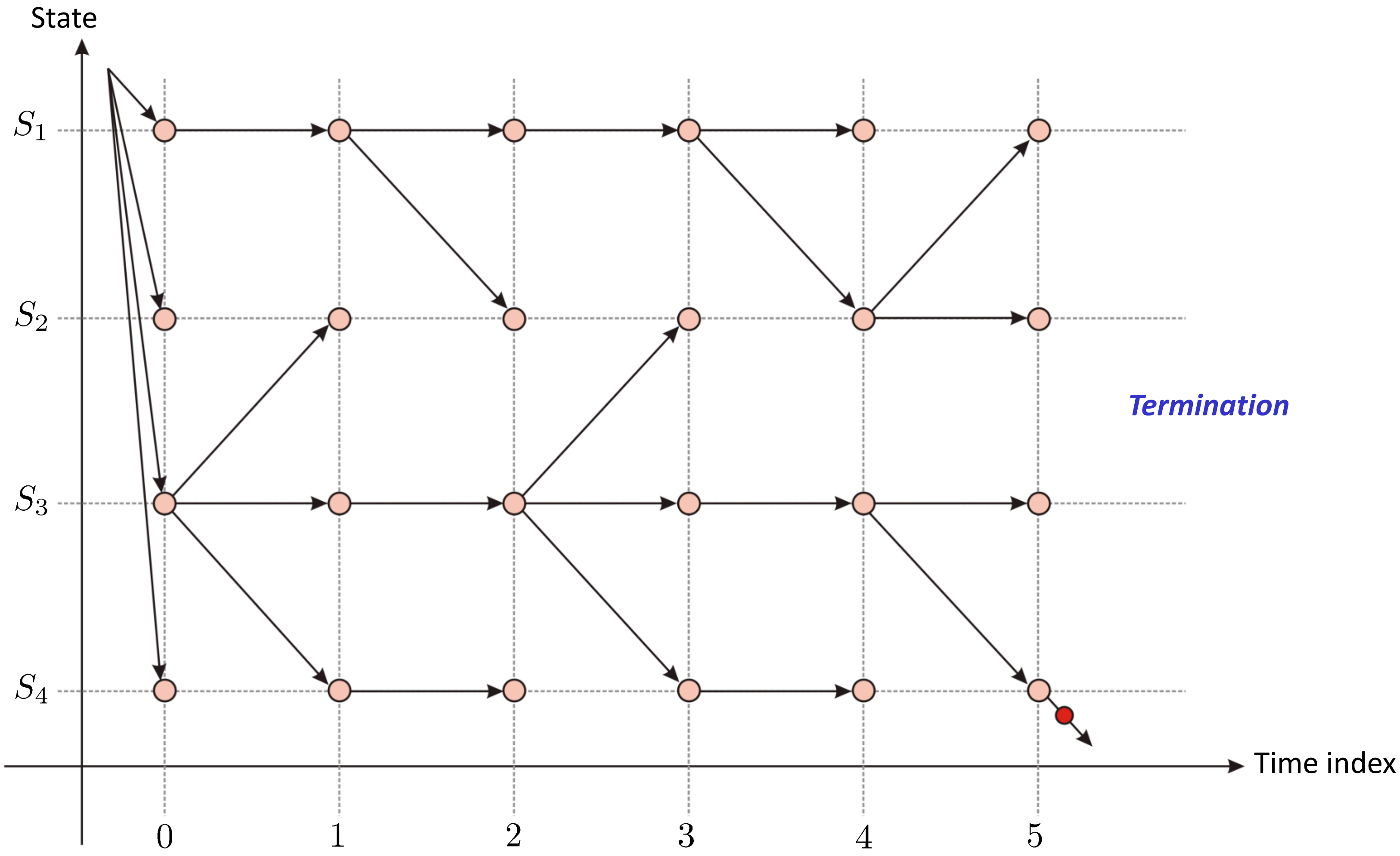
Recursion for the fourth state

Decoding problem – Part 13



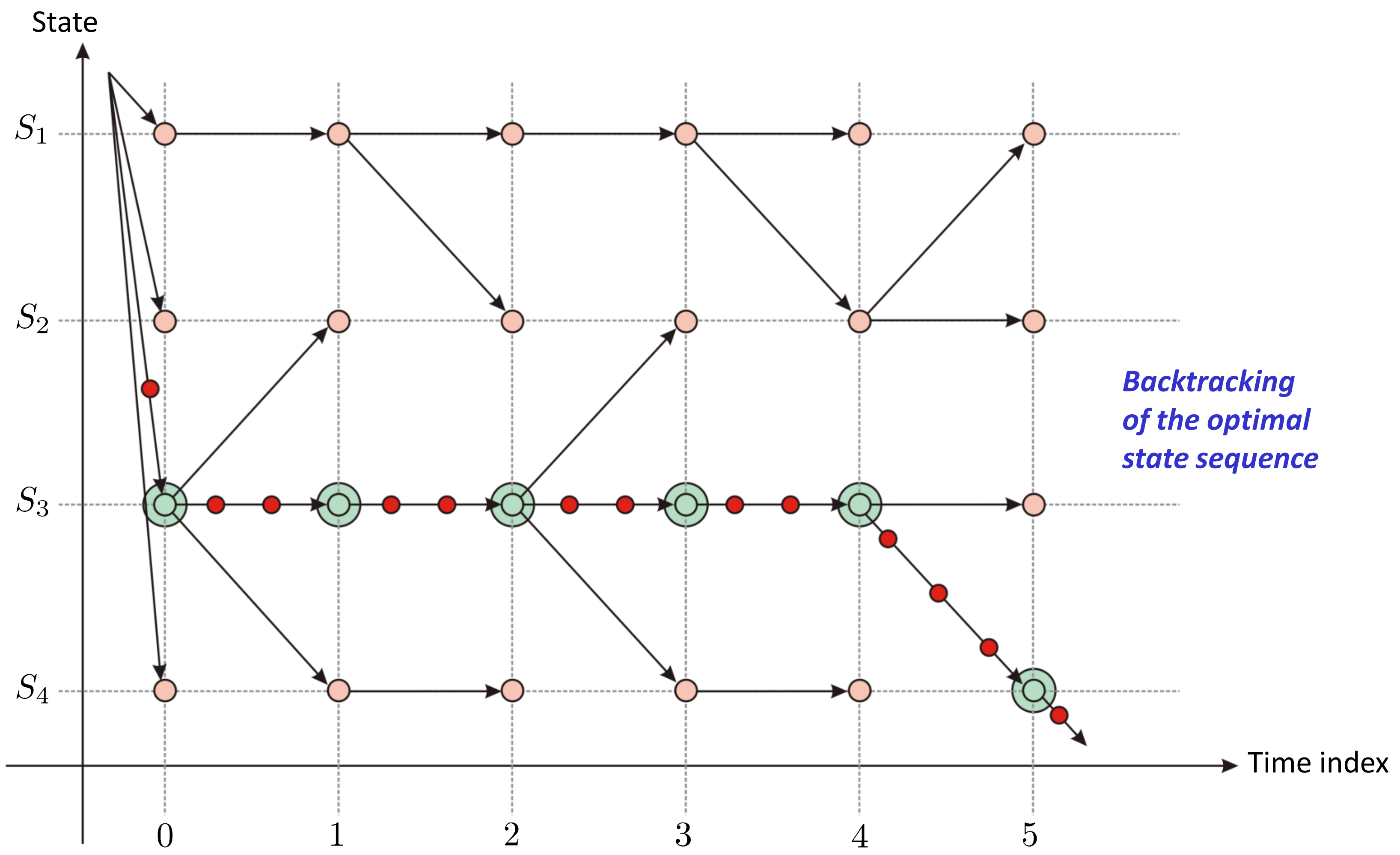
Complete recursion

Decoding problem – Part 14



Termination

Decoding problem – Part 15

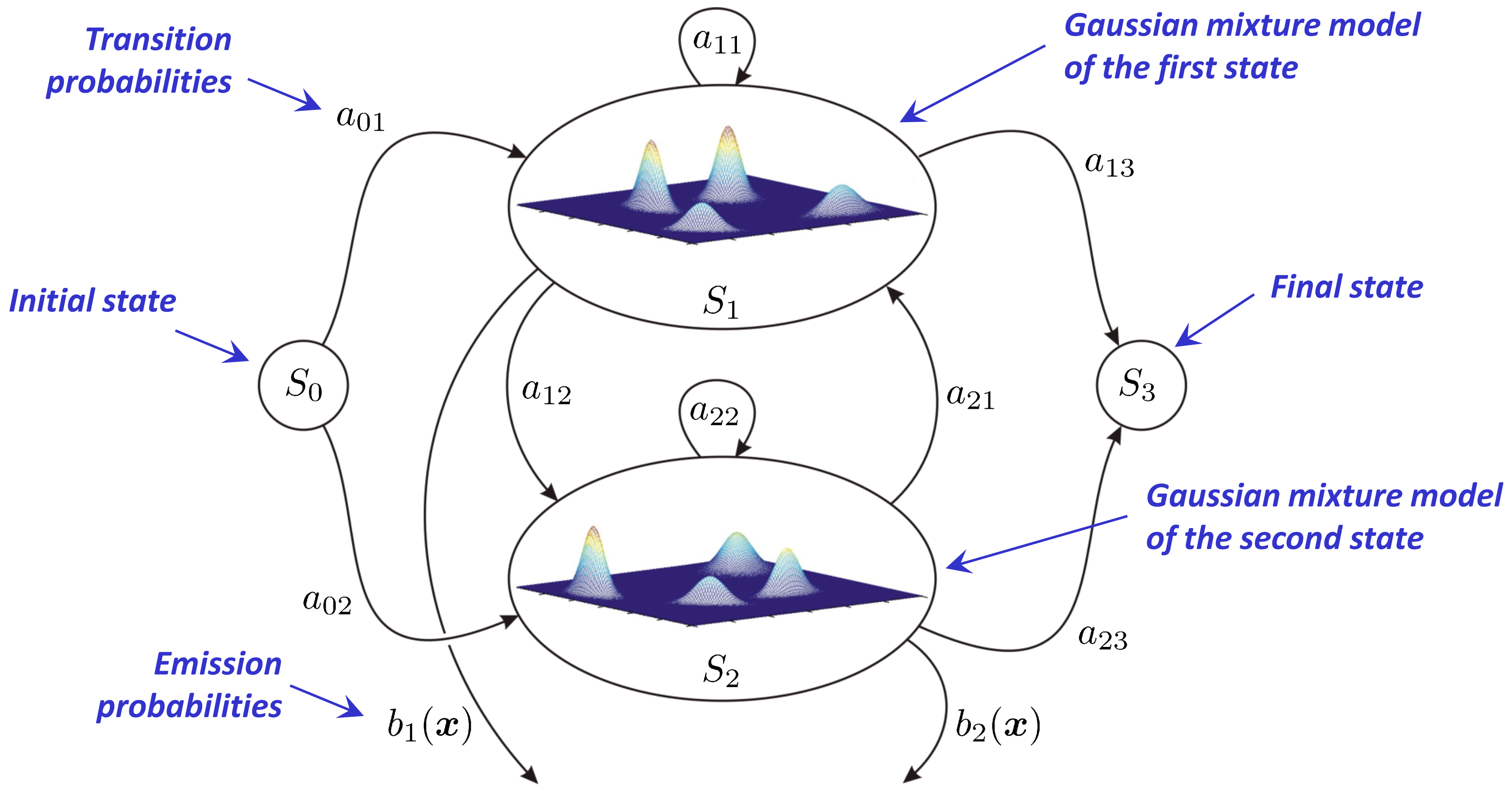


Backtracking of the optimal state sequence

Christian-Albrechts-Universität zu Kiel

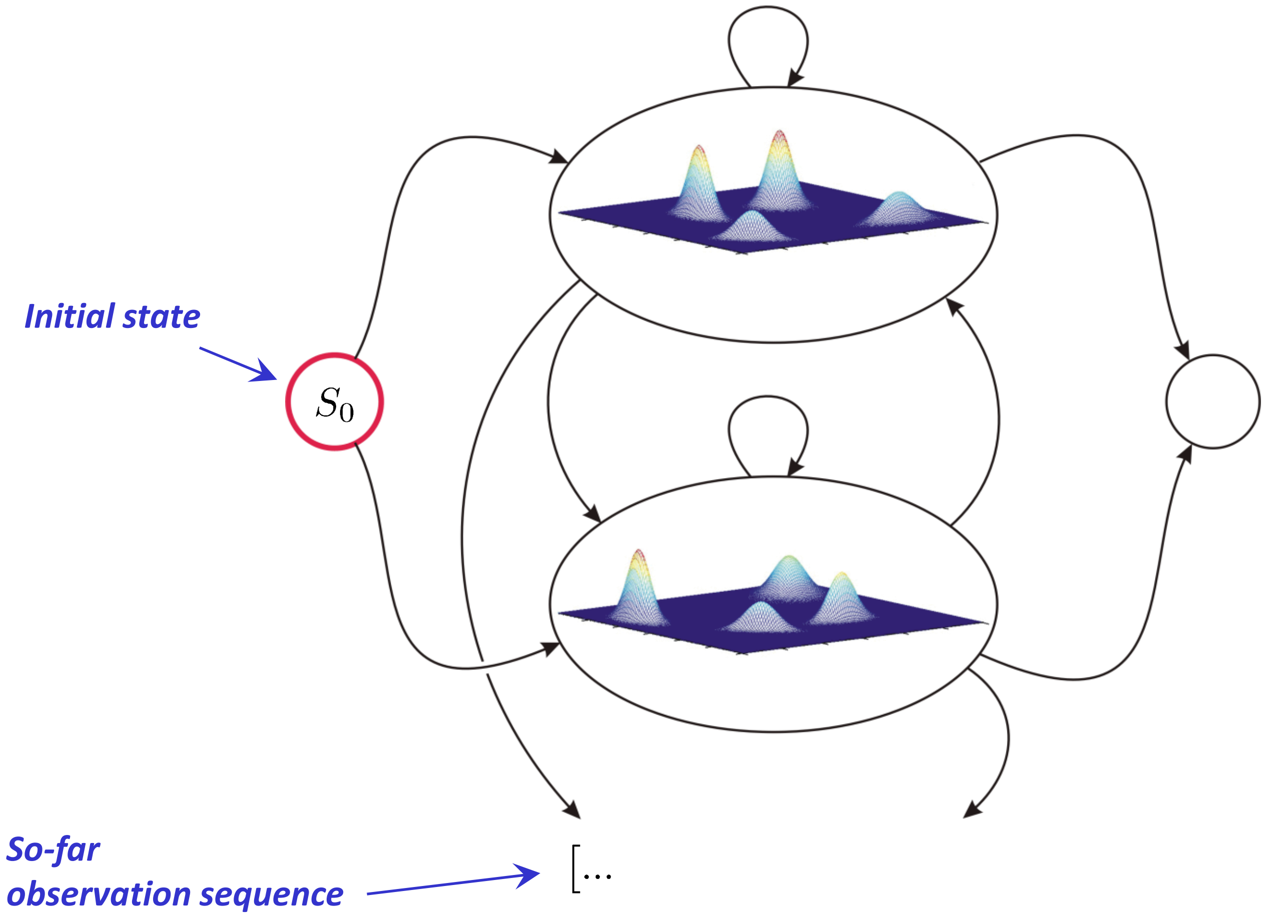
Generating feature vectors using a hidden Markov model – Part 1

Basics



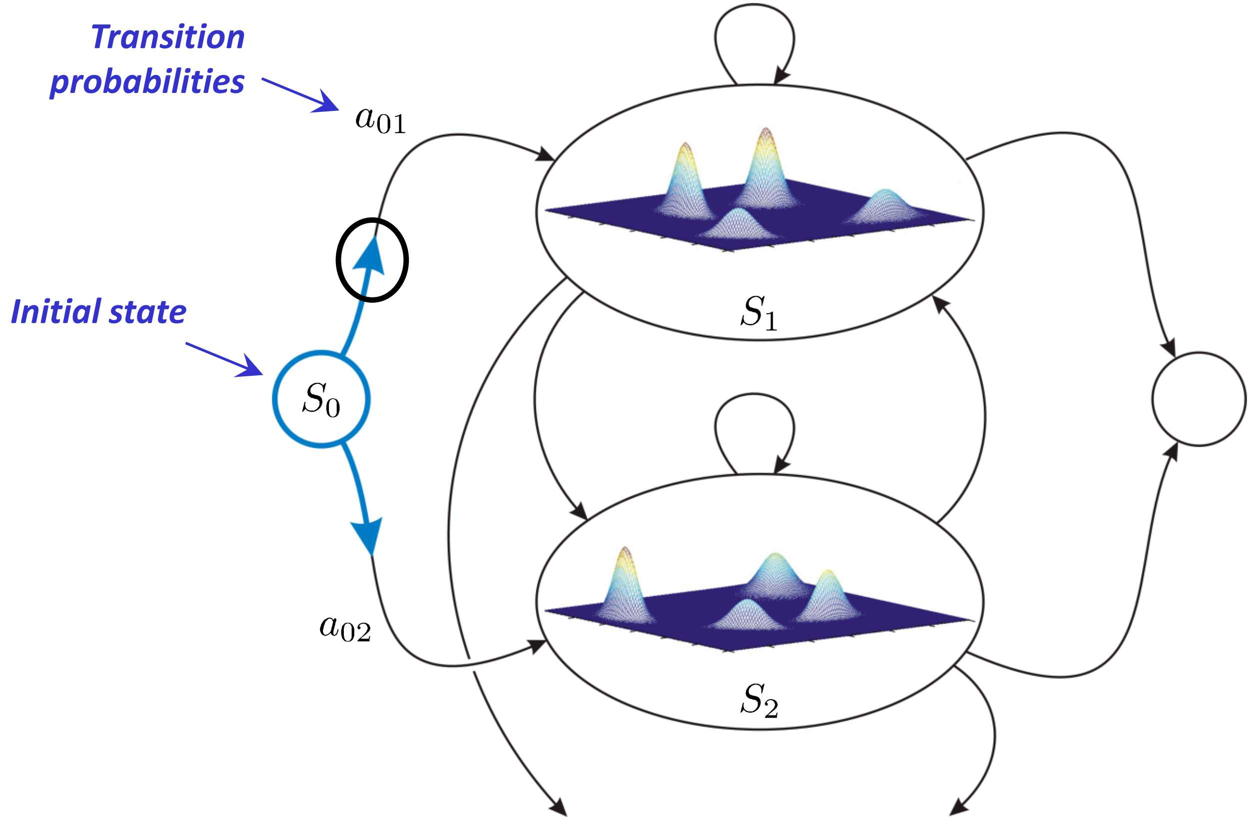
Generating feature vectors using a hidden Markov model – Part 2

Initial state



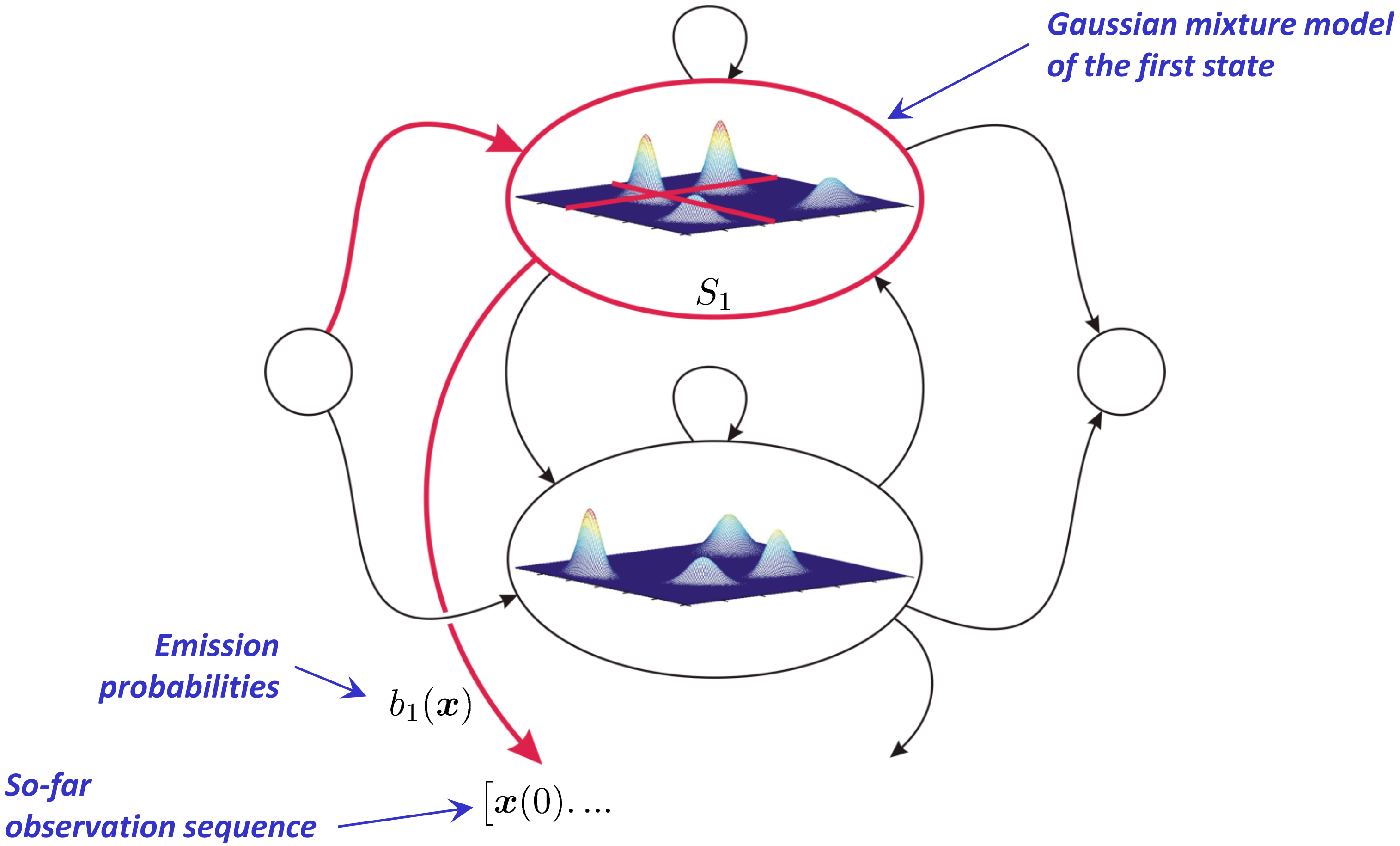
Generating feature vectors using a hidden Markov model – Part 3

Determining the first transition



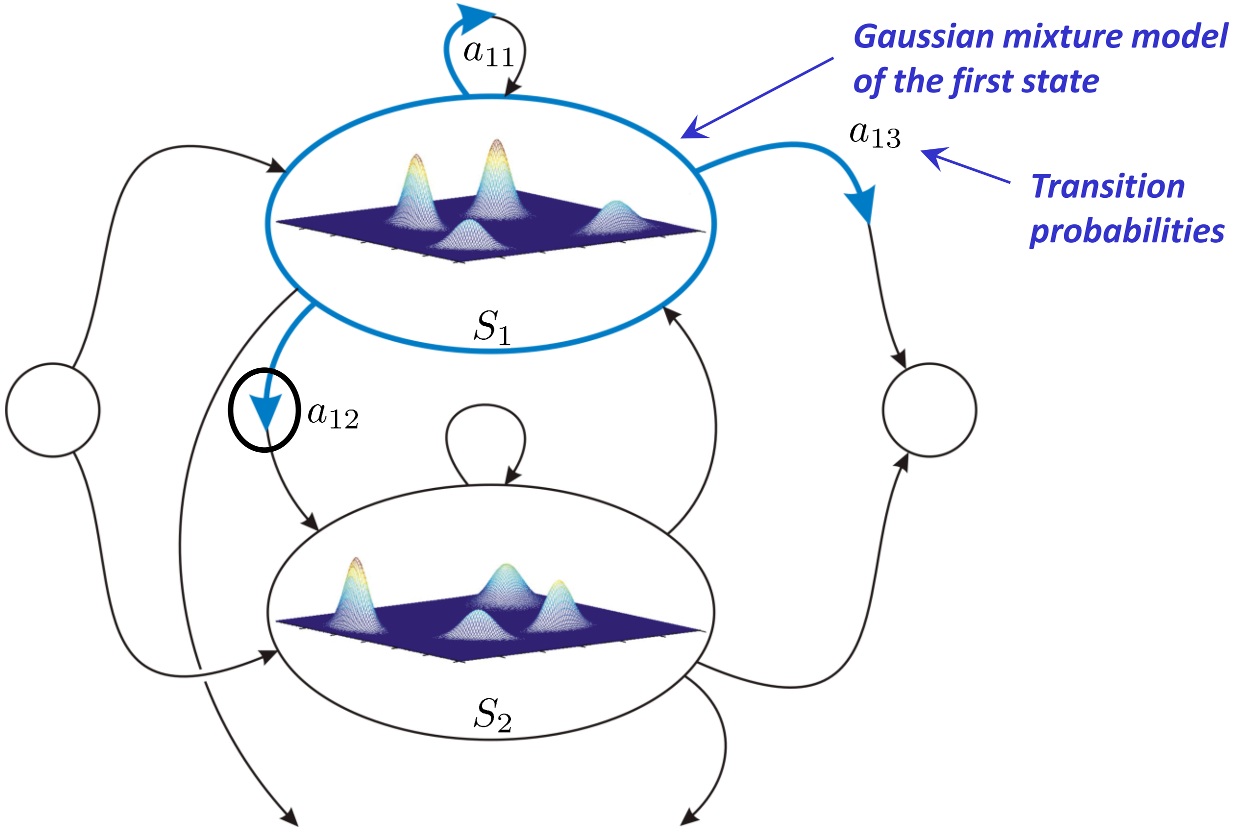
Generating feature vectors using a hidden Markov model – Part 4

Generating the first observation vector



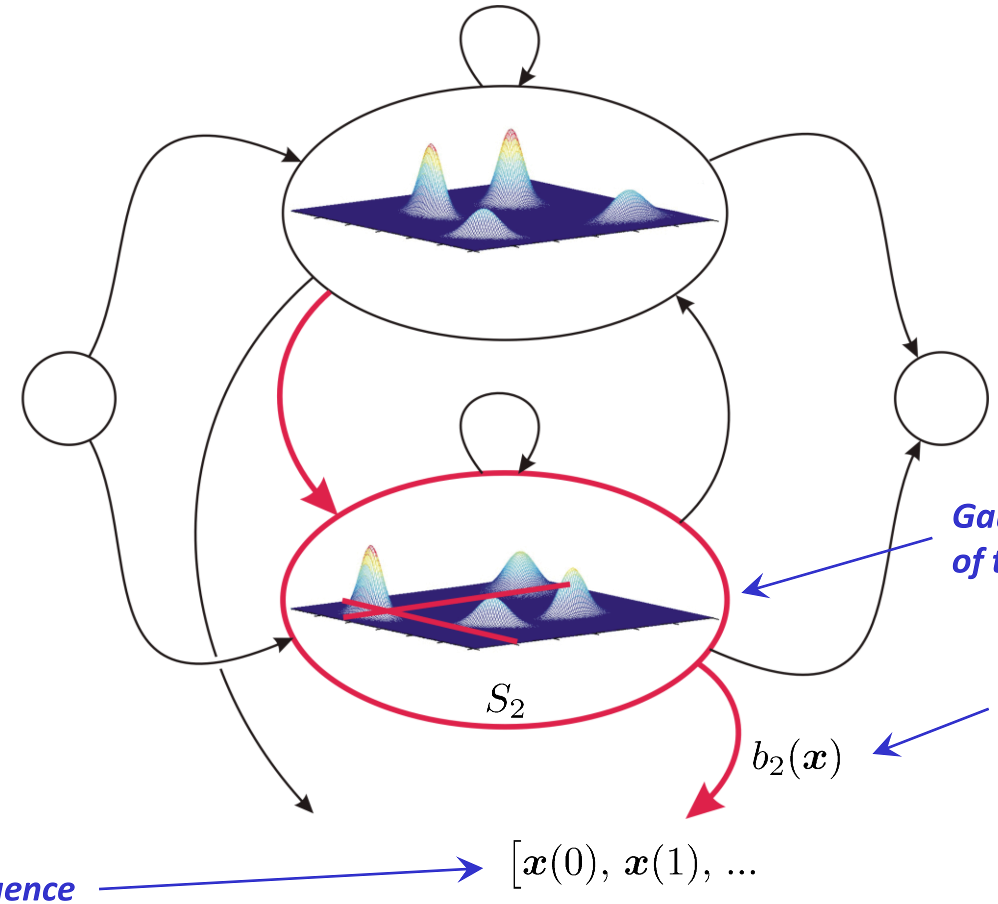
Generating feature vectors using a hidden Markov model – Part 5

Determining the second transition



Generating feature vectors using a hidden Markov model – Part 6

Generation of the second observation vector



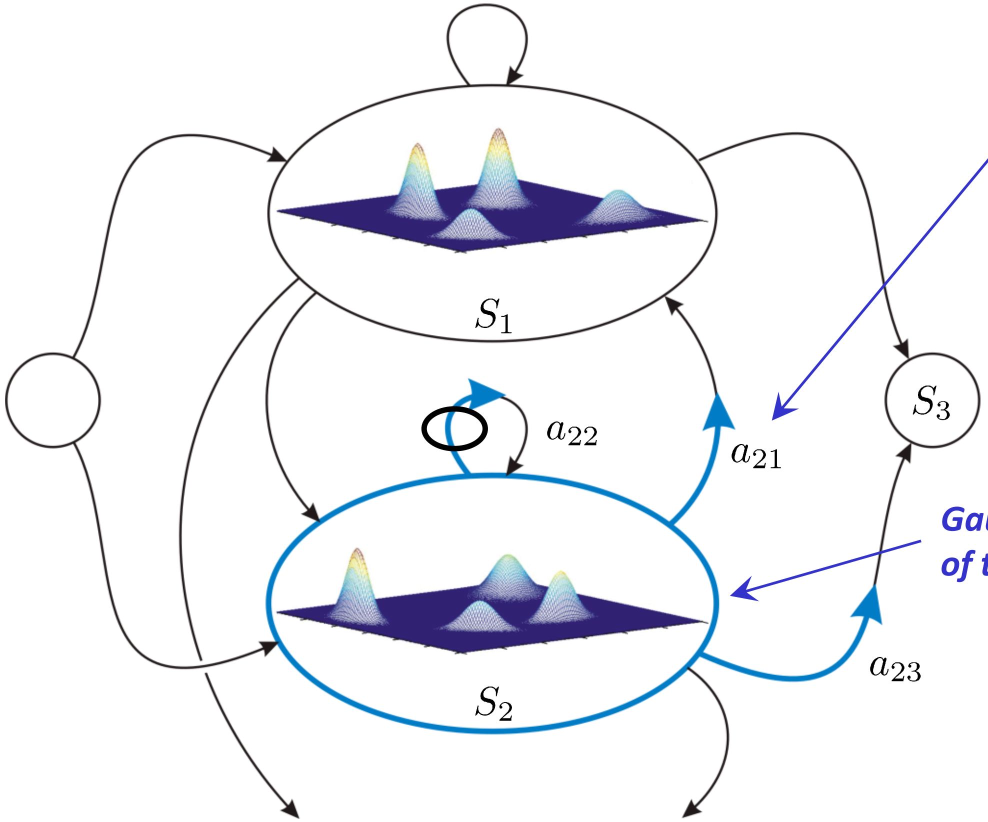
Gaussian mixture model of the second state

Emission probabilities

So-far observation sequence

Generating feature vectors using a hidden Markov model – Part 7

Determining the third transition

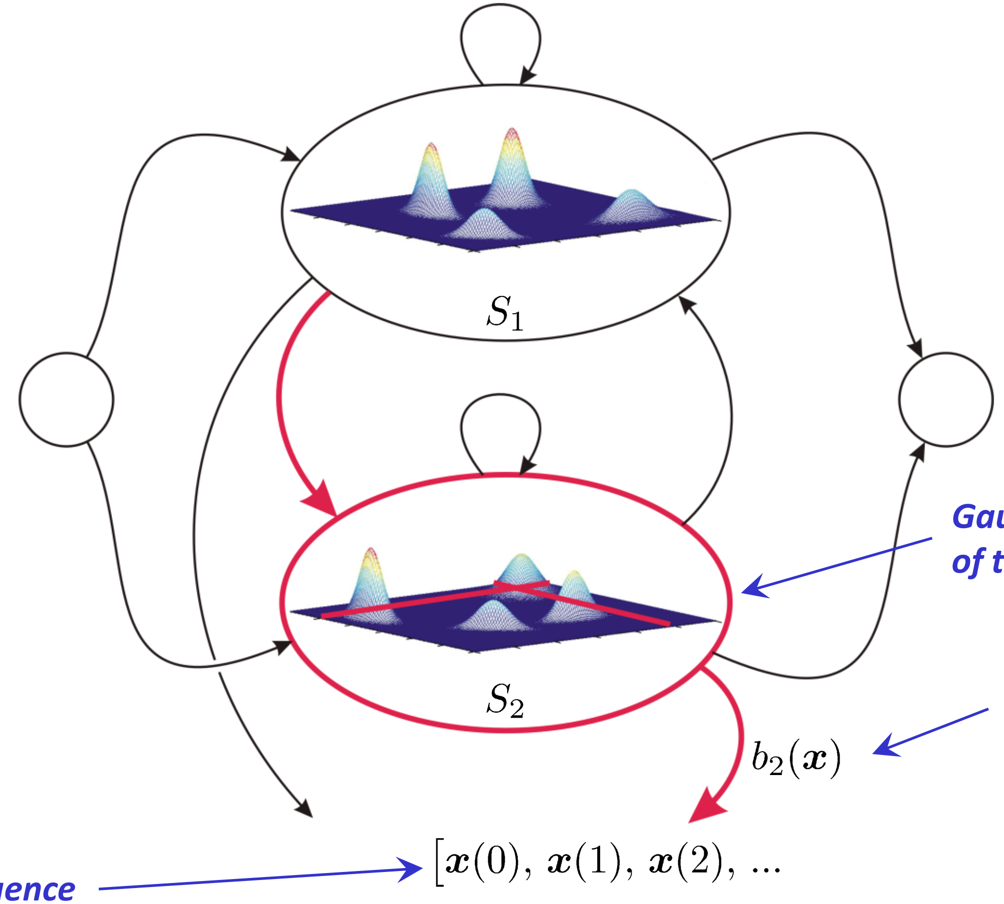


Transition probabilities

Gaussian mixture model of the second state

Generating feature vectors using a hidden Markov model – Part 8

Generation of the third observation vector



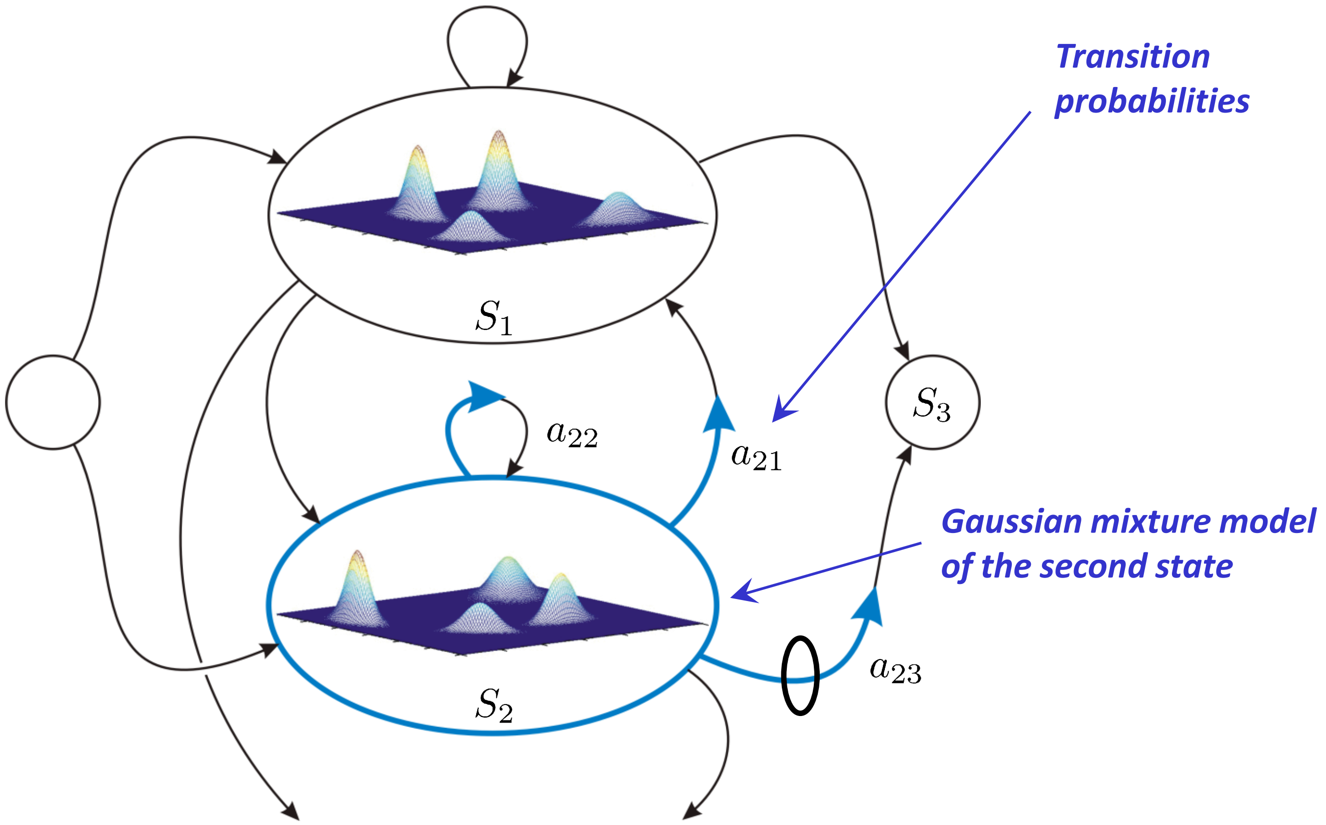
Gaussian mixture model of the second state

Emission probabilities

So-far observation sequence

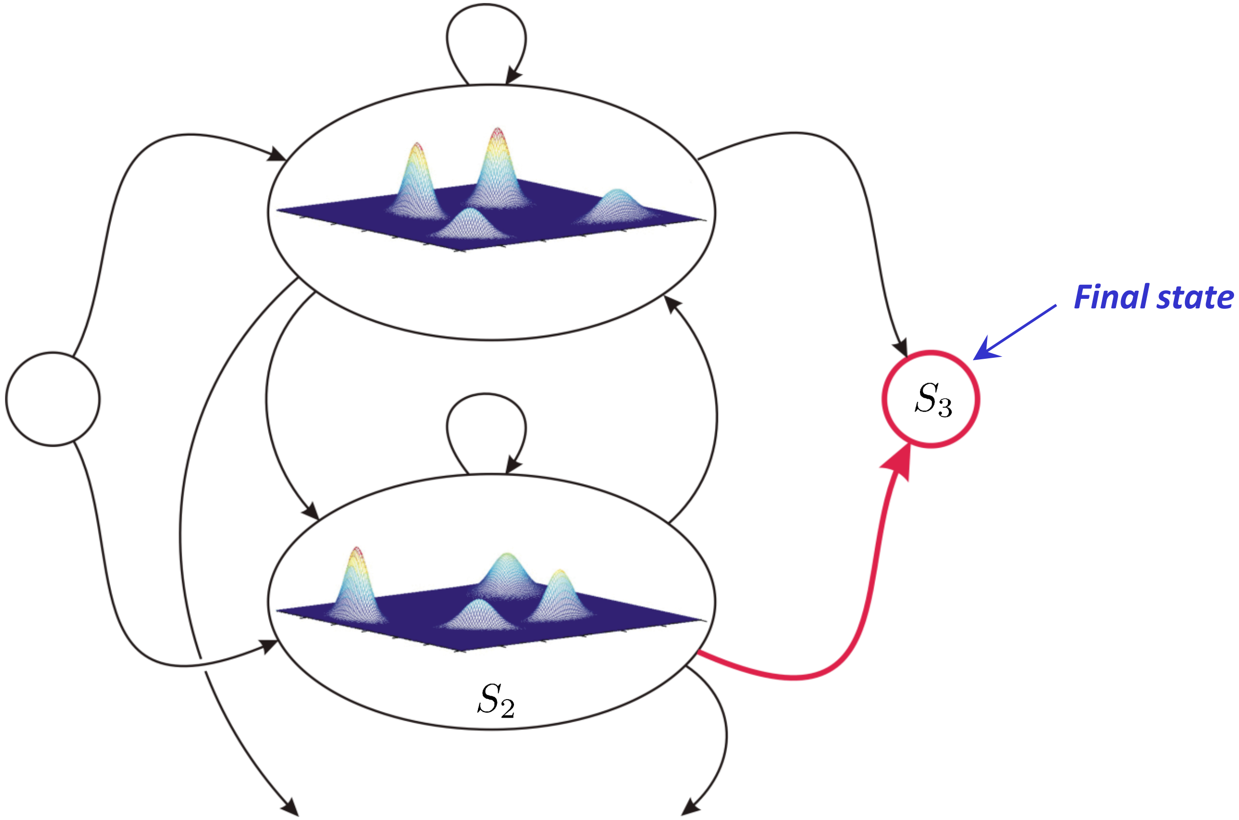
Generating feature vectors using a hidden Markov model – Part 9

Determining the fourth transition



Generating feature vectors using a hidden Markov model – Part 10

Final state

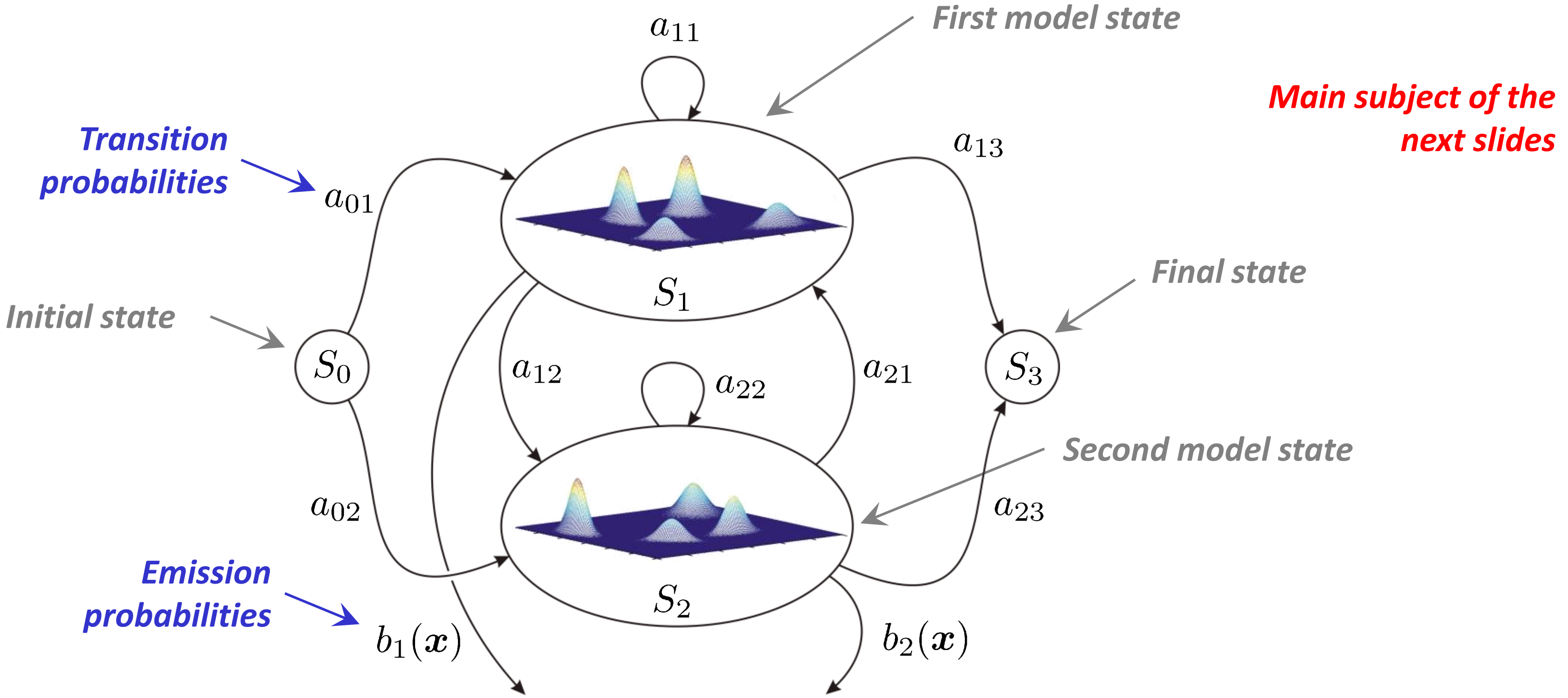


Overall observation sequence

$$\boldsymbol{X} = \big[\boldsymbol{x}(0),\, \boldsymbol{x}(1),\, \boldsymbol{x}(2)\big]$$

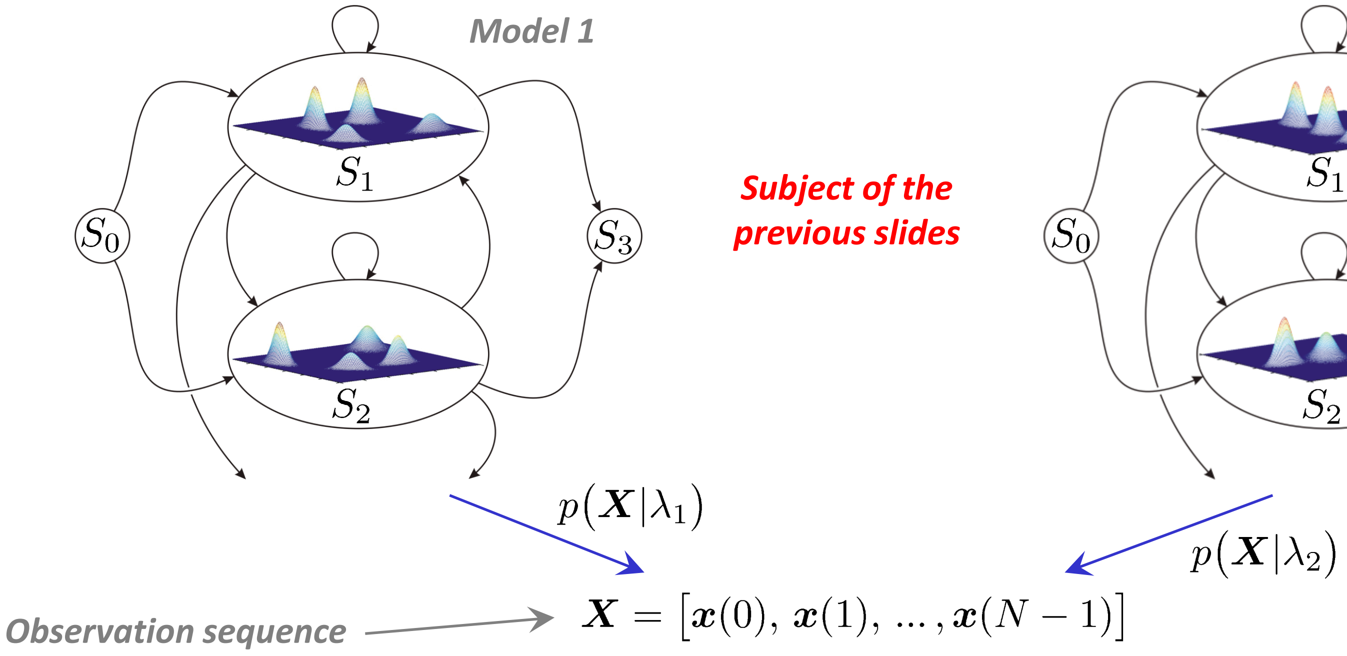
The three problems with hidden Markov models – Part 1

- After the model topology has been defined, the model parameters are to be estimated.



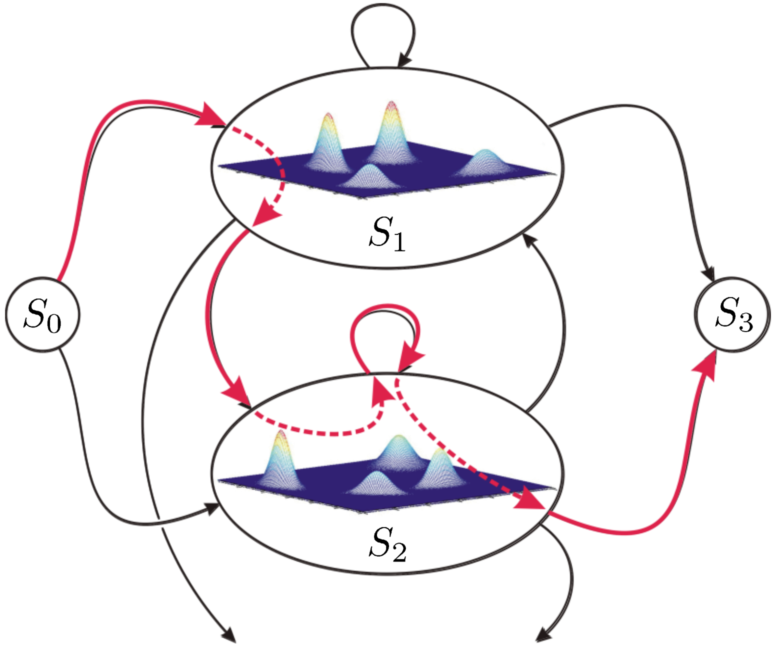
The three problems with hidden Markov models – Part 2

- After the model topology has been defined, the model parameters are to be estimated.
- The probability that a model generates an observed feature sequence has to be calculated in an efficient way.



The three problems with hidden Markov models – Part 3

- After the model topology has been defined, the model parameters are to be estimated.
- The probability that a model generates an observed feature sequence has to be calculated in an efficient way.
- The state sequence that generates the observed feature sequence with highest probability has to calculated efficiently.



$$\hat{\boldsymbol{q}} = \underset{\boldsymbol{q}_j}{\operatorname{argmax}} \left\{ p(\boldsymbol{q}_j, \boldsymbol{X}|\lambda) \right\}$$

Also subject of the previous slides!

Overall observation sequence

$$\boldsymbol{X} = \left[\boldsymbol{x}(0),\, \boldsymbol{x}(1),\, \boldsymbol{x}(2)\right]$$

Lecture Evaluation

- Please help to improve the lecture by filling out our survey ….

Solving the estimation problem – Part 1

Estimation problem

- For one or more given observation sequences $\boldsymbol{X}$ the ***parameters*** (transition and emission probabilities) are to be found in such a way, that

$$p(\boldsymbol{X}|\lambda) \longrightarrow \max .$$

- To do so, we assume that an initial HMM is already existing. This model is ***optimized iteratively***, until a certain optimization criterion is fulfilled or a maximum number of iterations was computed.
- The iteration methods known so far only are able to find ***local maxima***.
- The most common method is based on a maximum likelihood estimation and is called ***Baum-Welch*** or ***forward-backward algorithm***.

Solving the estimation problem – Part 2

Backward probability

- In analogy to the forward probability (see previous slides)

$$f_i(n) \;=\; p(\boldsymbol{X}^{(n)},\, q(n) = S_i|\lambda)$$

we now introduce the ***backward probability***

$$r_i(n) \;=\; p(\boldsymbol{X}_{(n+1)},\, q(n) = S_i|\lambda)$$

The partial observation sequence $\boldsymbol{X}_{(n)}$ describes all observations from the n$^{\text{th}}$ time index up to the end of the sequence,

$$\boldsymbol{X}_{(n)} \;=\; \big[\boldsymbol{x}(n),\, \boldsymbol{x}(n+1),\, \ldots,\, \boldsymbol{x}(T-1)\big].$$

- The backward probability, similar to the forward probability, can be calculated ***recursively***,

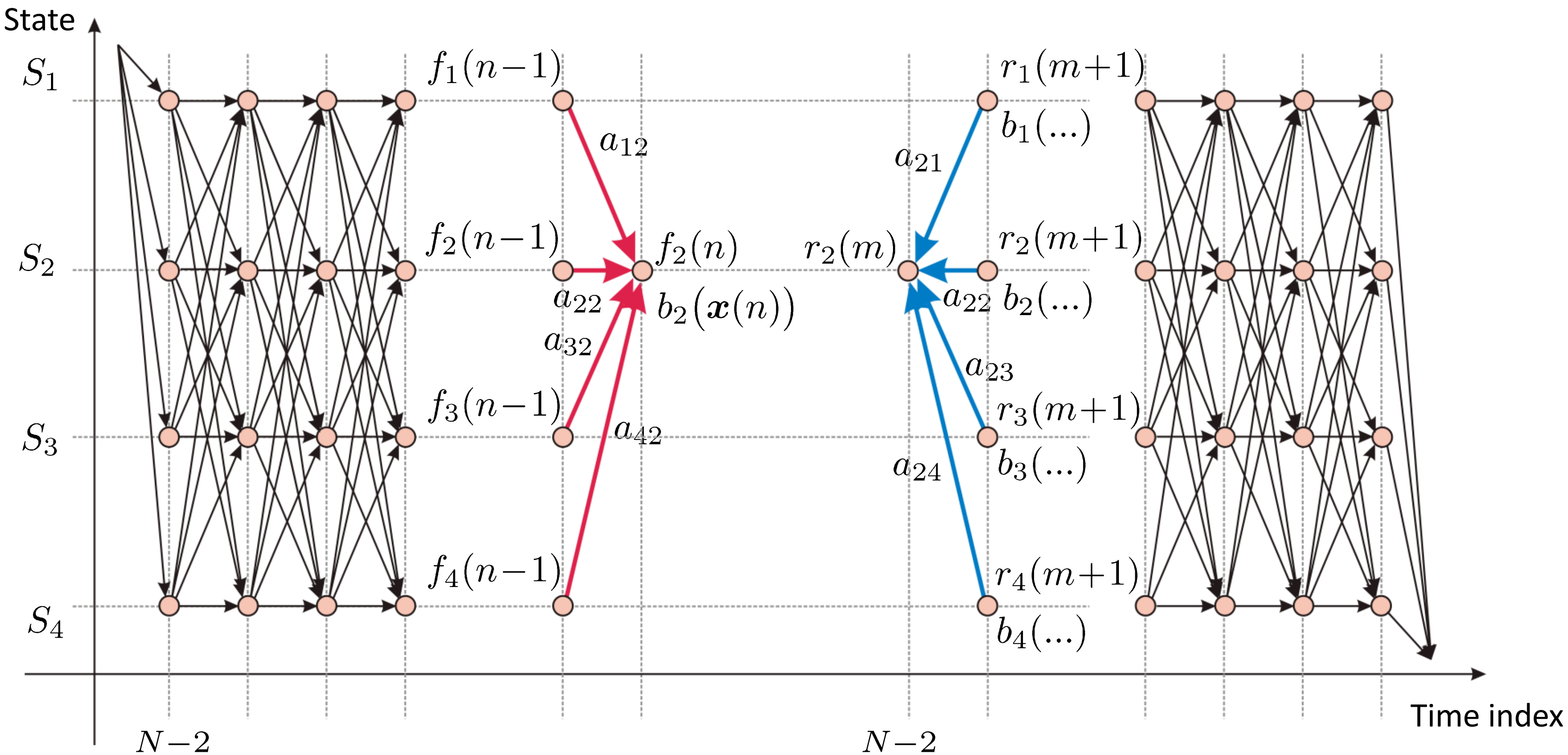
$$r_i(n) \;=\; \sum_{j=1}^{N-2} r_j(n+1)\, b_j\big(\boldsymbol{x}(n+1)\big)\, a_{ij}.$$

- The ***initialization*** is done as follows,

$$r_i(T) = a_{i,N}.$$

Solving the estimation problem – Part 3

Forward and backward probability



$$f_i(n) = \sum_{j=1}^{N-2} f_j(n-1)\, a_{j,i}\, b_i\big(\boldsymbol{x}(n)\big) \qquad r_i(m) = \sum_{j=1}^{N-2} r_j(m+1)\, b_j\big(\boldsymbol{x}(m+1)\big)\, a_{ij}$$

Solving the estimation problem – Part 4

Probability distribution over states

- Using the forward and backward probabilities, we can calculate the probability that the ***state*** S_i ***is active*** at time index n,

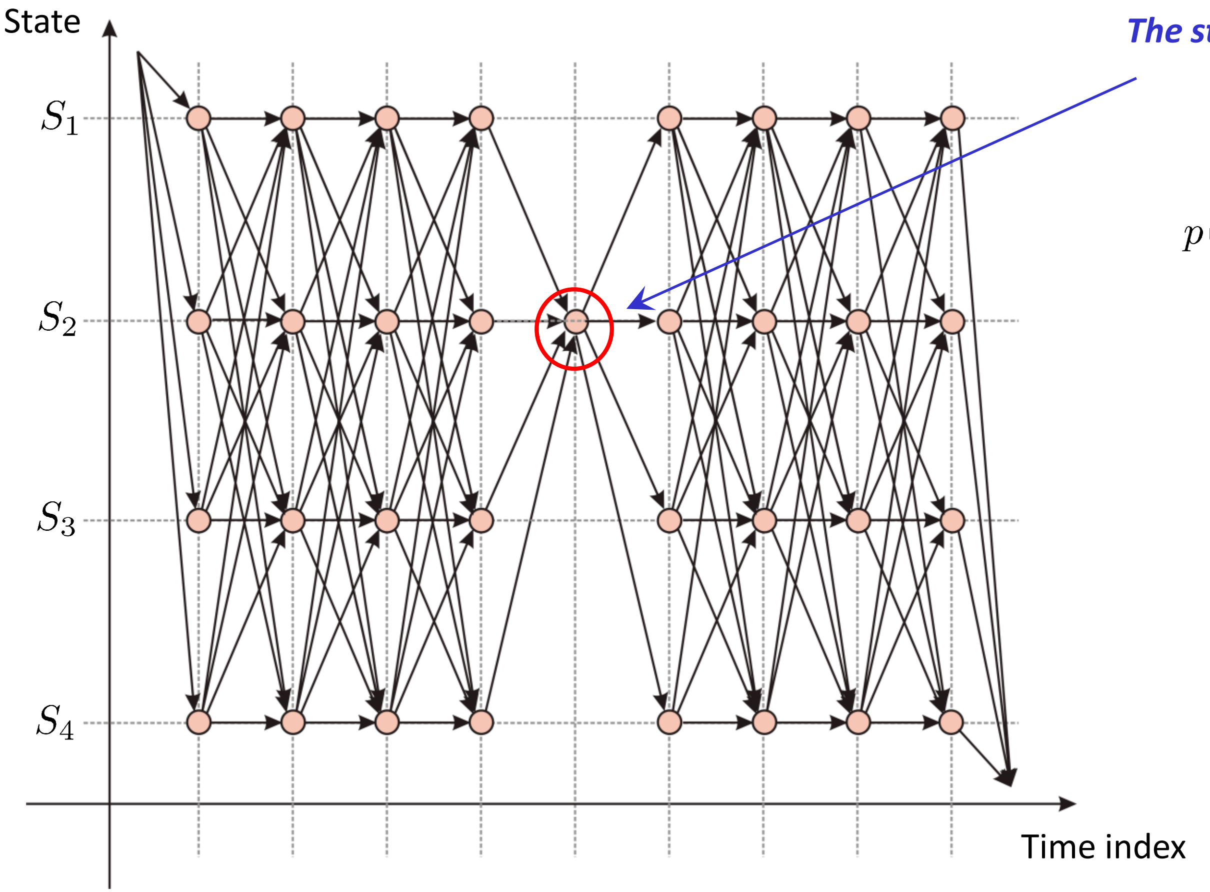
$$\begin{aligned}\gamma_i(n) \quad &= \quad p\big(q(n) = S_i|\boldsymbol{X}, \lambda\big) \quad = \quad \frac{p\big(q(n) = S_i,\ \boldsymbol{X}|\lambda\big)}{p\big(\boldsymbol{X}, \lambda\big)} \\ &= \quad \frac{p\big(q(n) = S_i,\ \boldsymbol{X}^{(n)}|\lambda\big)\ \ p\big(q(n) = S_i,\ \boldsymbol{X}_{(n+1)}|\lambda\big)}{p\big(\boldsymbol{X}|\lambda\big)} \\ &= \quad \frac{f_i(n)\ \ r_i(n)}{p\big(\boldsymbol{X}|\lambda\big)}.\end{aligned}$$

- The "normalization" can be calculated either using the forward or the backward probability,

$$p(\boldsymbol{X}|\lambda) \quad = \quad \sum_{j=1}^{N-2} f_j(T-1)\, a_{j,N-1} \quad = \quad \sum_{j=1}^{N-2} r_j(0)\, b_j\big(\boldsymbol{x}(0)\big)\, a_{0,j}.$$

Solving the estimation problem – Part 5

Probability distribution over states



The state S_i is active at time index n

$$p\big(q(n) = S_i,\, \boldsymbol{X}|\lambda\big) = f_i(n)\ r_i(n)$$

Solving the estimation problem – Part 6

Transition probabilities

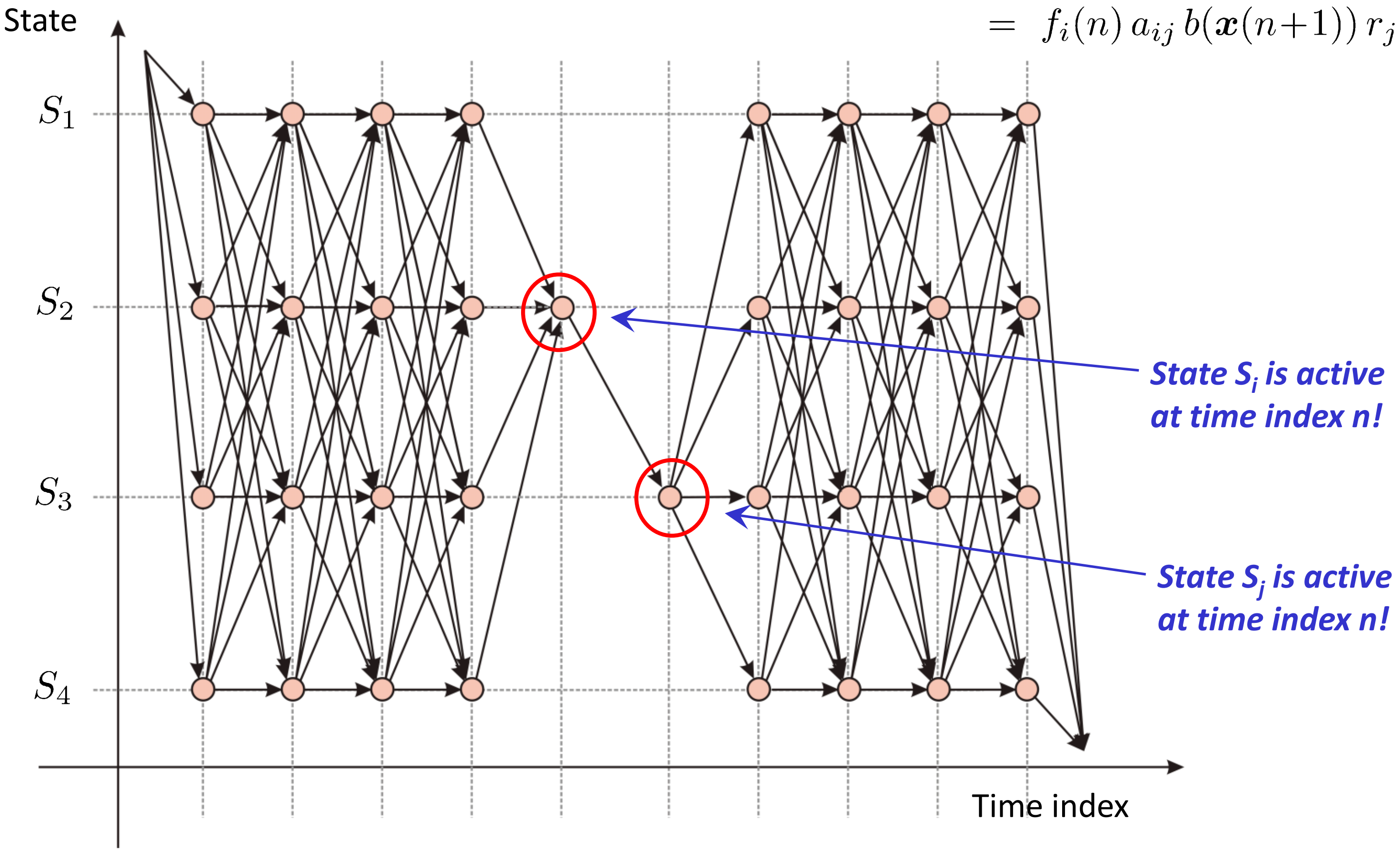
- Using the forward and backward probability, we can also easily calculate the probability that the ***state*** of the hidden Markov model ***changes*** from state S_i to state S_j ***at time index n***,

$$\begin{aligned}
\xi_{i,j}(n) &= p\big(q(n) = S_i,\, q(n+1) = S_j | \boldsymbol{X}, \lambda\big) \\
&= \frac{p\big(q(n) = S_i,\, q(n+1) = S_j,\, \boldsymbol{X} | \lambda\big)}{p\big(\boldsymbol{X}, \lambda\big)} \\
&= \frac{f_i(n)\; a_{ij}\; b_j\big(\boldsymbol{x}(n+1)\big)\, r_j(n+1)}{p\big(\boldsymbol{X}, \lambda\big)}.
\end{aligned}$$

Solving the estimation problem – Part 7

Transition probabilities

$$p\big(q(n) = S_i,\, q(n+1) = S_j,\, \boldsymbol{X}|\lambda\big)$$
$$= f_i(n)\, a_{ij}\, b(\boldsymbol{x}(n+1))\, r_j(n+1)$$



Solving the estimation problem – Part 8

Estimation of the Markov transition probabilities

- For the next iteration, the following transition probabilities are used,

$$\begin{aligned}
a_{i,j} &= \frac{\sum\limits_{n=0}^{T-1} \xi_{i,j}(n)}{\sum\limits_{n=0}^{T-1} \gamma_i(n)}, \quad i,j \in \{1,\, N-2\}, \\
a_{0,j} &= \gamma_j(0), \\
a_{i,N-1} &= \frac{\gamma_i(T-1)}{\sum\limits_{n=0}^{T-1} \gamma_i(n)}, \quad i \in \{1,\, N-2\}
\end{aligned}$$

Expected average number of state transitions from state S_i to state S_j

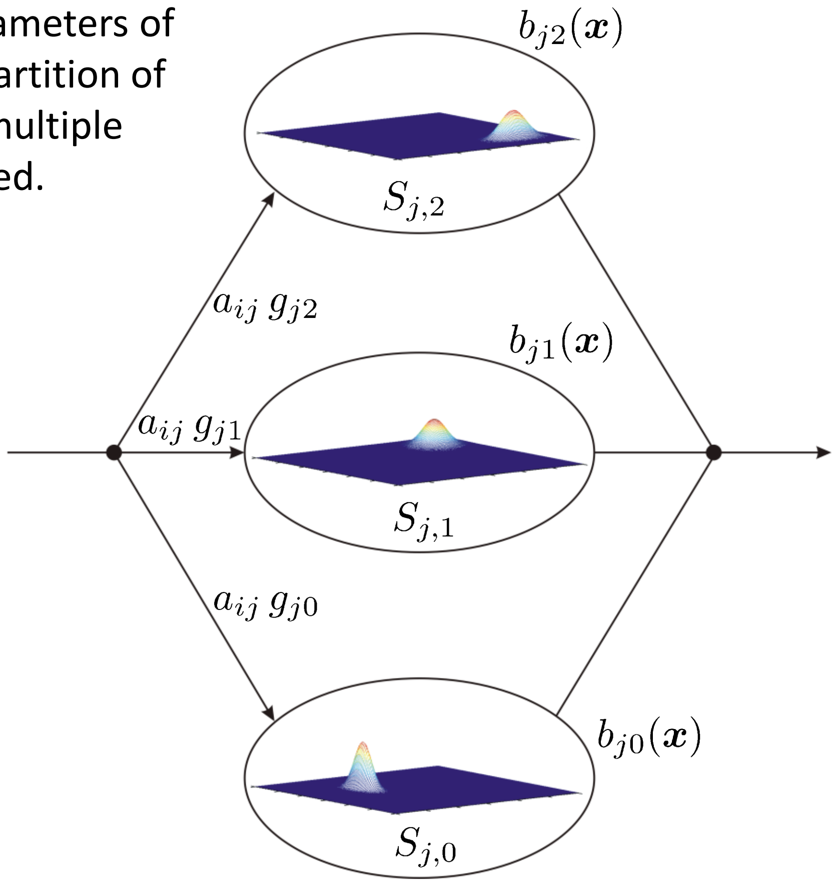
Expected average number of state transitions that start in state S_i

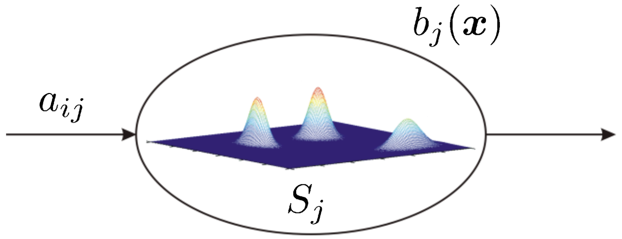
- Additionally, the parameters mentioned above are to be calculated based on multiple observation sequences ***X*** and averaged before being used in the next step.

Solving the estimation problem – Part 9

Emission probabilities

- In order to determine the individual parameters of the Gaussian densities, in a first step a partition of the states with multiple Gaussians into multiple states with just one Gaussian is performed.





$$
\begin{aligned}
b_j(\boldsymbol{x}) &= \sum_{k=0}^{K-1} g_{jk}\, \mathcal{N}\big(\boldsymbol{x}|\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}\big) \\
&= \sum_{k=0}^{K-1} g_{jk}\, b_{jk}(\boldsymbol{x})
\end{aligned}
$$

Solving the estimation problem – Part 10

Emission probabilities

- In analogy to the first approach, individual transition probabilities can be calculated for this extended model,

$$\zeta_{i,j,k}(n) \quad = \quad p\big(q(n) = S_i,\, q(n+1) = S_j, \boldsymbol{x}(n+1) \mapsto \mathcal{N}_{jk} | \boldsymbol{X}, \lambda\big)$$

Probability that a transition from state S_i into state S_j was performed at time index n while the k-th Gaussian of the state S_j was creating the observation vector.

- These can again be expressed by forward and backward probabilities,

$$\zeta_{i,j,k}(n) \quad = \quad \frac{f_i(n)\, a_{ij}\, g_{jk}\, b_{jk}\big(\boldsymbol{x}(n+1)\big)\, r_j(n+1)}{p\big(\boldsymbol{X}|\lambda\big)}.$$

Solving the estimation problem – Part 11

Emission probabilities

- Summing all transition probabilities over the outgoing states results in the probability that the *k*-th Gaussian of the *j*-th state generated the observed vector at time index *n*,

$$\begin{aligned} \zeta_{j,k}(n) &= \sum_{i=1}^{N-1} \zeta_{i,j,k}(n) \\ &= \frac{\sum\limits_{i=1}^{N-1} f_i(n)\, a_{ij}\, g_{jk}\, b_{jk}\big(\boldsymbol{x}(n+1)\big)\, r_j(n+1)}{p\big(\boldsymbol{X}|\lambda\big)}. \end{aligned}$$

- Now, analogously to the “main transition probabilities“, also the GMM parameters can be determined by iteration.

Solving the estimation problem – Part 12

Adaption of the GMM parameters

- The emission probability was defined as follows,

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

- The adaptation of the weights is done as follows,

$$g_{jk} \quad = \quad \frac{\sum\limits_{n=0}^{T-1} \zeta_{jk}(n)}{\sum\limits_{n=0}^{T-1} \gamma_j(n)}.$$

Average number of transitions from the outgoing state S_j to the incoming state S_i

Average number of state transitions that start in state S_j

- The adaption of the averages vectors is done as follows,

$$\boldsymbol{\mu}_{jk} \quad = \quad \frac{\sum\limits_{n=0}^{T-1} \zeta_{jk}(n)\, \boldsymbol{x}(n)}{\sum\limits_{n=0}^{T-1} \zeta_{jk}(n)}.$$

Solving the estimation problem – Part 13

Adaption of the GMM parameters

- The adaptation of the covariance matrices is performed as follows,

$$\boldsymbol{\Sigma}_{jk} \quad = \quad \frac{\sum\limits_{n=0}^{T-1} \zeta_{jk}(n) \left[\boldsymbol{x}(n) - \boldsymbol{\mu}_{jk}\right] \left[\boldsymbol{x}(n) - \boldsymbol{\mu}_{jk}\right]^{\mathrm{T}}}{\sum\limits_{n=0}^{T-1} \zeta_{jk}(n)}.$$

Solving the estimation problem – Part 14

Viterbi training

- The method to estimate the model parameters that was described above is called Baum-Welch algorithm. It is a special case of the EM algorithm that was described in the GMM lecture.
- Alternatively, the so-called Viterbi training can be applied. To do so, in a first step the state sequence

$$\hat{\boldsymbol{q}} = \underset{\boldsymbol{q}_j}{\operatorname{argmax}} \left\{ p(\boldsymbol{q}_j, \boldsymbol{X} | \lambda) \right\}$$

  with the highest probability is computed.
- Then it is assumed that this path was taken with “certain” probability, i.e., it holds

$$\gamma_i(n) \quad = \quad \begin{cases} 1, & \text{if } \hat{q}(n) = S_i, \\ 0, & \text{else.} \end{cases}$$

$$\xi_{i,j}(n) \quad = \quad \begin{cases} 1, & \text{if } \hat{q}(n) = S_i \text{ and } \hat{q}(n+1) = S_j \\ 0, & \text{else.} \end{cases}$$

Solving the estimation problem – Part 15

Viterbi training

- For the internal transitions, the following consequently holds,

$$\zeta_{i,j,k}(n) \;=\; \begin{cases} 1, & \text{if } \hat{q}(n) = S_i \text{ and } \hat{q}(n+1) = S_j \\ & \qquad \text{and according to a Viterbi search the internal state } k \\ & \qquad \text{was selected,} \\ 0, & \text{else.} \end{cases}$$

- The subsequent iterations to optimize the model parameters are performed as described at the Baum-Welch algorithm.
- Similar to the Baum-Welch algorithm, the iterations are performed until the probability that the model generates the observation sequence is no longer increasing significantly or the maximum number of iterations is reached.

Solving the estimation problem – Part 16

Initializing a hidden Markov model

- In a first step, the number of states and their topology is defined (forbidden transitions are marked, i.e. their probability is set to zero).

- Per state, just one Gaussian distribution is used.

- While the training is running, the number of Gaussian distributions is gradually increased. For example, the Gaussian distributions are doubled and initialized as follows,

$$\begin{aligned} g \quad &\longrightarrow \quad g_0 = \frac{g}{2},\ g_1 = \frac{g}{2}, \\ \boldsymbol{\mu} \quad &\longrightarrow \quad \boldsymbol{\mu}_0 = \boldsymbol{\mu} + 0.2\,\sqrt{\mathrm{diag}\{\boldsymbol{\Sigma}\}},\ \boldsymbol{\mu}_0 = \boldsymbol{\mu} - 0.2\,\sqrt{\mathrm{diag}\{\boldsymbol{\Sigma}\}}, \\ \boldsymbol{\Sigma} \quad &\longrightarrow \quad \boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma},\ \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}. \end{aligned}$$

- This is repeated until the probability that the model generates the training sequences is no longer increased significantly or a maximum number of parameters is reached.

Summary and Outlook

Summary:

- Motivation
- Basics
  - The „hidden“ part of the model
  - The „inner“ random processes
- Basic problems of Hidden Markov Models
  - Efficient computation of the probabilities of state sequences
  - Efficient computation of the most probable sequence
  - Computation (estimation) of the parameters of the model

Next part:

- Explainable artificial intelligence