Human-Oriented Robotics

Temporal Reasoning

Part 1/3

Kai Arras

Social Robotics Lab, University of Freiburg

Winter term 2014/2015

Temporal Reasoning

Contents

Temporal Reasoning

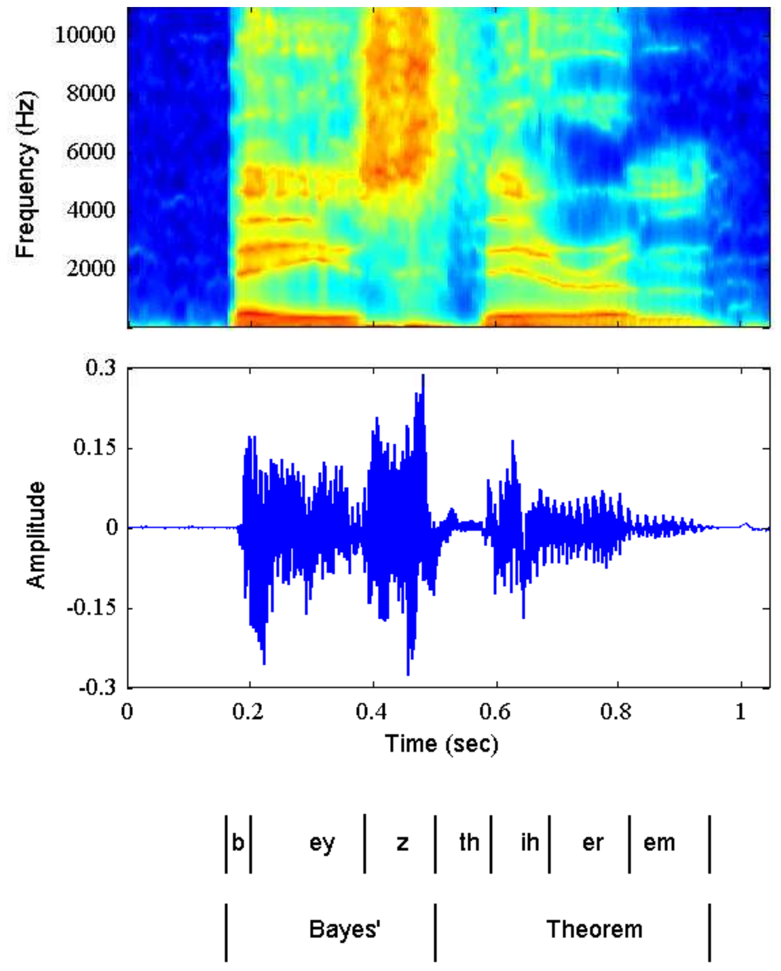
Introduction: Sequential Data (from Ch. 3)

- Sequential data often arise through **measurements of time series**, for example:
  - Rainfall measurements on successive days at a particular location
  - Daily currency exchange rates
  - Acoustic features at successive time frames used for speech recognition
  - A human's arm and hand movements used for sign language understanding
- Other forms of sequential data, e.g. **over space**, exist as well to which the considered models equally apply
- In applications, we typically wish to be able to **predict the next value** given observations of the previous values (e.g. in financial forecasting)
- We expect that **recent observations** are likely to be **more informative** than **more historical observations**

Temporal Reasoning

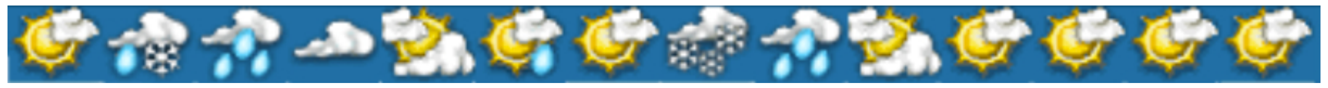
Introduction: Sequential Data (from Ch. 3)

- This is the case when **successive values** in time series are **correlated**
- **Examples:** spectrogram of the spoken words, weather



Source [2]

→ Time

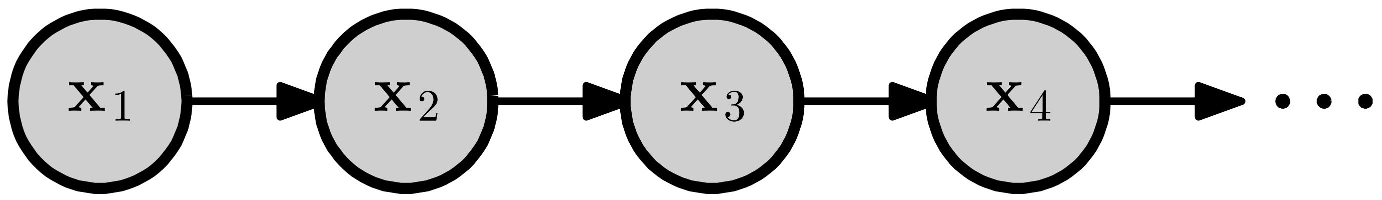


→ Time

Temporal Reasoning

Introduction: Markov Models (from Ch. 3)

- Consider a model that postulates dependencies of future observations **on all previous observations**. Such a model would be impractical because its complexity would **grow without limits** as the number of observations increases
- This leads us to consider **Markov Models**



$$\begin{tikzcd} \mathbf{x}_1 \arrow[r] & \mathbf{x}_2 \arrow[r] & \mathbf{x}_3 \arrow[r] & \mathbf{x}_4 \arrow[r] & \cdots \end{tikzcd}$$

- Markov models assume that future **predictions** are independent of all **but the most recent observations**

Temporal Reasoning

Introduction: Markov Models (from Ch. 3)

- Formally, we recall the **chain rule**

$$p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K) = \prod_{k=1}^{K} p(\mathbf{x}_k \mid \mathbf{x}_1, \dots, \mathbf{x}_{k-1})$$

- If we now assume that each of the conditional distributions on the right hand side is independent of all previous observations **except the most recent one,**

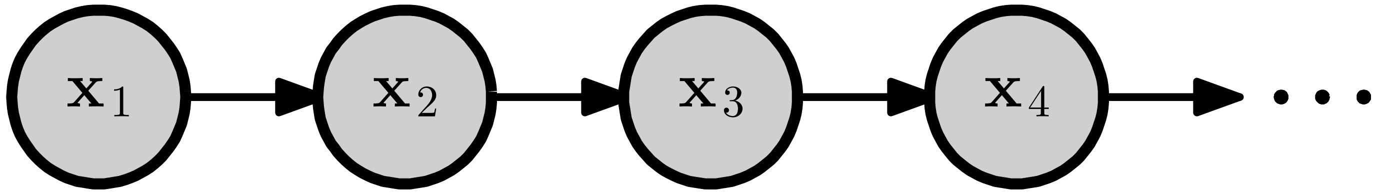
$$\begin{aligned} p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K) &= \prod_{k=1}^{K} p(\mathbf{x}_k | \mathbf{x}_1, \dots, \mathbf{x}_{k-1}) \\ &= p(\mathbf{x}_1)\, p(\mathbf{x}_2|\mathbf{x}_1)\, p(\mathbf{x}_3|\cancel{\mathbf{x}_1}, \mathbf{x}_2)\, p(\mathbf{x}_4|\cancel{\mathbf{x}_1}, \cancel{\mathbf{x}_2}, \mathbf{x}_3) \cdots \\ &\quad \cdot p(\mathbf{x}_K|\cancel{\mathbf{x}_1}, \cancel{\mathbf{x}_2}, \cancel{\dots}, \mathbf{x}_{K-1}) \end{aligned}$$

Temporal Reasoning

Introduction: Markov Models (from Ch. 3)

we obtain the **first-order Markov chain**

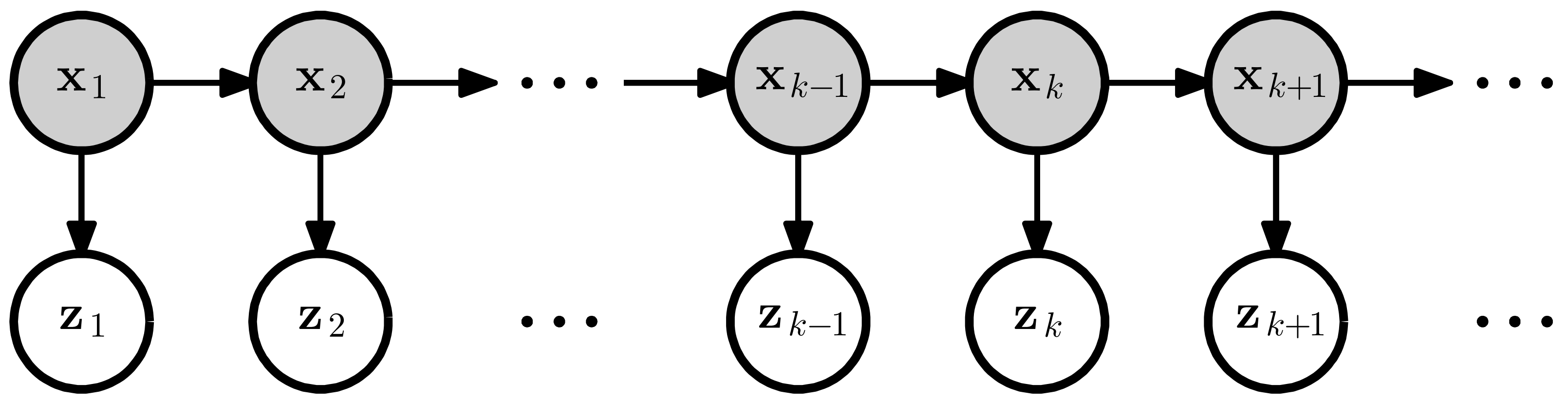
$$
\begin{aligned}
p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K) &= \prod_{k=1}^{K} p(\mathbf{x}_k | \mathbf{x}_1, \ldots, \mathbf{x}_{k-1}) \\
&= p(\mathbf{x}_1)\, p(\mathbf{x}_2|\mathbf{x}_1)\, p(\mathbf{x}_3|\cancel{\mathbf{x}_1}, \mathbf{x}_2)\, p(\mathbf{x}_4|\cancel{\mathbf{x}_1}, \cancel{\mathbf{x}_2}, \mathbf{x}_3) \cdots \\
&\quad \cdot p(\mathbf{x}_K|\cancel{\mathbf{x}_1}, \cancel{\mathbf{x}_2}, \cancel{\ldots}, \mathbf{x}_{K-1}) \\
&= p(\mathbf{x}_1) \prod_{k=2}^{K} p(\mathbf{x}_k|\mathbf{x}_{k-1})
\end{aligned}
$$



Temporal Reasoning

Introduction: State Space Model (from Ch. 3)

- Let's add **latent or hidden variables** to our model, one for each random variable and let the latent variables form a Markov chain



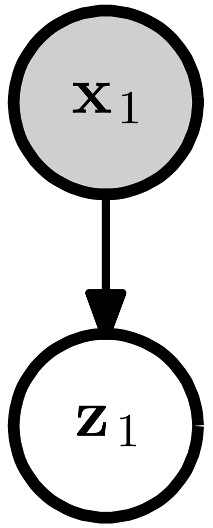
- Notice the change in notation: we denote **latent** variables by $\mathbf{x}$ and **observations** by $\mathbf{z}$ (this notation is widely used in particular for LDS)
- It is sometimes common to **shade the nodes** of latent variables in the graphical representation

Temporal Reasoning

Introduction: State Space Model (from Ch. 3)

- In this model, we view the model to describe a system that **evolves on its own**, with **observations of it** occurring in a **separate** process
- This model is called **state space model** or **state observation model**

- Hidden variables $\mathbf{x}$ are often called **states.** They are un-observable and typically what we want to estimate
- Variables $\mathbf{z}$ are called **observations** (or "evidence variables"). It is only through the observations that we can indirectly estimate the $\mathbf{x}$'s
- Observations may be of different type and dimensionality than the states
- We assume the interval Δt to be **fixed**. Thus we can label time by integer time indices k (or t, i)



Temporal Reasoning

Introduction: State Space Model (from Ch. 3)

- In addition to the independence assumption of the first-order Markov model, we assume that **observations** at time index k are **conditionally independent of the entire state sequence** given the **state variable** at time index k
- The **joint distribution** of this model is derived as follows

$$\begin{aligned} p(\mathbf{x}_1, \ldots, \mathbf{x}_K, \mathbf{z}_1, \ldots, \mathbf{z}_K) \;=\; & p(\mathbf{x}_1)\, p(\mathbf{x}_2|\mathbf{x}_1)\, p(\mathbf{x}_3|\mathbf{x}_1, \mathbf{x}_2) \cdots \\ & \cdot\, p(\mathbf{x}_K|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{K-1}) \\ & \cdot\, p(\mathbf{z}_1|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K)\, p(\mathbf{z}_2|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K, \mathbf{z}_1) \cdots \\ & \cdot\, p(\mathbf{z}_K|\mathbf{x}_1, \ldots, \mathbf{x}_K, \mathbf{z}_1, \ldots, \mathbf{z}_{K-1}) \end{aligned}$$

Temporal Reasoning

Introduction: State Space Model (from Ch. 3)

- In addition to the independence assumption of the first-order Markov model, we assume that **observations** at time index k are **conditionally independent of the entire state sequence** given the **state variable** at time index k
- The **joint distribution** of this model is derived as follows

$$
\begin{aligned}
p(\mathbf{x}_1, \dots, \mathbf{x}_K, \mathbf{z}_1, \dots, \mathbf{z}_K) = \; & p(\mathbf{x}_1)\, p(\mathbf{x}_2 | \mathbf{x}_1)\, p(\mathbf{x}_3 | \cancel{\mathbf{x}_1}, \mathbf{x}_2) \cdots \\
& \cdot p(\mathbf{x}_K | \cancel{\mathbf{x}_1}, \cancel{\mathbf{x}_2}, \dots, \mathbf{x}_{K-1}) \\
& \cdot p(\mathbf{z}_1 | \mathbf{x}_1, \cancel{\mathbf{x}_2}, \dots, \cancel{\mathbf{x}_K})\, p(\mathbf{z}_2 | \cancel{\mathbf{x}_1}, \mathbf{x}_2, \dots, \cancel{\mathbf{x}_K}, \cancel{\mathbf{z}_1}) \cdots \\
& \cdot p(\mathbf{z}_K | \cancel{\mathbf{x}_1}, \dots, \mathbf{x}_K, \cancel{\mathbf{z}_1}, \dots, \cancel{\mathbf{z}_{K-1}})
\end{aligned}
$$

Temporal Reasoning

Introduction: State Space Model (from Ch. 3)

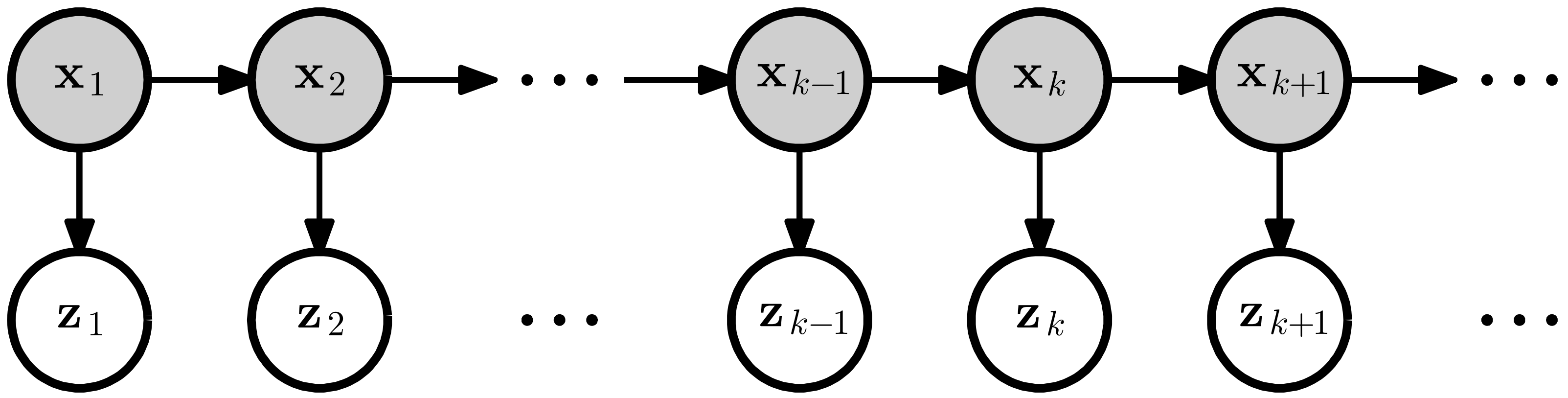
- In addition to the independence assumption of the first-order Markov model, we assume that **observations** at time index k are **conditionally independent of the entire state sequence** given the **state variable** at time index k
- The **joint distribution** of this model is derived as follows

$$\begin{aligned} p(\mathbf{x}_1, \dots, \mathbf{x}_K, \mathbf{z}_1, \dots, \mathbf{z}_K) &= p(\mathbf{x}_1)\, p(\mathbf{x}_2|\mathbf{x}_1)\, p(\mathbf{x}_3|\cancel{\mathbf{x}_1}, \mathbf{x}_2) \cdots \\ &\quad \cdot p(\mathbf{x}_K|\cancel{\mathbf{x}_1}, \cancel{\mathbf{x}_2}, \dots, \mathbf{x}_{K-1}) \\ &\quad \cdot p(\mathbf{z}_1|\mathbf{x}_1, \cancel{\mathbf{x}_2}, \dots, \cancel{\mathbf{x}_K})\, p(\mathbf{z}_2|\cancel{\mathbf{x}_1}, \mathbf{x}_2, \dots, \cancel{\mathbf{x}_K}, \cancel{\mathbf{z}_1}) \cdots \\ &\quad \cdot p(\mathbf{z}_K|\cancel{\mathbf{x}_1}, \dots, \mathbf{x}_K, \cancel{\mathbf{z}_1}, \dots, \cancel{\mathbf{z}_{K-1}}) \\ &= p(\mathbf{x}_1) \left[\prod_{k=2}^{K} p(\mathbf{x}_k|\mathbf{x}_{k-1}) \right] \prod_{k=1}^{K} p(\mathbf{z}_k|\mathbf{x}_k) \end{aligned}$$

Temporal Reasoning

Introduction: State Space Model (from Ch. 3)

- Two important models for sequential data that are described by this graph



1. **Discrete case:** if the latent variables are discrete, we obtain a **hidden Markov Model** (HMM). Observed variables can either be discrete or continuous in HMMs
2. **Continuous case:** If both the latent and the observed variables are continuous and Gaussian, we have the **linear dynamical system** (LDS)

Temporal Reasoning

State Space Model

- The state space representation is made up by three components:
  1. The **transition model** describes how the world/system evolves. It specifies the probability distribution $p(\mathbf{x}_k \mid \mathbf{x}_{0:k-1})$ over the latest state variable given the previous values. Thanks to the Markov assumption, this is

$$p(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$$

for first-order Markov chains. For second-order Markov chains we have

$$p(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{x}_{k-2})$$

where we have introduced the notation $\mathbf{x}_{a:b}$ to denote the sequence of states $\{\mathbf{x}_a, \mathbf{x}_{a+1}, \dots, \mathbf{x}_{b-1}, \mathbf{x}_b\}$

Temporal Reasoning

State Space Model

- The state space representation is made up by three components:

2. The **observation (or sensor) model** specifies the probability distribution over the observed variable given the previous state and observation sequence

$$p(\mathbf{z}_k \mid \mathbf{x}_{0:k}, \mathbf{z}_{0:k-1})$$

This is simplified to be

$$p(\mathbf{z}_k \mid \mathbf{x}_{0:k}, \mathbf{z}_{0:k-1}) = p(\mathbf{z}_k \mid \mathbf{x}_k)$$

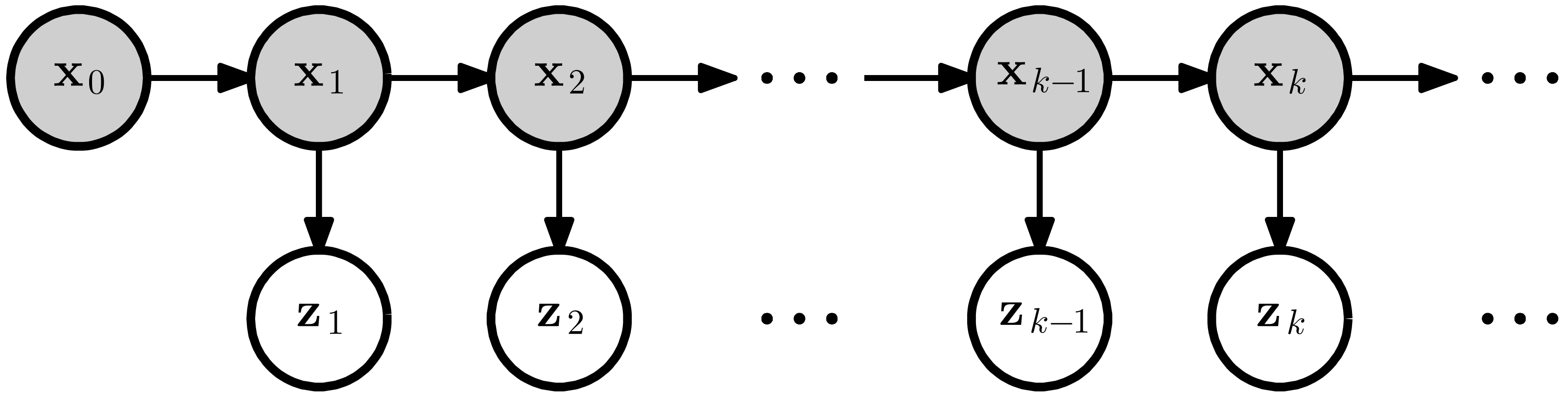
which is sometimes called the **sensor Markov assumption**

3. The **prior probability distribution** over the state at time 0, $p(\mathbf{x}_0)$, sometimes also called initial state model

Temporal Reasoning

State Space Model

- With that, we have specified the complete joint distribution over the domain (= over all our random variables)

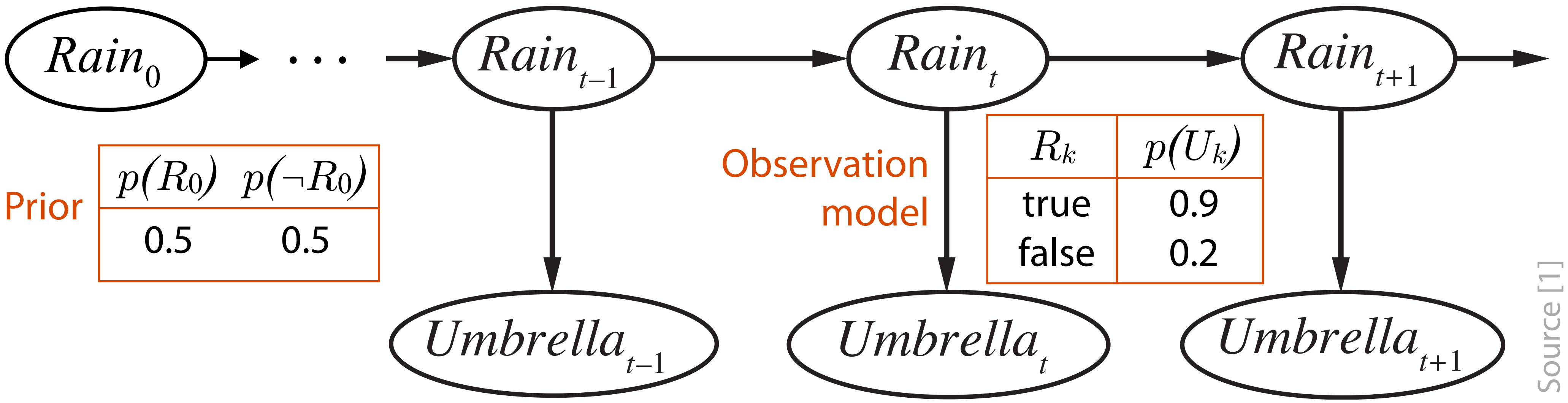


$$\boxed{p(\mathbf{x}_{0:K}, \mathbf{z}_{1:K}) = \underset{\text{Prior}}{p(\mathbf{x}_0)} \prod_{k=1}^{K} \underset{\text{Transition model}}{p(\mathbf{x}_k | \mathbf{x}_{k-1})} \, \underset{\text{Observation model}}{p(\mathbf{z}_k | \mathbf{x}_k)}}$$

Temporal Reasoning

State Space Model Example

- Suppose you are a security guard stationed at an underground facility and the only way to know whether it is raining today is by observing the director who comes in with or without an umbrella each morning
- Let $\mathbf{x}_k = Rain_k$ be the binary state variable and $\mathbf{z}_k = Umbrella_k$ the binary observation variable, both with values true or false. Then, the state space model is



Transition model

R_{k-1}	$p(R_k)$	$p(\neg R_k)$
true	0.7	0.3
false	0.3	0.7

Prior

$p(R_0)$	$p(\neg R_0)$
0.5	0.5

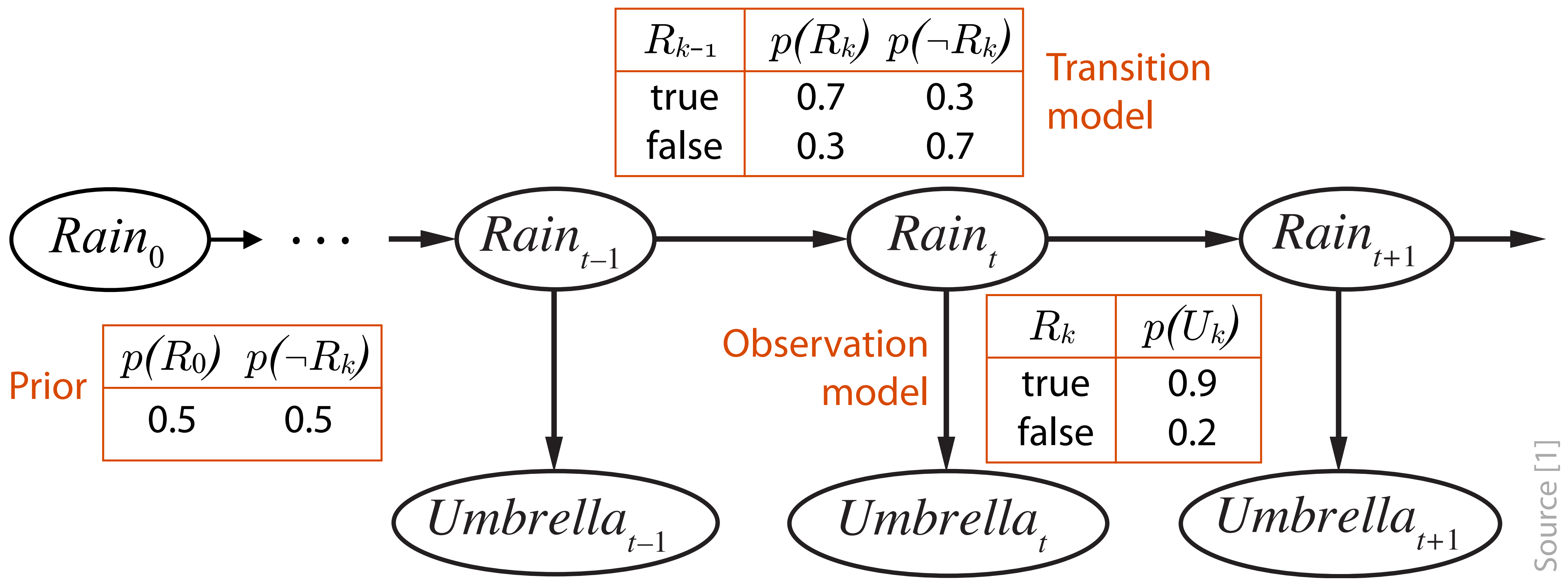
Observation model

R_k	$p(U_k)$
true	0.9
false	0.2

Source [1]

Temporal Reasoning

State Space Model Example



R_{k-1}	$p(R_k)$	$p(\neg R_k)$
true	0.7	0.3
false	0.3	0.7

$p(R_0)$	$p(\neg R_k)$
0.5	0.5

R_k	$p(U_k)$
true	0.9
false	0.2

$$\begin{tikzcd} Rain_0 \arrow[r] & \cdots \arrow[r] & Rain_{t-1} \arrow[r] \arrow[d] & Rain_t \arrow[r] \arrow[d] & Rain_{t+1} \arrow[r] \arrow[d] & {} \\ & & Umbrella_{t-1} & Umbrella_t & Umbrella_{t+1} & \end{tikzcd}$$

- Note the dependencies between states and sensors: arrows go from actual states to sensor values because the state of the world/system **causes** the sensors to take on particular values: the rain **causes** the umbrella to appear
- The **inference process** goes in the other direction: we seek to estimate the state given observations of the world

Temporal Reasoning

State Space Model

- The example uses a **first-order** Markov process which implies that the current state variable contains all the information needed to compute the probability of rain for the next time step
- Whether this assumption is **reasonable** depends on the context/domain. The assumption may be exactly true or may be approximate
- There are two ways to **improve the accuracy** of approximate models:
  - **Increasing the order** of the Markov process model. E.g. we could add $Rain_{k-2}$ as a parent of $Rain_k$
  - **Adding more state variables.** For example, we could add $Season$ or $Humidity$. However, adding more variables might improve the system's predictive power but increases the prediction requirements: we have to predict the new variables as well
- Looking for a self-sufficient set of variables that reflect the physics of the modeled process

Temporal Reasoning

Inference

- Having set up the representation of a generic temporal model consisting of $p(\mathbf{x}_0)$, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and $p(\mathbf{z}_k|\mathbf{x}_k)$, we can formulate the **four basic inference tasks:**

  - **Filtering:** computing the posterior distribution over the **most recent** state given all observations to date

    $$p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$$

    Keeping track of the system's current state in an online fashion

  - **Prediction:** computing the posterior distribution over the **future** state given all observations to date

    $$p(\mathbf{x}_{k+t} \mid \mathbf{z}_{1:k}) \quad t > 0$$

    Projecting the state into the future without observations (e.g. weather)

Temporal Reasoning

Inference

- **Smoothing:** computing the posterior distribution over a **past** state given all observations up to the present

  $$p(\mathbf{x}_t \mid \mathbf{z}_{1:k}) \quad 0 \leq t < k$$

  As opposed to filtering incorporates "the future". Leads to smoother state estimates than filtering

- **Most likely sequence:** given a sequence of observations, finding the most likely **state sequence** to have generated those observations

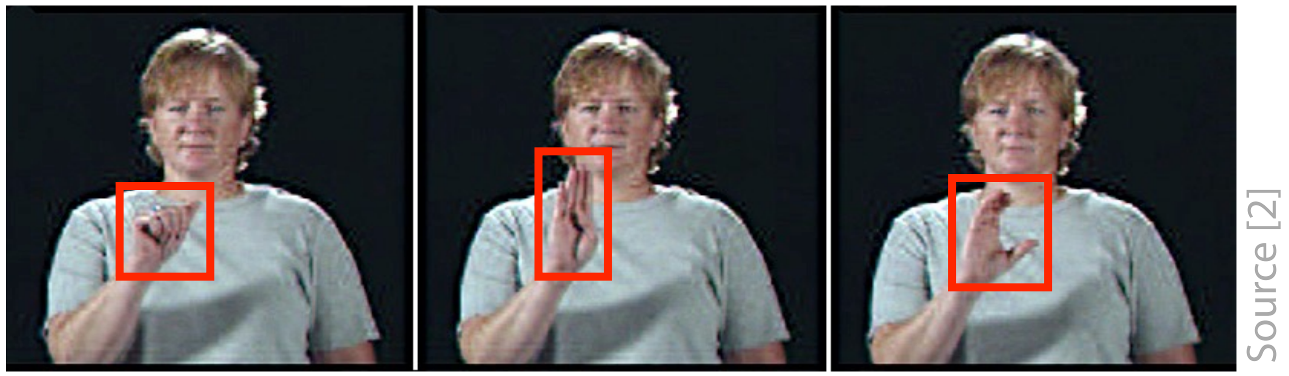
  $$\arg\max_{\mathbf{x}_{1:k}} p(\mathbf{x}_{1:k} \mid \mathbf{z}_{1:k})$$

  Highly relevant e.g. in speech recognition where the aim is to find the most likely word given a series of sounds

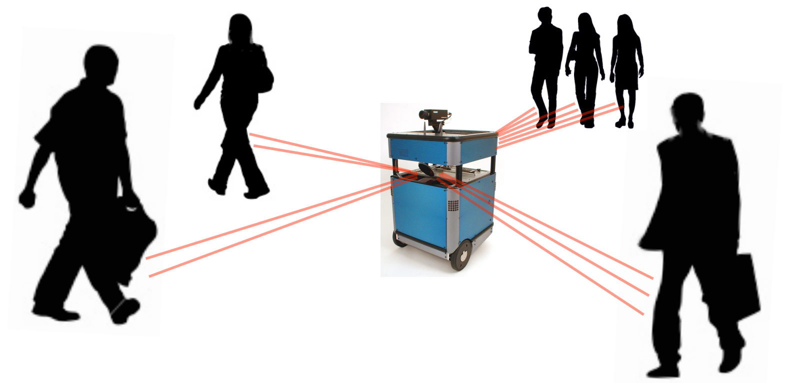
Temporal Reasoning

Examples

- Consider a motion tracking system able to follow human motion



Interpreting sign language from a sequence of images. Hands are detected and tracked



Keeping track of human actions from sequences of laser-based positions

- Suppose we have two goals:
  - Tracking and state estimation: associating single-frame detections (of the hand, the human) over time and accurately estimating their position from noisy observations
  - Recognizing different movement primitives which may serve later to infer high-level activities such as the sign that is presented or the action the human is engaged in
- The former is a continuous, the latter a discrete state estimation problem. We will first consider the **discrete case**

Hidden Markov Models

Hidden Markov Models (HMM)

- An HMM is a temporal probabilistic model in which the state of the process is described by a **single discrete** random variable
- The random variable $\mathbf{x}_k$ can take S possible states $s \in \{1,...,S\}$
- The HMM is widely used in speech recognition, natural language modeling, human activity recognition, on-line handwriting recognition, analysis of protein/DNA sequences, etc.
- Notice the **similarity to mixture models** where the latent variables are also discrete, describing which mixture component is responsible for generating the corresponding observation
- Except that here we have that the probability distribution over $\mathbf{x}_k$ depends on the state of the previous latent variable $\mathbf{x}_{k-1}$ through the transition model $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$

Hidden Markov Models

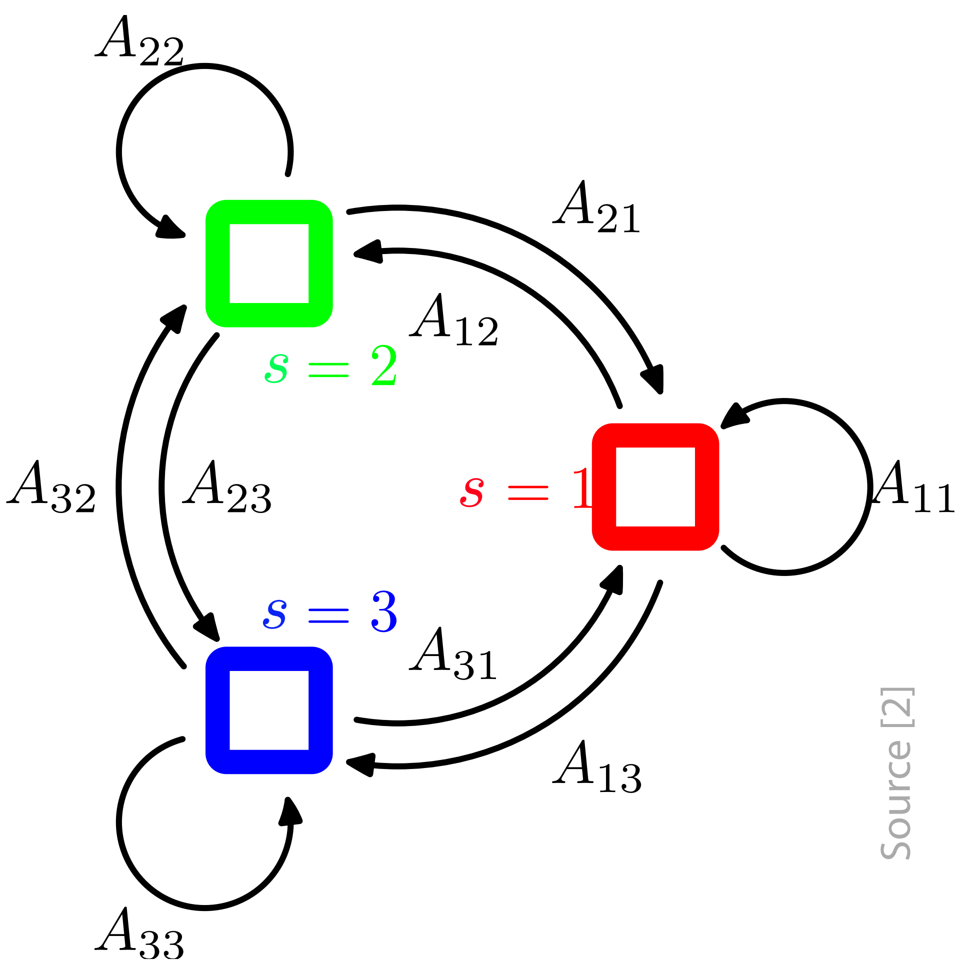
Transition Model

- The transition model is represented as a square matrix A with transition probabilities A_{ij} where $0 \leq A_{ij} \leq 1$ and $\sum_j A_{ij} = 1$, that is, outgoing probabilities sum up to 1
- A has $S(S–1)$ free parameters

$$A = \begin{array}{cc} & j \text{ (next)} \\ i \text{ (current)} & \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & \boxed{A_{32}} & A_{33} \end{pmatrix} \end{array}$$

$\sum_j A_{1j} = 1$

Probability of a transition from state 3 to state 2



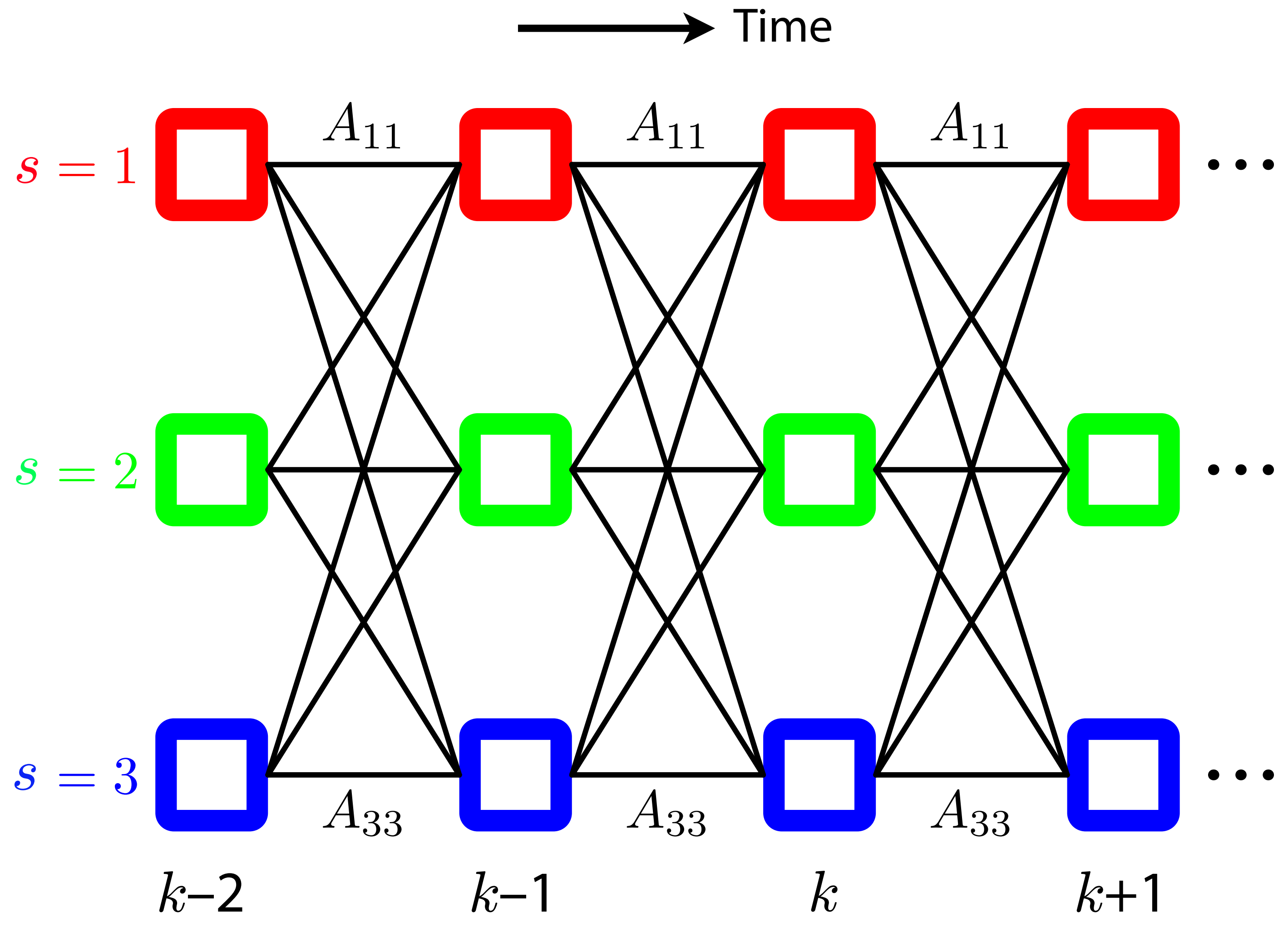
Source [2]

- The state transition diagram is **not** a graphical model, because the nodes are not separate variables but rather states of a single variable

Hidden Markov Models

Transition Model

- Unfolded over time, we obtain a lattice (or trellis) diagram

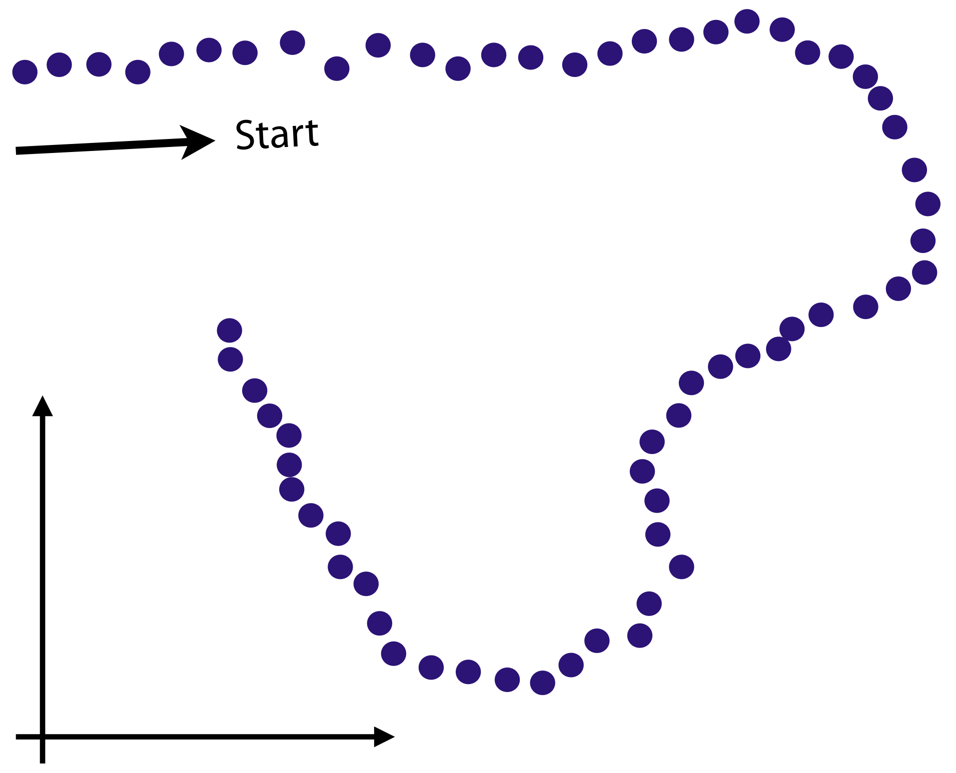


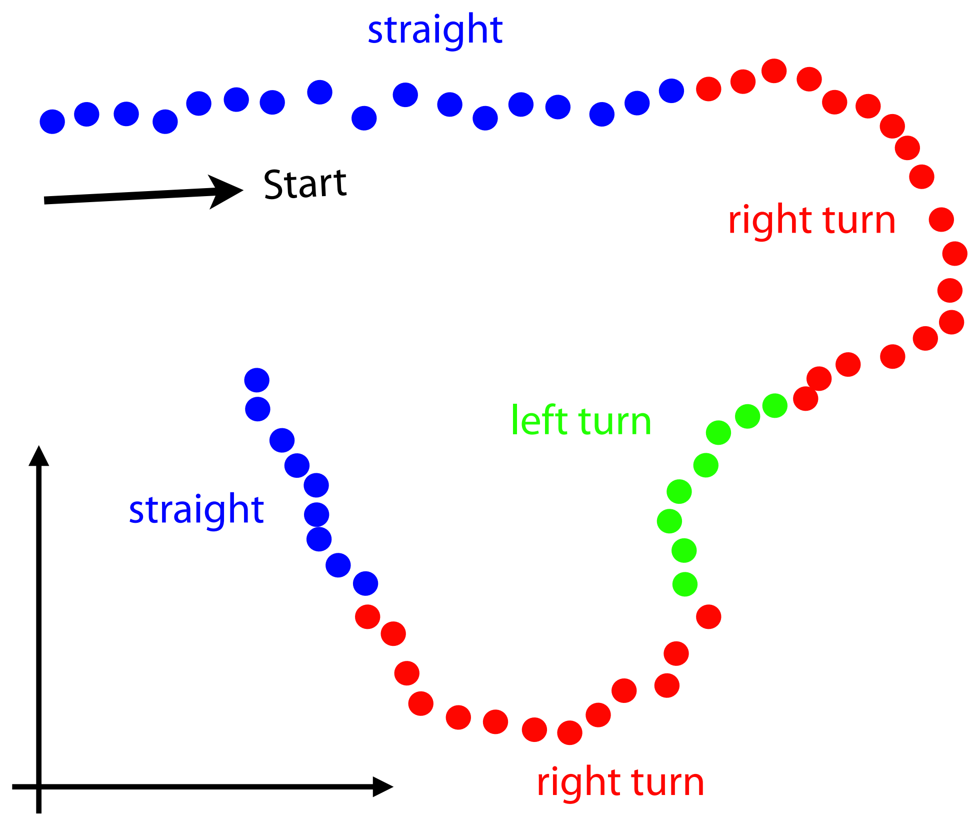
Source [2]

Hidden Markov Models

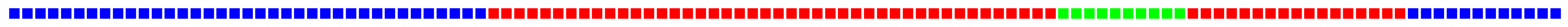
Transition Model

- States in a simple human action recognition example





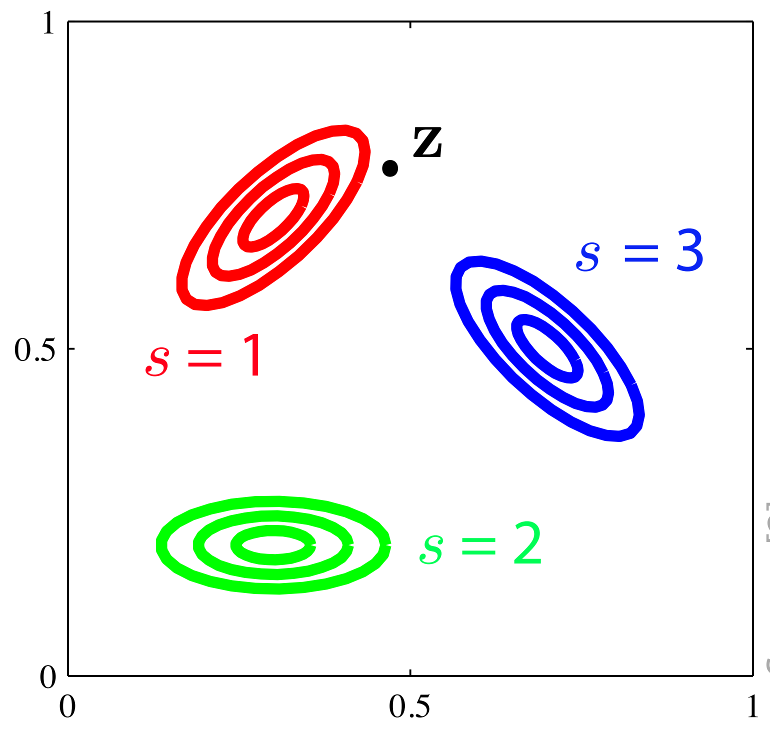
- Ground truth state sequence:



Hidden Markov Models

Observation Model

- Observed variables can either be discrete or continuous in HMMs
- They may be of different type and dimensionality than the states
- The observation model $p(\mathbf{z}_k|\mathbf{x}_k)$ "maps" the space of observations to the space of discrete HMM states. Concretely, for each $\mathbf{z}_k$ it computes a probability distribution over the states $s \in$ {1,...,S} specifying the **emission probability** that state s caused observation $\mathbf{z}_k$ to appear
- In the **continuous case**, a parametric distribution with parameters θ, such as a Gaussian, is typically chosen. We may also write $p(\mathbf{z}_k|\mathbf{x}_k, \theta)$
- We then need a conditional probability $p(\mathbf{z}_k|\mathbf{x}_k = s, \theta)$ for each state s



Source [2]

Hidden Markov Models

Observation Model and Prior

- In the **discrete case**, the observation model is represented as a matrix E of emission probabilities E_{ij} where $0 \leq E_{ij} \leq 1$ and $\sum_j E_{ij} = 1$

$$E = \begin{array}{c} \\ i \text{ (states)} \end{array} \begin{array}{c} j \text{ (observation symbol)} \\ \begin{pmatrix} E_{11} & E_{12} & E_{13} & E_{14} & E_{15} \\ E_{21} & E_{22} & E_{23} & E_{24} & E_{25} \\ E_{31} & E_{32} & E_{33} & \boxed{E_{34}} & E_{35} \end{pmatrix} \end{array} \quad \sum_j E_{1j} = 1$$

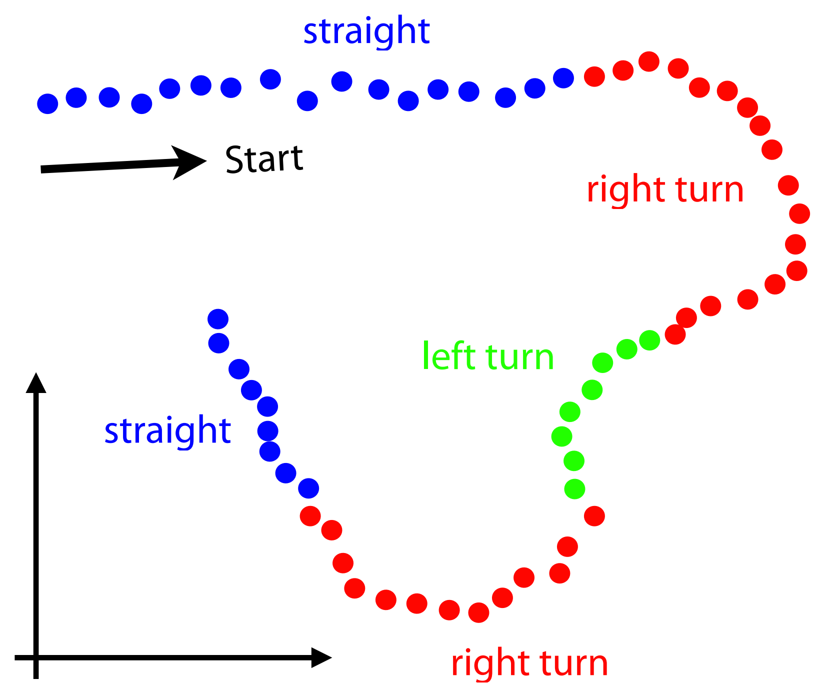
Emission probability of symbol 4 from state 3

- Every row of E describes a distribution $p(\mathbf{z}_t | \mathbf{x}_t = s)$
- Let observations be from a finite set of O symbols. A special case is when E is squared, that is, $S = O$
- Finally, we also have to specify the **prior distribution** $p(\mathbf{x}_0)$ over states s, using, for example, a categorial distribution

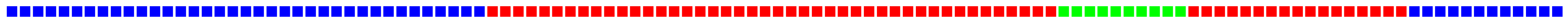
Hidden Markov Models

Observation Model

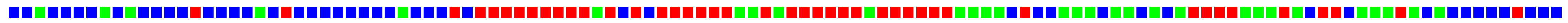
- States and observations in our running example: $S = O = 3$
- Suppose we have a noisy "sensor" for movement primitives, for example, a poorly trained 3-way classifier



- Ground truth sequence of state



- Sequence of observations



Hidden Markov Models

Inference

- Having set up the representation of an HMM with parameters $p(\mathbf{x}_0)$, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{z}_k|\mathbf{x}_k)$ we recall the **four basic inference tasks:**
  - **Filtering:** computing the posterior distribution $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ over the **most recent** state given all observations to date
  - **Prediction:** computing the posterior distribution over the **future** state given all observations to date: $p(\mathbf{x}_{k+t} \mid \mathbf{z}_{1:k}) \quad t > 0$
  - **Smoothing:** computing the posterior distribution over a **past** state given all observations up to the present, i.e. $p(\mathbf{x}_t \mid \mathbf{z}_{1:k}) \quad 0 \leq t < k$
  - **Most likely sequence:** given a sequence of observations, finding the most likely **state sequence** to have generated those observations. Formally, $\arg\max_{\mathbf{x}_{1:k}} p(\mathbf{x}_{1:k} \mid \mathbf{z}_{1:k})$

Hidden Markov Models

Inference: Smoothing

- Let us start with the **more general smoothing** which contains **filtering** and **prediction** as a special case
- We want to compute $p(\mathbf{x}_t \mid \mathbf{z}_{1:k}) \quad 0 \leq t < k$

$$\begin{aligned}
p(\mathbf{x}_t \mid \mathbf{z}_{1:k}) &= \eta\, p(\mathbf{x}_t, \mathbf{z}_{1:k}) && \text{conditional probability} \\
&= \eta\, p(\mathbf{x}_t, \mathbf{z}_{1:t}, \mathbf{z}_{t+1:k}) && \text{“dividing up the evidence”} \\
&= \eta\, p(\mathbf{z}_{t+1:k} \mid \mathbf{x}_t, \mathbf{z}_{1:t})\, p(\mathbf{x}_t, \mathbf{z}_{1:t}) && \text{chain rule} \\
&= \eta\, \underbrace{p(\mathbf{z}_{t+1:k} \mid \mathbf{x}_t)}_{\text{backward}}\, \underbrace{p(\mathbf{x}_t, \mathbf{z}_{1:t})}_{\text{forward}} && \text{sensor Markov assumption}
\end{aligned}$$

- We will see next that the first term can be computed in a **backward recursion** and the second term in a **forward recursion** through the chain

Hidden Markov Models

Inference: Smoothing

- Computing $p(\mathbf{x}_t, \mathbf{z}_{1:t})$ is called **forward step**

$$p(\mathbf{x}_t, \mathbf{z}_{1:t})$$

$$= \sum_{\mathbf{x}_{t-1}=1}^{S} p(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{z}_{1:t})$$

marginalization

$$= \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{z}_t)$$

“dividing up the evidence”

$$= \sum_{\mathbf{x}_{t-1}} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}) \; p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}) \; p(\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$$

chain rule

$$= \sum_{\mathbf{x}_{t-1}} p(\mathbf{z}_t | \mathbf{x}_t) \; p(\mathbf{x}_t | \mathbf{x}_{t-1}) \; p(\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$$

conditional independence

Hidden Markov Models

Inference: Smoothing

- Computing $p(\mathbf{x}_t, \mathbf{z}_{1:t})$ is called **forward step**

$$\boxed{p(\mathbf{x}_t, \mathbf{z}_{1:t})}$$

$$= \sum_{\mathbf{x}_{t-1}=1}^{S} p(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{z}_{1:t})$$ marginalization

$$= \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{z}_t)$$ "dividing up the evidence"

$$= \sum_{\mathbf{x}_{t-1}} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})\ p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})\ p(\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$$ chain rule

$$= \sum_{\mathbf{x}_{t-1}} p(\mathbf{z}_t | \mathbf{x}_t)\ p(\mathbf{x}_t | \mathbf{x}_{t-1})\ \boxed{p(\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})}$$ conditional independence

We have found a recursion!

Hidden Markov Models

Inference: Smoothing

- All terms in

$$p(\mathbf{x}_t, \mathbf{z}_{1:t}) = \sum_{\mathbf{x}_{t-1}=1}^{S} p(\mathbf{z}_t \mid \mathbf{x}_t)\, p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})$$

  are known HMM parameters (emission and transition probabilities). Let the recursive term be α_{t-1}, then we can write

$$\boxed{\alpha_t = p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, \alpha_{t-1}}$$

- How to compute α_1? We find

$$\alpha_1 = p(\mathbf{x}_1, \mathbf{z}_{1:1}) = p(\mathbf{x}_1, \mathbf{z}_1) = p(\mathbf{x}_1)\, p(\mathbf{z}_1 | \mathbf{x}_1)$$

  Again, this depends only on known parameters (priors/emission probs.)

Hidden Markov Models

Inference: Filtering

- Let us rewrite the α-term using conditional probability

$$\alpha_t = p(\mathbf{x}_t, \mathbf{z}_{1:t}) = \eta \, p(\mathbf{x}_t | \mathbf{z}_{1:t})$$

- We recognize the **wanted posterior probability for filtering** (up to normalization)
- The **recursiveness** to compute α is an important property of **filtering algorithms**: the update only depends on the previous estimate and the "new" observation,

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) \; = \; f(\mathbf{z}_t, p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}))$$

rather than going back over the entire observation history every time

Hidden Markov Models

Inference: Filtering

- Considering the rewritten α-term:

$$p(\mathbf{x}_t \mid \mathbf{z}_{1:t}) \;=\; \eta\, p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1})$$

  leads us immediately – after dropping the conditioning on the observation history – to an important result: the **recursive Bayes filter**

$$\boxed{p(\mathbf{x}_t \mid \mathbf{z}_t) \;=\; \eta\, p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1} | \mathbf{z}_{t-1})}$$

- The Bayes filter is a **general recursive state estimation scheme** (in the continuous case, the sum becomes an integral)
- Various applications in control and robotics: localization, SLAM, tracking, …

Hidden Markov Models

Inference: Filtering

- Filtering has a **prediction–update scheme**
- For this to become evident, we transform the sum on the right hand side

$$\sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}) =$$

$$= \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1})\, p(\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}) \quad \text{conditional independence}$$

$$= \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}) \quad \text{chain rule}$$

$$= p(\mathbf{x}_t, \mathbf{z}_{1:t-1}) \quad \text{marginalization}$$

$$= \eta\, p(\mathbf{x}_t \mid \mathbf{z}_{1:t-1}) \quad \text{conditional probability}$$

- This represent a **one-step prediction** of the next state

Hidden Markov Models

Inference: Filtering

- Filtering has a **prediction–update scheme**

$$\boxed{p(\mathbf{x}_t \mid \mathbf{z}_t) = \eta\, p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1} | \mathbf{z}_{t-1})}$$

$$= \eta'\, \underbrace{p(\mathbf{z}_t \mid \mathbf{x}_t)}_{\text{update}}\, \underbrace{p(\mathbf{x}_t \mid \mathbf{z}_{t-1})}_{\text{prediction}}$$

- **Kalman filter** and **particle filter** are two **continuous** instances of the recursive Bayes filter. They have explicit prediction and update steps (later in this course)

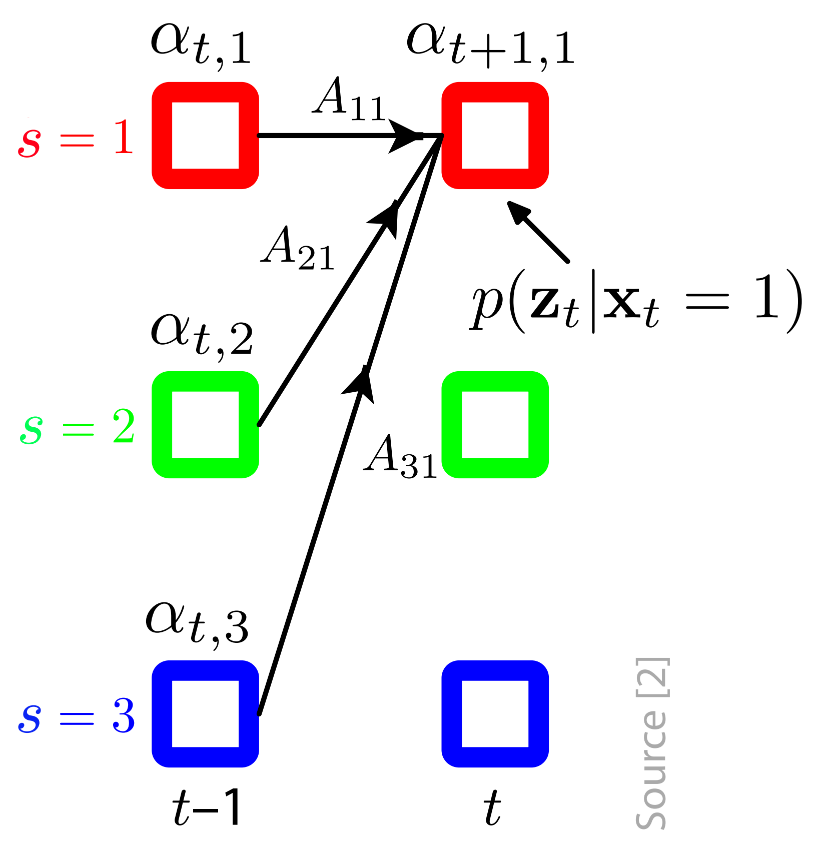
Hidden Markov Models

Inference: Filtering

- Back to HMMs, let us look at the forward step
- In filtering, we compute $\alpha_t = p(\mathbf{x}_t, \mathbf{z}_{1:t})$ as

$$\boxed{\alpha_t = p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, \alpha_{t-1}}$$

  in each step by taking the previous $\alpha_{t,s}$ values for every state s, summing them up with weights given by the transition matrix A, and then multiplying by the observation model for state s, $p(\mathbf{z}_t|\mathbf{x}_t = s)$



Source [2]

- This is called the **forward algorithm**. We start at the first node of the chain, work along the chain and evaluate α_t for every latent node
- $O(S^2)$ complexity per step, $O(S^2K)$ for a chain of length K

Hidden Markov Models

Inference: Smoothing

- Let us get back to smoothing and the **backward term**
- We recall

$$\begin{aligned}
p(\mathbf{x}_t \mid \mathbf{z}_{1:k}) &= \eta\, p(\mathbf{x}_t, \mathbf{z}_{1:k}) && \text{conditional probability} \\
&= \eta\, p(\mathbf{x}_t, \mathbf{z}_{1:t}, \mathbf{z}_{t+1:k}) && \text{“dividing up the evidence”} \\
&= \eta\, p(\mathbf{z}_{t+1:k} \mid \mathbf{x}_t, \mathbf{z}_{1:t})\, p(\mathbf{x}_t, \mathbf{z}_{1:t}) && \text{chain rule} \\
&= \eta\, \underbrace{p(\mathbf{z}_{t+1:k} \mid \mathbf{x}_t)}_{\text{backward}}\, \underbrace{p(\mathbf{x}_t, \mathbf{z}_{1:t})}_{\text{forward}} && \text{sensor Markov assumption}
\end{aligned}$$

- As we will see next, the backward term has a very similar derivation than the forward term

Hidden Markov Models

Inference: Smoothing

- Computing $p(\mathbf{z}_{k+1:k} \mid \mathbf{x}_t)$ is called **backward step**

$$p(\mathbf{z}_{t+1:k} \mid \mathbf{x}_t) = \sum_{\mathbf{x}_{t+1}=1}^{S} p(\mathbf{z}_{t+1:k}, \mathbf{x}_{t+1} \mid \mathbf{x}_t)$$

marginalization

$$= \sum_{\mathbf{x}_{t+1}} p(\mathbf{z}_{t+1}, \mathbf{z}_{t+2:k}, \mathbf{x}_{t+1} \mid \mathbf{x}_t)$$

"dividing up the evidence"

$$= \sum_{\mathbf{x}_{t+1}} p(\mathbf{z}_{t+2:k} \mid \mathbf{z}_{t+1}, \mathbf{x}_{t+1}, \mathbf{x}_t)\, p(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1}, \mathbf{x}_t)\, p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)$$

chain rule

$$= \sum_{\mathbf{x}_{t+1}} p(\mathbf{z}_{t+2:k} \mid \mathbf{x}_{t+1})\, p(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1})\, p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)$$

conditional independence

Inference: Smoothing

- Computing $p(\mathbf{z}_{k+1:k} \mid \mathbf{x}_t)$ is called **backward step**

$$p(\mathbf{z}_{t+1:k} \mid \mathbf{x}_t) = \sum_{\mathbf{x}_{t+1}=1}^{S} p(\mathbf{z}_{t+1:k}, \mathbf{x}_{t+1} \mid \mathbf{x}_t)$$

marginalization

$$= \sum_{\mathbf{x}_{t+1}} p(\mathbf{z}_{t+1}, \mathbf{z}_{t+2:k}, \mathbf{x}_{t+1} \mid \mathbf{x}_t)$$

"dividing up the evidence"

$$= \sum_{\mathbf{x}_{t+1}} p(\mathbf{z}_{t+2:k} \mid \mathbf{z}_{t+1}, \mathbf{x}_{t+1}, \mathbf{x}_t)\, p(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1}, \mathbf{x}_t)\, p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)$$

chain rule

$$= \sum_{\mathbf{x}_{t+1}} p(\mathbf{z}_{t+2:k} \mid \mathbf{x}_{t+1})\, p(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1})\, p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)$$

conditional independence

Again a recursion, this time in a backward sense!

Hidden Markov Models

Inference: Smoothing

- Again, all terms in

$$p(\mathbf{z}_{t+1:k} \mid \mathbf{x}_t) = \sum_{\mathbf{x}_{t+1}=1}^{S} p(\mathbf{z}_{t+2:k} \mid \mathbf{x}_{t+1})\, p(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1})\, p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)$$

are known HMM parameters (emission and transition probabilities). Let the recursive term be β_t , then we write

$$\boxed{\beta_t = \sum_{\mathbf{x}_{t+1}} \beta_{t+1}\, p(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1})\, p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)}$$

- This is valid for $t = 1, ..., k-1$. But what about β_k ? Let us consider the second last step $t = k-1$

Hidden Markov Models

Inference: Smoothing

- For the second last step, $t = k - 1$, we redo the derivation

$$\begin{aligned}
\beta_{k-1} &= p(\mathbf{z}_{k:k} \mid \mathbf{x}_{k-1}) \\
&= p(\mathbf{z}_k \mid \mathbf{x}_{k-1}) \\
&= \sum_{\mathbf{x}_k} p(\mathbf{z}_k, \mathbf{x}_k \mid \mathbf{x}_{k-1}) && \text{marginalization} \\
&= \sum_{\mathbf{x}_k} p(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{x}_{k-1})\, p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) && \text{chain rule} \\
&= \sum_{\mathbf{x}_k} p(\mathbf{z}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) && \text{conditional independence}
\end{aligned}$$

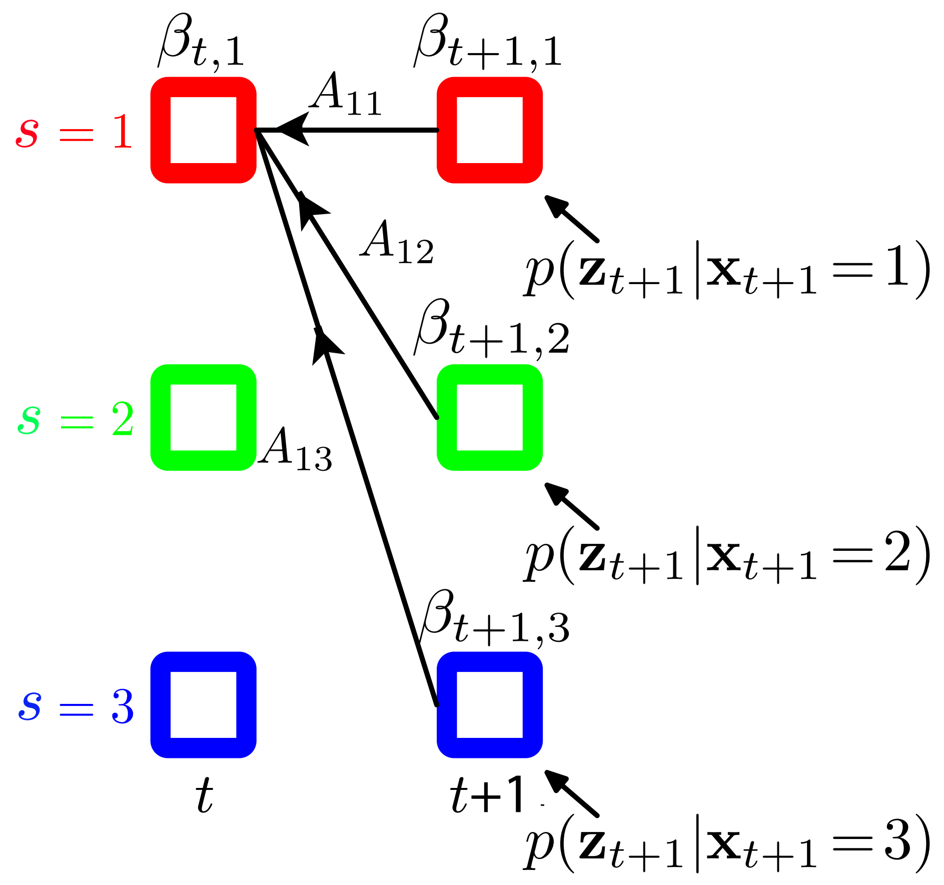
- It follows that $\beta_k = 1$ for this to be true

Hidden Markov Models

Inference: Smoothing

- Let us look at the backward step
- We compute $\beta_t = p(\mathbf{z}_{t+1:k} \mid \mathbf{x}_t)$ as

$$\boxed{\beta_t = \sum_{\mathbf{x}_{t+1}} \beta_{t+1}\, p(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1})\, p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)}$$


$\beta_{t,1}$
$\beta_{t+1,1}$
$s=1$
A_{11}
A_{12}
$p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1}=1)$
$\beta_{t+1,2}$
$s=2$
A_{13}
$p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1}=2)$
$\beta_{t+1,3}$
$s=3$
t
t+1
$p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1}=3)$

in each step by taking the future $\beta_{t+1,s}$ values for every state s, multiplying the corresponding elements of A and observation model for state s, $p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1} = s)$, and summing up

- This is called the **backward algorithm**. We start at the last node of the chain, work backwards, and evaluate β_t for every latent node
- $O(S^2)$ complexity per step, $O(S^2K)$ for a chain of length K

Hidden Markov Models

Inference: Smoothing

- Let us wrap up: in our derivation we have found the forward algorithm and the backward algorithm to compute the two individual terms of

$$\begin{aligned} p(\mathbf{x}_t \mid \mathbf{z}_{1:k}) &= \eta\, p(\mathbf{z}_{t+1:k} \mid \mathbf{x}_t)\, p(\mathbf{x}_t, \mathbf{z}_{1:t}) \\ &= \eta\, \underset{\text{backward}}{\beta_t}\, \underset{\text{forward}}{\alpha_t} \end{aligned}$$

- The α term is computed by **filtering**, running forward in time from 1 to t, the β term is computed by running backward in time from k down to t+1
- The **smoothing result** is finally obtained by multiplication and normalization of the terms. The corresponding algorithm is the **forward-backward algorithm**

Hidden Markov Models

Inference: Smoothing

- The forward-backward algorithm for smoothing

```
Algorithm: Forward-Backward Algorithm
    In: $\mathbf{z}$, vector of observations
        $prior$, prior distribution on initial state $p(\mathbf{x}_0)$
    Out: $\mathbf{x}$, vector of smoothed probability distributions
    Local: $\mathbf{f}$, vector of forward probabilities
           $\mathbf{b}$, vector of backward probabilities, initially all 1
    $\mathbf{f}[0] \leftarrow prior$                                  } filtering   } smoothing
    for $t = 1 \dots k$ do
        $\mathbf{f}[t] \leftarrow \alpha_t(\mathbf{f}[t-1], \mathbf{z}[t])$
    end
    for $t = k \dots 1$ do
        $\mathbf{x}[t] \leftarrow \text{normalize}(\mathbf{f}[t] \times \mathbf{b})$
        $\mathbf{b} \leftarrow \beta_t(\mathbf{b}, \mathbf{z}[t])$
    end
    return $\mathbf{x}$
```

Hidden Markov Models

Inference: Smoothing

- The forward-backward algorithm (and variants thereof) is the backbone for many applications that deal with sequences of noisy observations
- However, it is an off-line algorithm and does not work in an **online setting** where smoothed estimates are required as new observations are continuously added to the end of the sequence
- **Fixed-lag smoothing** is able to do this. It involves computing the smoothed estimated

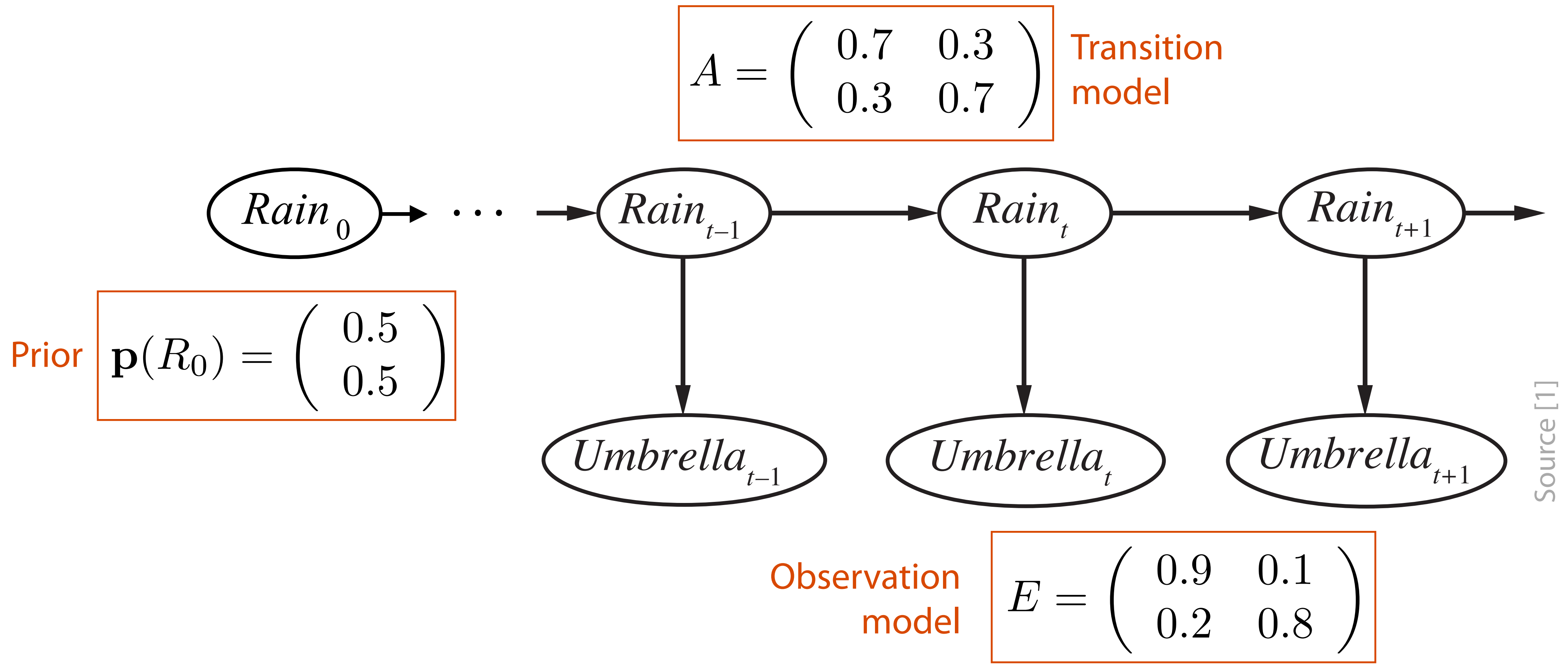
  $$p(\mathbf{x}_{t-d} \mid \mathbf{z}_{1:t})$$

  for fixed d. That is, smoothing is done for the time index d steps behind the current time t
- Efficient fixed-lag smoothing algorithms with **constant time updates** exist (see literature)

Hidden Markov Models

Inference: Filtering Example

- Let us illustrate filtering and smoothing in the umbrella example


Transition model
$A = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$
$Rain_0$
$Rain_{t-1}$
$Rain_t$
$Rain_{t+1}$
Prior
$\mathbf{p}(R_0) = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$
$Umbrella_{t-1}$
$Umbrella_t$
$Umbrella_{t+1}$
Source [1]
Observation model
$E = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}$

- **Day 0**: we have no observation, only the security guard's prior beliefs. Let us assume uniformity, that is, $\mathbf{p}(R_0) = (0.5, 0.5)^T$

Hidden Markov Models

$$\underline{p(\mathbf{x}_t \mid \mathbf{z}_t)} = \eta\, p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, \underline{p(\mathbf{x}_{t-1} | \mathbf{z}_{t-1})}$$

Inference: Filtering Example

- **Day 1:** U_1 = true, the umbrella appears
- The prediction from k = 0 to k = 1 is

$$\begin{aligned}\mathbf{p}(R_1) &= \textstyle\sum_{r_0=\{t,f\}} \mathbf{p}(R_1|R_0 = r_0)\, p(R_0 = r_0) \\ &= \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} \cdot 0.5 + \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix} \cdot 0.5 = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}\end{aligned}$$

- Then the update step simply multiplies by the probability of the observation for k = 1 and normalizes

$$\begin{aligned}\mathbf{p}(R_1|U_1 = u_1) &= \eta\, \mathbf{p}(U_1 = u_1|R_1)\, p(R_1) \\ &= \eta \begin{pmatrix} 0.9 \\ 0.2 \end{pmatrix} \cdot \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} = \eta \begin{pmatrix} 0.45 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.818 \\ 0.182 \end{pmatrix}\end{aligned}$$

Hidden Markov Models

$$p(\mathbf{x}_t \mid \mathbf{z}_t) = \eta\, p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1}|\mathbf{z}_{t-1})$$

Inference: Filtering Example

- **Day 2:** U_2 = true, umbrella appears again
- The prediction from k = 1 to k = 2 is

$$\begin{aligned}\mathbf{p}(R_2|U_1 = u_1) &= \textstyle\sum_{r_1=\{t,f\}} \mathbf{p}(R_2|R_1 = r_1)\, p(R_1 = r_1|U_1 = u_1) \\ &= \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} \cdot 0.818 + \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix} \cdot 0.182 = \begin{pmatrix} 0.627 \\ 0.373 \end{pmatrix}\end{aligned}$$

- The update step for k = 2 is

$$\begin{aligned}\mathbf{p}(R_2|U_2 = u_2) &= \eta\, \mathbf{p}(U_2 = u_2|R_2)\, p(R_2|U_1 = u_1) \\ &= \eta \begin{pmatrix} 0.9 \\ 0.2 \end{pmatrix} \cdot \begin{pmatrix} 0.627 \\ 0.373 \end{pmatrix} = \eta \begin{pmatrix} 0.565 \\ 0.075 \end{pmatrix} = \begin{pmatrix} 0.883 \\ 0.117 \end{pmatrix}\end{aligned}$$

Hidden Markov Models

$$p(\mathbf{x}_t \mid \mathbf{z}_t) = \eta \, p(\mathbf{z}_t \mid \mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \, p(\mathbf{x}_{t-1} | \mathbf{z}_{t-1})$$

Inference: Filtering Example

- **Day 2:** U_2 = true, umbrella appears again
- The prediction from $k = 1$ to $k = 2$ is

$$\begin{aligned}\mathbf{p}(R_2|U_1 = u_1) &= \textstyle\sum_{r_1=\{t,f\}} \mathbf{p}(R_2|R_1 = r_1) \, p(R_1 = r_1|U_1 = u_1) \\ &= \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} \cdot 0.818 + \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix} \cdot 0.182 = \begin{pmatrix} 0.627 \\ 0.373 \end{pmatrix}\end{aligned}$$

- The update step for $k = 2$ is

Increases because rain persists

$$\begin{aligned}\mathbf{p}(R_2|U_2 = u_2) &= \eta \, \mathbf{p}(U_2 = u_2|R_2) \, p(R_2|U_1 = u_1) \\ &= \eta \begin{pmatrix} 0.9 \\ 0.2 \end{pmatrix} \cdot \begin{pmatrix} 0.627 \\ 0.373 \end{pmatrix} = \eta \begin{pmatrix} 0.565 \\ 0.075 \end{pmatrix} = \begin{pmatrix} \boxed{0.883} \\ 0.117 \end{pmatrix}\end{aligned}$$

Hidden Markov Models

$$\beta_t = \sum_{\mathbf{x}_{t+1}} \beta_{t+1}\, p(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1})\, p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)$$

Inference: Smoothing Example

- Let us compute the **smoothed estimate** for **day 1**, given the umbrella observations for day 1 and day 2: $U_1 = \text{true}, U_2 = \text{true}$

$$\mathbf{p}(R_1|U_1 = u_1, U_2 = u_2) = \eta\, \mathbf{p}(R_1|U_1 = u_1)\, \mathbf{p}(U_2 = u_2|R_1)$$

- The first term is known from the filtering pass, the second term can be computed by applying the backward recursion

$$\mathbf{p}(U_2 = u_2|R_1) = \textstyle\sum_{r_2=\{t,f\}} \beta_2\, p(U_2 = u_2|R_2 = r_2)\, \mathbf{p}(R_2 = r_2|R_1)$$
$$= 1 \cdot 0.9 \cdot \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} + 1 \cdot 0.2 \cdot \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix} = \begin{pmatrix} 0.69 \\ 0.41 \end{pmatrix}$$

- Plugging this back in yields the smoothed estimate

$$\mathbf{p}(R_1|U_1 = u_1, U_2 = u_2) = \eta \begin{pmatrix} 0.818 \\ 0.182 \end{pmatrix} \times \begin{pmatrix} 0.69 \\ 0.41 \end{pmatrix} = \begin{pmatrix} 0.883 \\ 0.117 \end{pmatrix}$$

Hidden Markov Models

$$\underline{\beta_t} = \sum_{\mathbf{x}_{t+1}} \underline{\beta_{t+1}\, p(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1})\, p(\mathbf{x}_{t+1} \mid \mathbf{x}_t)}$$

Inference: Smoothing Example

- Let us compute the **smoothed estimate** for **day 1**, given the umbrella observations for day 1 and day 2: $U_1 = \text{true}, U_2 = \text{true}$

$$\mathbf{p}(R_1 | U_1 = u_1, U_2 = u_2) = \eta\, \mathbf{p}(R_1 | U_1 = u_1)\, \mathbf{p}(U_2 = u_2 | R_1)$$

- The first term is known from the filtering pass, the second term can be computed by applying the backward recursion

$$\begin{aligned}\mathbf{p}(U_2 = u_2 | R_1) &= \textstyle\sum_{r_2 = \{t,f\}} \beta_2\, p(U_2 = u_2 | R_2 = r_2)\, \mathbf{p}(R_2 = r_2 | R_1) \\ &= 1 \cdot 0.9 \cdot \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} + 1 \cdot 0.2 \cdot \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix} = \begin{pmatrix} 0.69 \\ 0.41 \end{pmatrix}\end{aligned}$$

- Plugging this back in yields the smoothed estimate

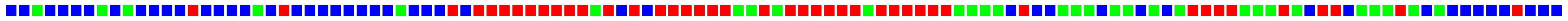
$$\mathbf{p}(R_1 | U_1 = u_1, U_2 = u_2) = \eta \begin{pmatrix} 0.818 \\ 0.182 \end{pmatrix} \times \begin{pmatrix} 0.69 \\ 0.41 \end{pmatrix} = \begin{pmatrix} \boxed{0.883} \\ 0.117 \end{pmatrix}$$

Higher than filtering for $k = 1$

Hidden Markov Models

Inference: Filtering Example

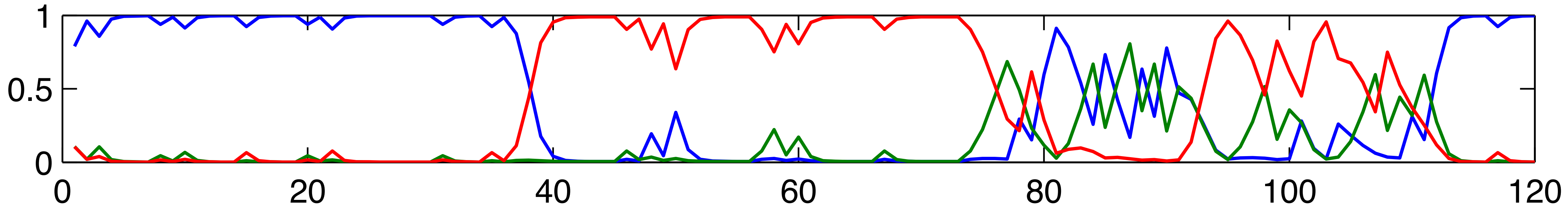
- In our simple human action recognition example, the **sequence of observations** is



- Let us assume the following HMM parameters

$$A = \begin{pmatrix} 0.98 & 0.01 & 0.01 \\ 0.04 & 0.95 & 0.01 \\ 0.025 & 0.025 & 0.95 \end{pmatrix} \qquad E = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{pmatrix} \qquad \mathbf{p}(\mathbf{x}_0) = \begin{pmatrix} 0.33 \\ 0.33 \\ 0.33 \end{pmatrix}$$

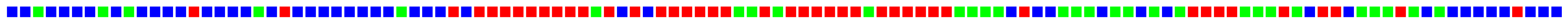
- The **filtered** probabilities are



Hidden Markov Models

Inference: Smoothing Example

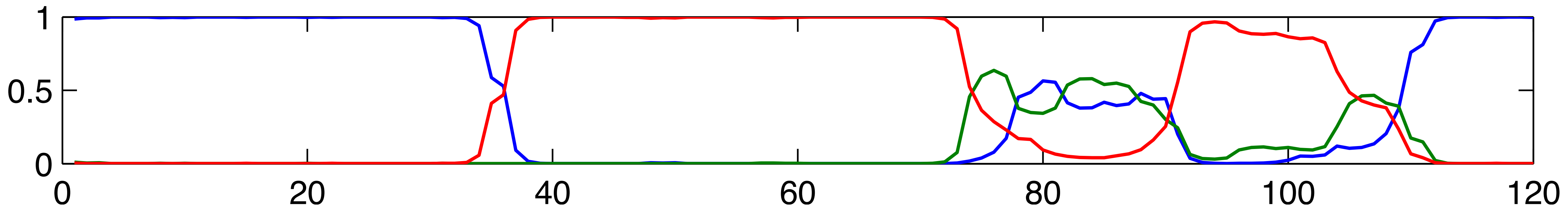
- In our simple human action recognition example, the **sequence of observations** is



- Let us assume the following HMM parameters

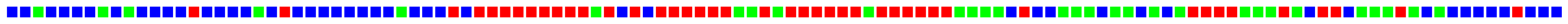
$$A = \begin{pmatrix} 0.98 & 0.01 & 0.01 \\ 0.04 & 0.95 & 0.01 \\ 0.025 & 0.025 & 0.95 \end{pmatrix} \qquad E = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{pmatrix} \qquad \mathbf{p}(\mathbf{x}_0) = \begin{pmatrix} 0.33 \\ 0.33 \\ 0.33 \end{pmatrix}$$

- The **smoothed** probabilities are



Inference: Smoothing Example

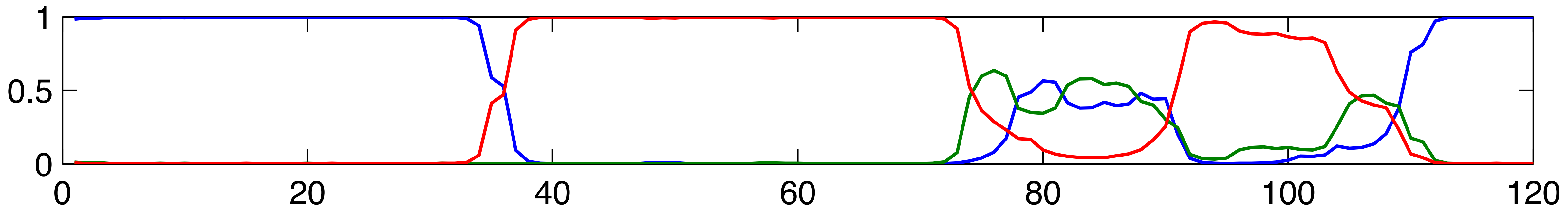
- In our simple human action recognition example, the **sequence of observations** is



- Let us assume the following HMM parameters

$$A = \begin{pmatrix} 0.98 & 0.01 & 0.01 \\ 0.04 & 0.95 & 0.01 \\ 0.025 & 0.025 & 0.95 \end{pmatrix} \qquad E = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{pmatrix} \qquad \mathbf{p}(\mathbf{x}_0) = \begin{pmatrix} 0.33 \\ 0.33 \\ 0.33 \end{pmatrix}$$

- The **smoothed** probabilities are



Smoothing provides much better state estimates!

Hidden Markov Models

Inference: Prediction

- The task of prediction can be seen as **filtering without new evidence**
- The filtering process already incorporates a **one-step prediction**. A general prediction of the state at time $t+i+1$ from a prediction for $t+i$ (with evidence only up to time t) is

$$p(\mathbf{x}_{t+i+1} \mid \mathbf{z}_{1:t}) = \sum_{\mathbf{x}_{t+i}} p(\mathbf{x}_{t+i+1} \mid \mathbf{x}_{t+i}) \, p(\mathbf{x}_{t+i} \mid \mathbf{z}_{1:t})$$

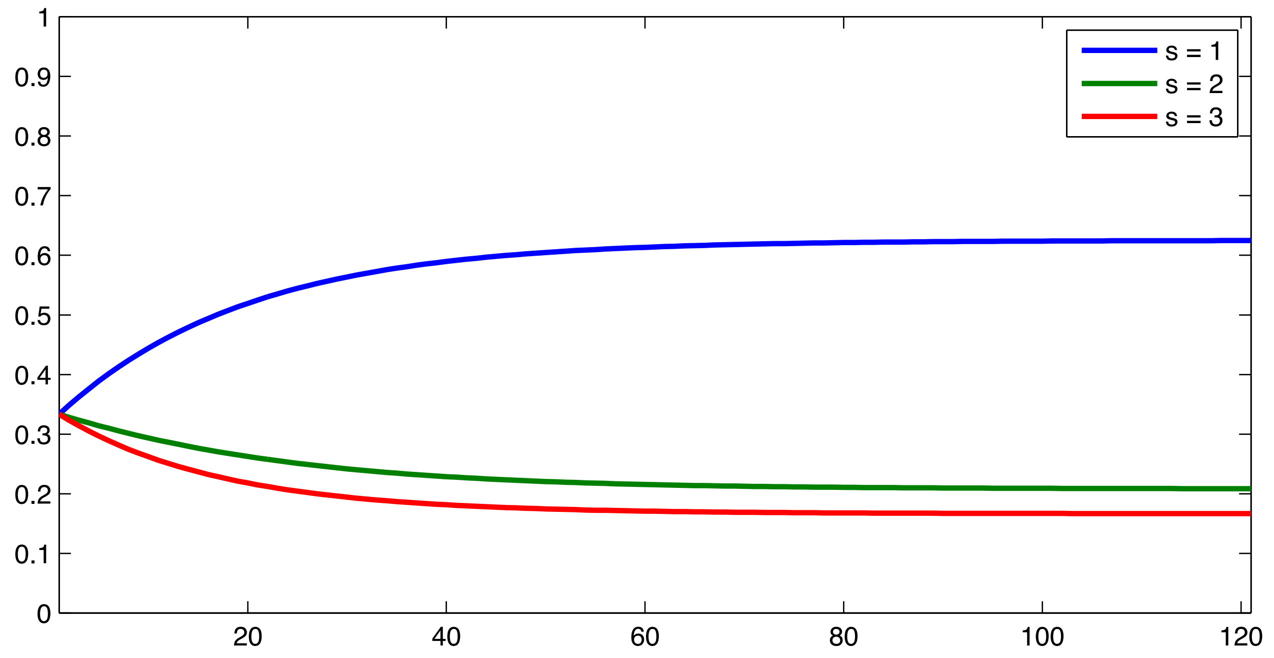
- Of course, this computation only involves the transition model, not the observation model
- Predicting further and further into the future makes the predicted distribution converge towards a **stationary distribution**, which is only determined by the transition matrix regardless the starting state

Hidden Markov Models

Inference: Prediction

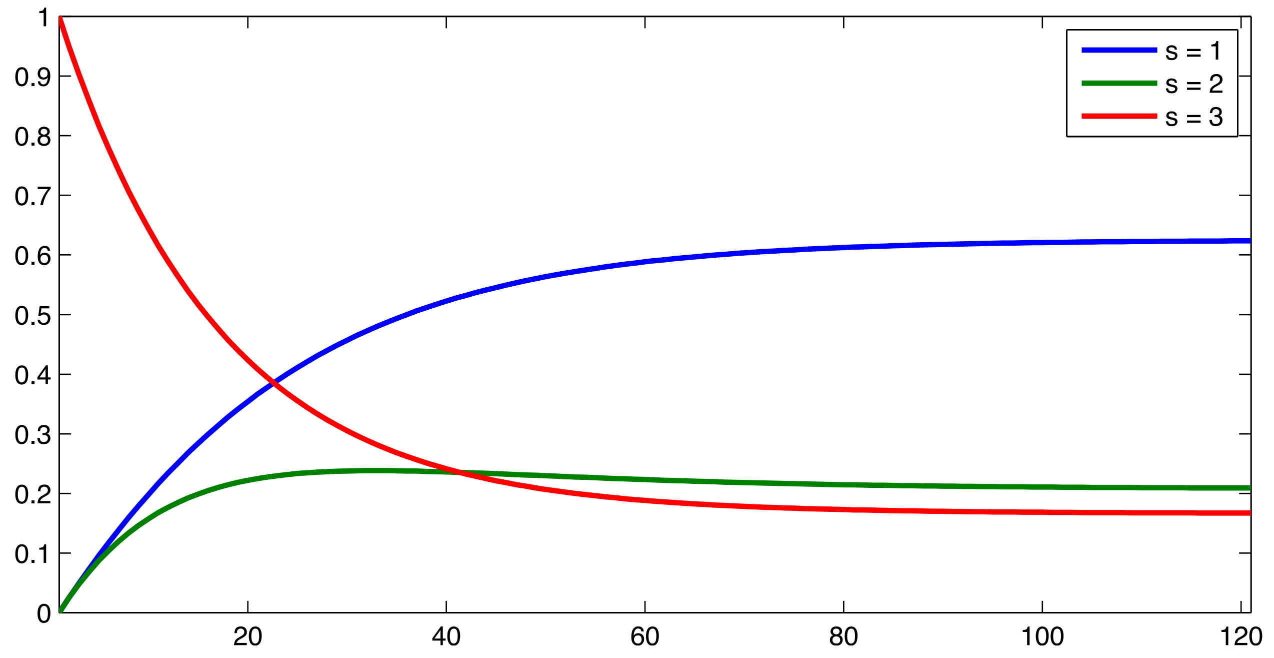
- Original transition model

$$A = \begin{pmatrix} 0.98 & 0.01 & 0.01 \\ 0.04 & 0.95 & 0.01 \\ 0.025 & 0.025 & 0.95 \end{pmatrix} \begin{matrix} 🟦 \\ 🟩 \\ 🟥 \end{matrix}$$



- Different priors

$$\mathbf{p}(\mathbf{x}_0) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \begin{matrix} 🟦 \\ 🟩 \\ 🟥 \end{matrix}$$

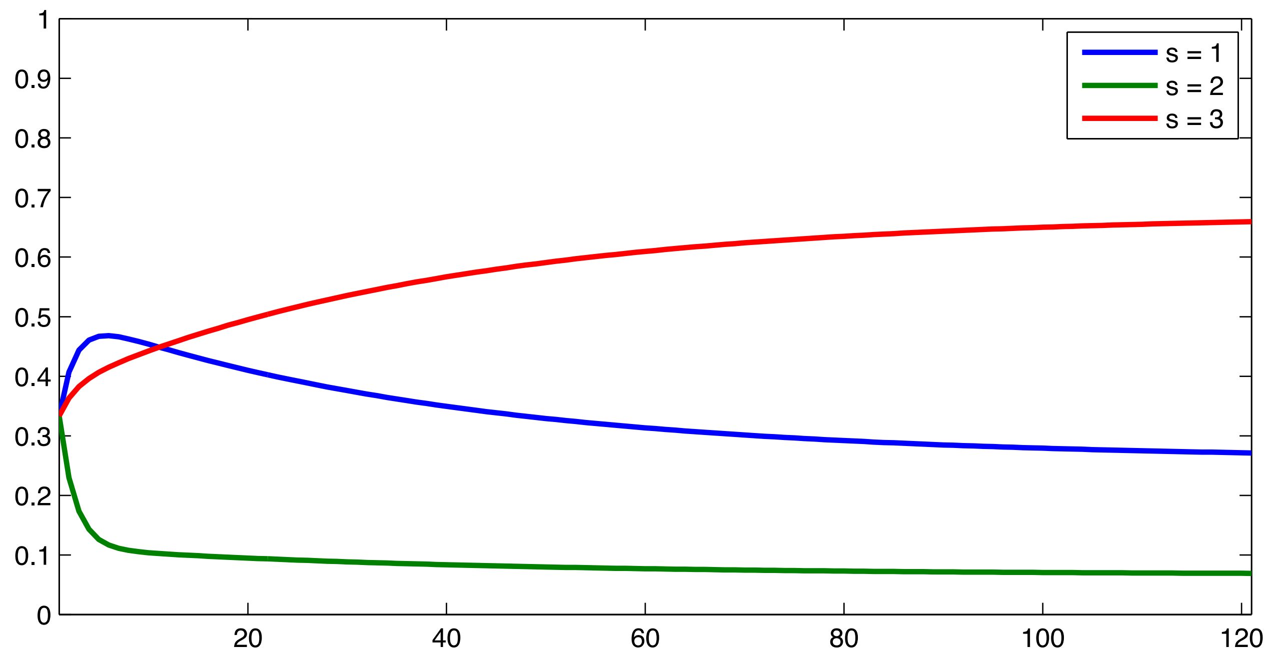


Hidden Markov Models

Inference: Prediction

- Uncertain second state

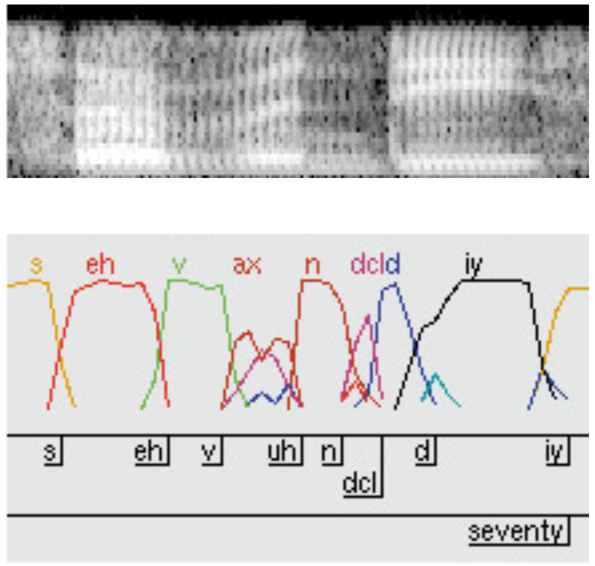
$$A = \begin{pmatrix} 0.92 & 0.08 & 0.0 \\ 0.3 & 0.6 & 0.1 \\ 0.009 & 0.001 & 0.99 \end{pmatrix}$$

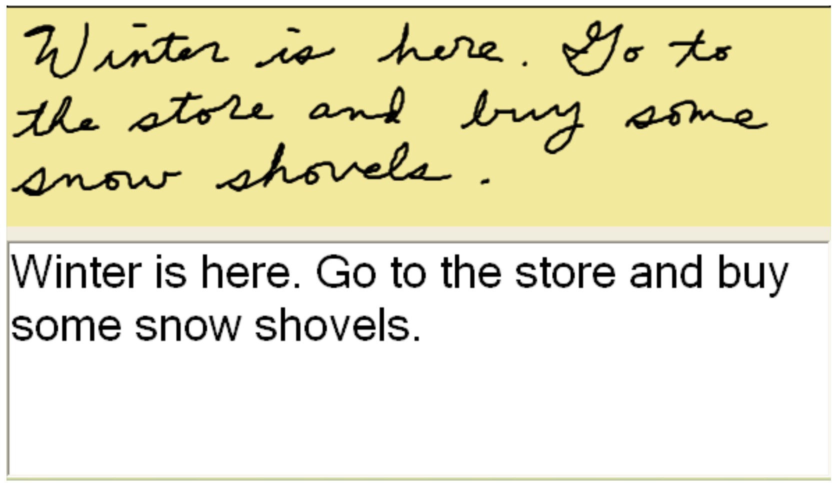


- The more uncertain (less peaked) a row distribution $p(\mathbf{x}_t \mid \mathbf{x}_{t-1} = s)$ in A is, the less likely it is to converge to state s in the **stationary distribution**
- The **mixing time** is the time it takes for the Markov process to converge
- The more uncertainty there is in the transition model (e.g. many nearly uniform row distributions), the shorter will be the mixing time and the more the future is obscured (stationary distribution nearly uniform)

Inference: Most Likely Sequence

- Given a sequence of observations, we want to find the **most likely state sequence** to have generated those observations
- Applications of this inference task include (among many others) **speech recognition** or **handwriting recognition** where we seek to find a word from a noisy sequence of recognized phonemes or letters
    - Phonemes or letters are typically obtained by probabilistic classification based on features that characterize portions of the original signal (e.g. small time frames)




Winter is here. Go to the store and buy some snow shovels.

Winter is here. Go to the store and buy some snow shovels.

Hidden Markov Models

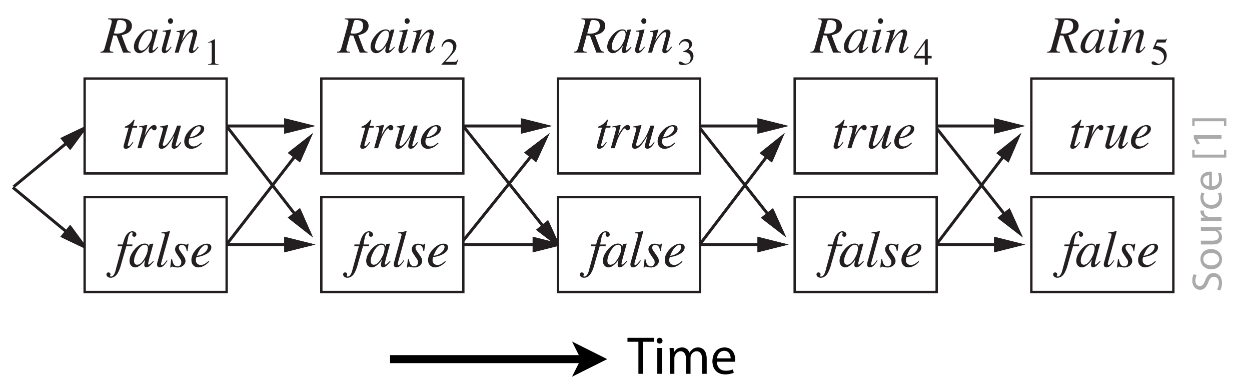
Inference: Most Likely Sequence

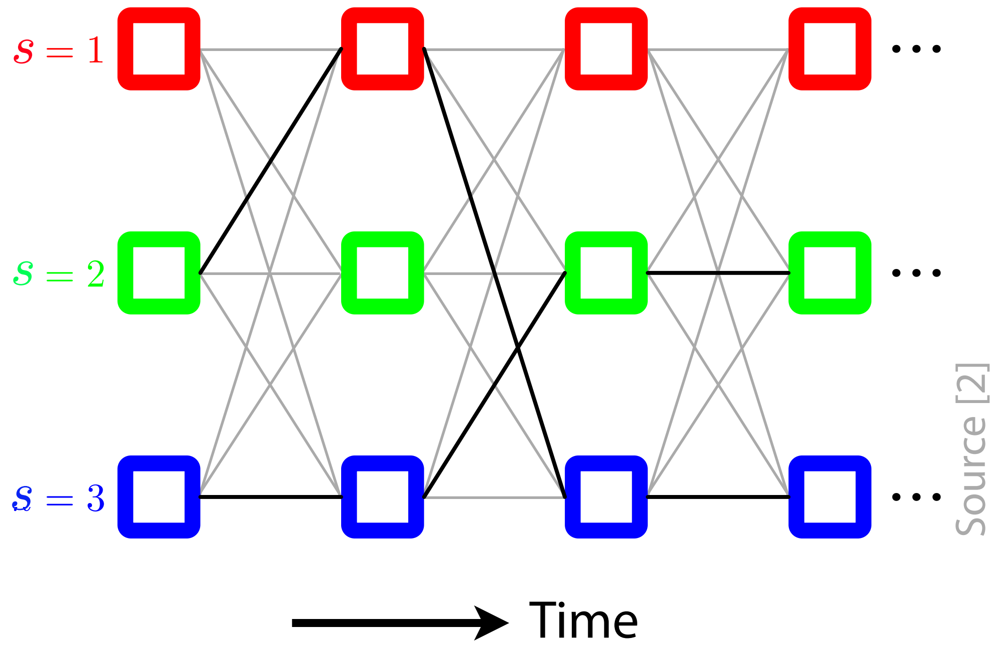
- For example, suppose the security guard observes the umbrella sequence $\{true, true, false, true, true\}$. What is the weather sequence most likely to explain this?
  - Does the absence of the umbrella on day 3 mean that it was not raining, or did the director forget to bring it?
  - If it didn't rain on day 3, perhaps – as weather tends to persist – it did not rain on day 4 either but the director brought it just in case. And so on...
- In all, there are 2^5 possible weather sequences. Is there a **smart way** to find the most likely, better than enumerating all of them?
- What about taking the smoothing result and choose the state with the highest posterior probability?
  - Nope! Those probabilities are distributions over **single** time steps. To find the most likely **sequence** we must consider **joint probabilities** over **all** time steps

Hidden Markov Models

Inference: Most Likely Sequence

- Let us view each sequence as a **path** through the trellis diagram

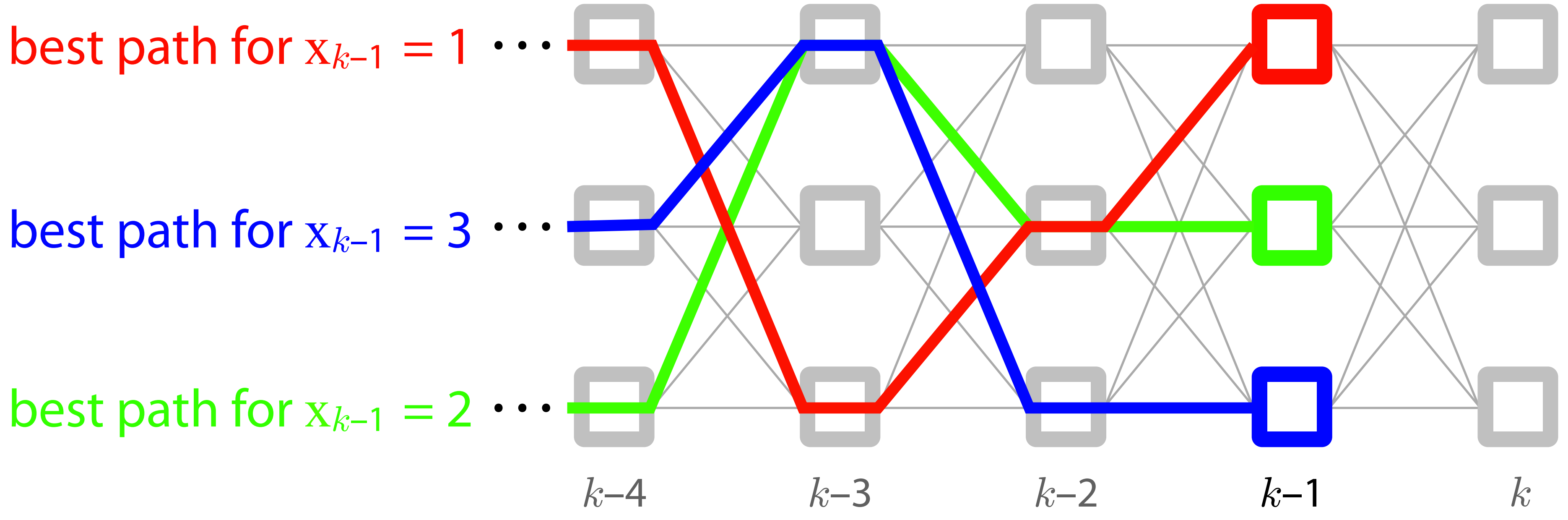




- Recall, we want to find the state sequence $\mathbf{x}^*$ that **maximizes the probability along its path**, i.e. $\mathbf{x}^* = \arg\max_{\mathbf{x}_{1:k}} p(\mathbf{x}_{1:k} \mid \mathbf{z}_{1:k})$

Inference: Most Likely Sequence

- Let us consider the **second last step** of a sequence of length k and see if we get an idea for an algorithm by induction
- Assume we already have the maximizing paths for time k–1

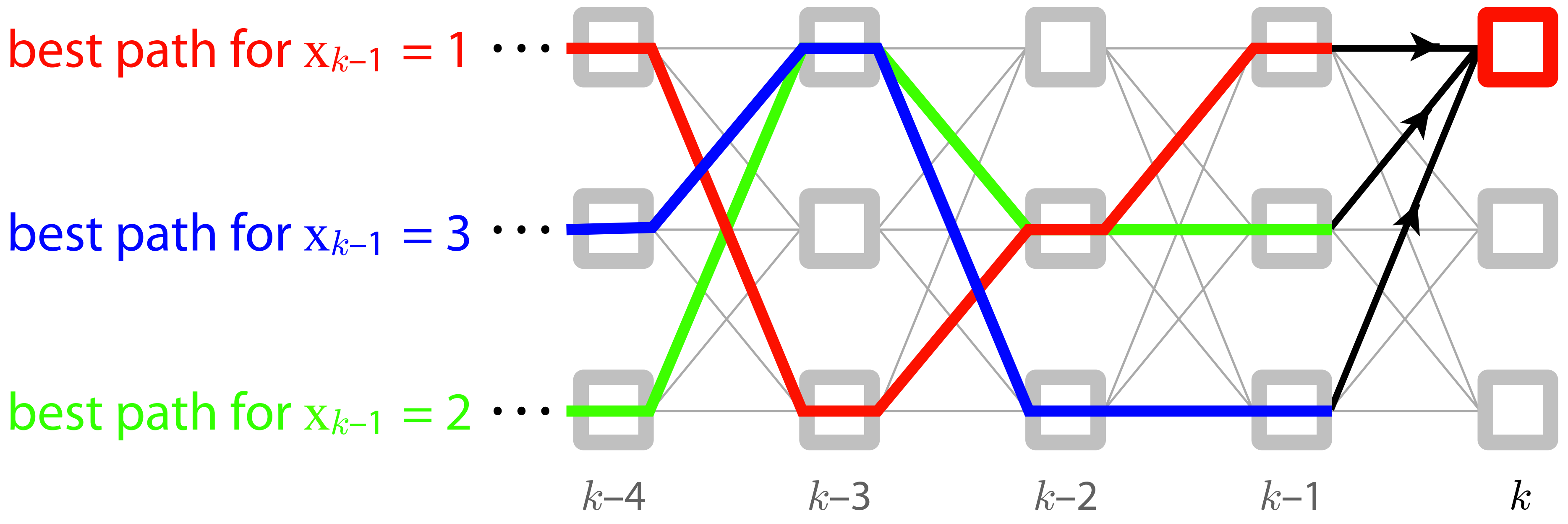

best path for $x_{k-1} = 1$ ···
best path for $x_{k-1} = 3$ ···
best path for $x_{k-1} = 2$ ···
k–4
k–3
k–2
k–1
k

- Can we **reuse** the results computed so far when extending the paths to time k?

Hidden Markov Models

Inference: Most Likely Sequence

- This would be the case if we were able to simply **append** the most likely path from time k–1 to time k to the computed ones, limiting the focus of our maximization onto a **single transition**


best path for $x_{k-1} = 1$ · · ·
best path for $x_{k-1} = 3$ · · ·
best path for $x_{k-1} = 2$ · · ·
k–4
k–3
k–2
k–1
k

- In other words: if there were a **recursive relationship** between most likely paths to each state in $\mathbf{x}_k$ and most likely paths to each state in $\mathbf{x}_{k-1}$

Hidden Markov Models

Inference: Most Likely Sequence

- Let us first consider the **maximum**

$$\max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k} \mid \mathbf{z}_{1:k})$$

  instead of the $\arg\max_{\mathbf{x}_{1:k}} p(\mathbf{x}_{1:k} \mid \mathbf{z}_{1:k})$, and see if we can later **recover** the state sequence from the maximum

- And consider the following useful relationship: given two nonnegative functions $f(a) \geq 0,\; g(a,b) \geq 0 \;\; \forall a, b$

$$\max_{a,b} f(a)\, g(a,b) = \max_{a} \left(f(a) \; \max_{b} g(a,b) \right)$$

- This can be verified by first ignoring the maximization over a making $f(a)$ a constant. Thus, we can "**pull**" the maximization over b into the product

Hidden Markov Models

Inference: Most Likely Sequence

- Developing the maximization term

$$\max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k} \mid \mathbf{z}_{1:k})$$

$$= \eta \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k}, \mathbf{z}_{1:k})$$ conditional probability

$$= \eta \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1}, \mathbf{x}_k, \mathbf{z}_k)$$

$$= \eta \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})\, p(\mathbf{x}_k | \mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})\, p(\mathbf{z}_k | \mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1}, \mathbf{x}_k)$$ chain rule

$$= \eta \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})\, p(\mathbf{x}_k | \mathbf{x}_{k-1})\, p(\mathbf{z}_k | \mathbf{x}_k)$$ conditional independence

$$= \eta\, p(\mathbf{z}_k | \mathbf{x}_k) \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_k | \mathbf{x}_{k-1})\, p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})$$

$$= \eta\, p(\mathbf{z}_k | \mathbf{x}_k) \max_{\mathbf{x}_{k-1}} \left(p(\mathbf{x}_k | \mathbf{x}_{k-1}) \max_{\mathbf{x}_{1:k-2}} p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1}) \right)$$ pulling in the max

Inference: Most Likely Sequence

- Developing the maximization term

$$\max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k} \mid \mathbf{z}_{1:k})$$

$$= \eta \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k}, \mathbf{z}_{1:k})$$ conditional probability

$$= \eta \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1}, \mathbf{x}_k, \mathbf{z}_k)$$

$$= \eta \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})\, p(\mathbf{x}_k | \mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})\, p(\mathbf{z}_k | \mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1}, \mathbf{x}_k)$$ chain rule

$$= \eta \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})\, p(\mathbf{x}_k | \mathbf{x}_{k-1})\, p(\mathbf{z}_k | \mathbf{x}_k)$$ conditional independence

$$= \eta\, p(\mathbf{z}_k | \mathbf{x}_k) \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_k | \mathbf{x}_{k-1})\, p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1})$$

$$= \eta\, p(\mathbf{z}_k | \mathbf{x}_k) \max_{\mathbf{x}_{k-1}} \left(p(\mathbf{x}_k | \mathbf{x}_{k-1}) \max_{\mathbf{x}_{1:k-2}} p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1}) \right)$$ pulling in the max

A recursion!

Hidden Markov Models

Inference: Most Likely Sequence

- Summarizing

$$\max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k}, \mathbf{z}_{1:k}) = p(\mathbf{z}_k \mid \mathbf{x}_k) \max_{\mathbf{x}_{k-1}} \Big(p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \max_{\mathbf{x}_{1:k-2}} p(\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1}) \Big)$$

- Let $\mu_k = \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k}, \mathbf{z}_{1:k})$, we have

$$\boxed{\mu_k = p(\mathbf{z}_k \mid \mathbf{x}_k) \max_{\mathbf{x}_{k-1}} \left(p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \, \mu_{k-1} \right)}$$

- This result is very similar to the filtering equation

$$\alpha_k = p(\mathbf{z}_k \mid \mathbf{x}_k) \sum_{\mathbf{x}_{k-1}} p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \, \alpha_{k-1}$$

where the **summation** over $\mathbf{x}$ is replaced by the **maximization** over $\mathbf{x}$

Hidden Markov Models

Inference: Most Likely Sequence

- Thus, the algorithm for computing the most likely sequence is **similar to filtering**: it runs forward along the sequence and computes μ at each time step
- At the end, it will have the probability for the most likely sequence reaching each of the final states
- We also require the **initial** state

$$\mu_1 = \max_{\mathbf{x}_{1:0}} p(\mathbf{x}_{1:1}, \mathbf{z}_{1:1}) = p(\mathbf{x}_1, \mathbf{z}_1) = p(\mathbf{x}_1)\, p(\mathbf{z}_1 \mid \mathbf{x}_1)$$

  which only depends on known HMM parameters

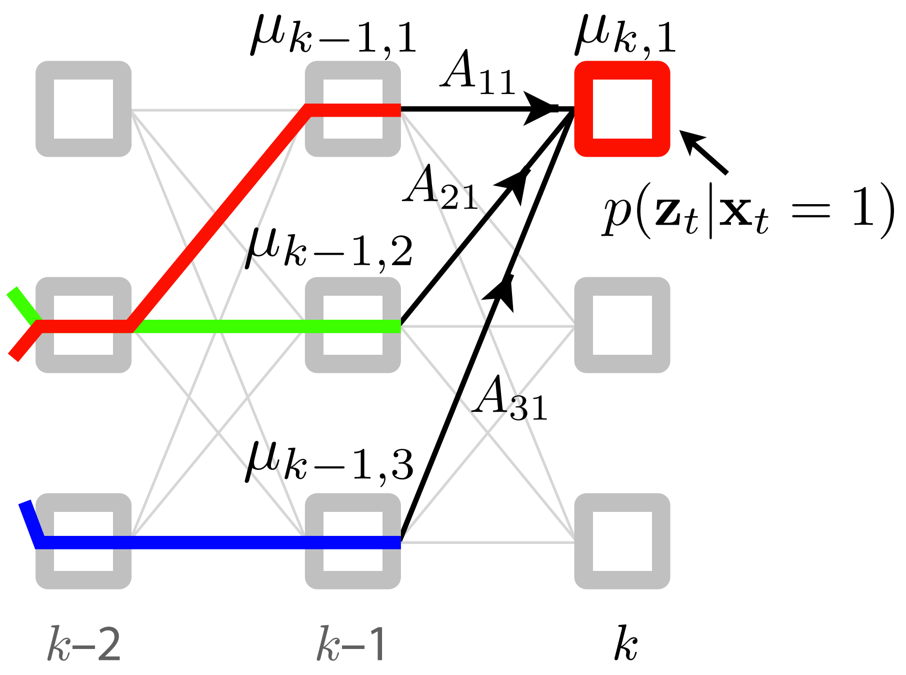
- This procedure computes the probability of the most likely sequence. The sought **state sequence**, $\mathbf{x}^*$, is obtained by storing pointers that, for each state, record the best state that leads to it. Finally, $\mathbf{x}^*$ is obtained through **backtracking**

Hidden Markov Models

Inference: Most Likely Sequence

- The algorithm computes $\mu_k = \max_{\mathbf{x}_{1:k-1}} p(\mathbf{x}_{1:k}, \mathbf{z}_{1:k})$ as

$$\boxed{\mu_k = p(\mathbf{z}_k \mid \mathbf{x}_k) \max_{\mathbf{x}_{k-1}} \left(p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, \mu_{k-1} \right)}$$

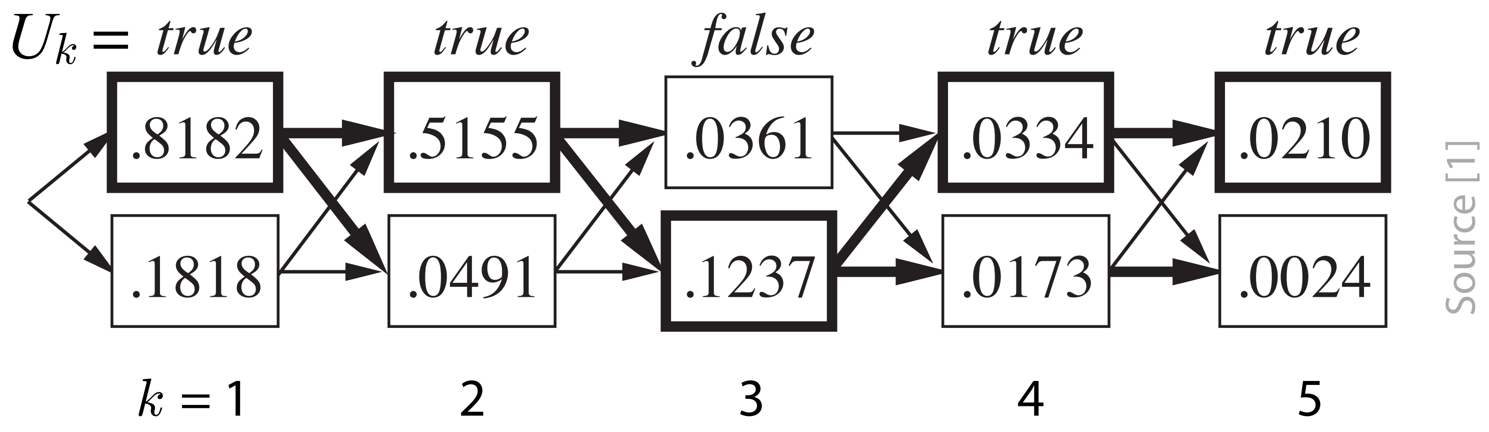


  in each step by maximizing over the previous $\mu_{k,s}$ values for every state s, weighted by the transition matrix A, and then multiplying by the observation model for state s, $p(\mathbf{z}_t|\mathbf{x}_t = s)$
- This is called the **Viterbi algorithm** after its inventor
- Like the filtering algorithm, its time complexity is also **linear** in the length of the sequence

Hidden Markov Models

Inference: Most Likely Sequence Example

- Let us illustrate the Viterbi algorithm in our umbrella example

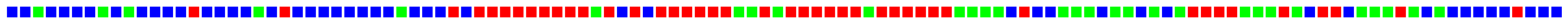


- Values of $\mu_{k,s}$ which give the probability of the **best sequence reaching each state** at time k are shown in the boxes
- **Bold arrows** indicate a state's best predecessor as measured by the product of the preceding $\mu_{k,s}$ probability and the transition probability
- Backtracking: **following the bold arrows back** from the most likely state in μ_5 gives the most likely sequence

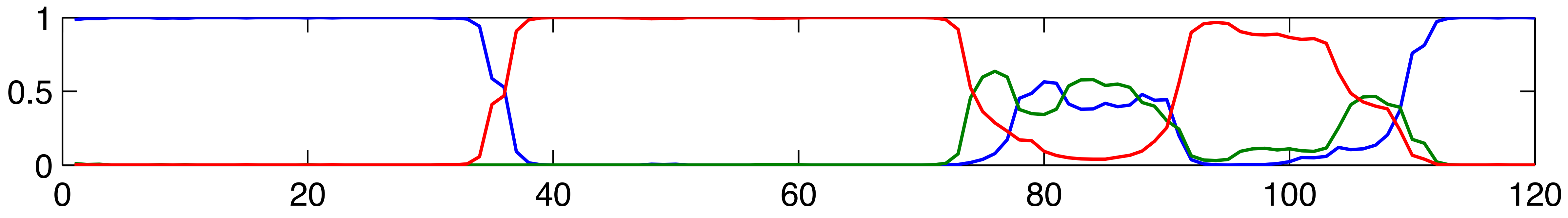
Hidden Markov Models

Inference: Most Likely Sequence Example

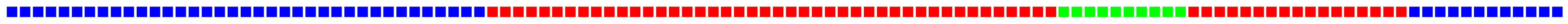
- In our simple human action recognition example, the **sequence of observations** is



- The **smoothed** probabilities



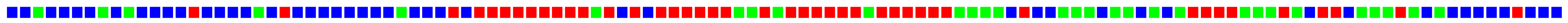
- The ground truth



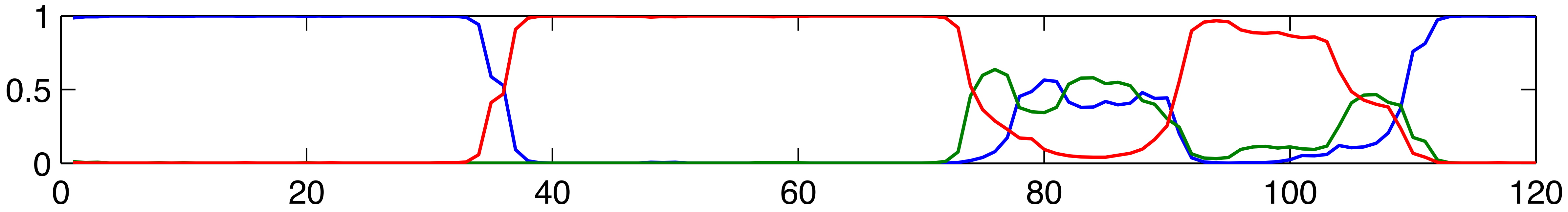
Hidden Markov Models

Inference: Most Likely Sequence Example

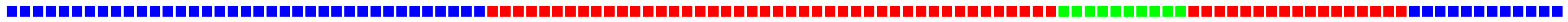
- In our simple human action recognition example, the **sequence of observations** is



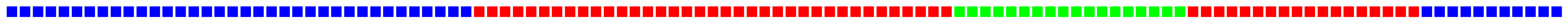
- The **smoothed** probabilities



- The ground truth



- The **most likely sequence**



Temporal Reasoning

Summary

- **Temporal reasoning** deals with the representation, inference and learning of and with sequential data
- The **state space model** describes systems that **evolve on their own**, with **observations of it** occurring in a **separate** process
- State space models possess **three parameters**: the transition model, the observation model, and the prior distribution
- An HMM is a temporal probabilistic model in which the state of the process is described by a **single discrete** random variable
- HMM is a very popular method and widely used in speech recognition, natural language modeling, human activity recognition, on-line handwriting recognition, analysis of protein/DNA sequences, etc.
- Many extensions of the basic HMMs exist

Temporal Reasoning

Summary

- We have considered four HMM inference tasks: filtering, smoothing, prediction, and most likely sequence
  - **Filtering** is done with the **forward algorithm**
  - **Smoothing** is done with the **forward-backward algorithm**
  - **Prediction** is like **filtering without new evidence**
  - Computing the **most likely sequence** is done with the **Viterbi algorithm**
- They are all **linear-time algorithms**
- Smoothing provides better state estimates than filtering due to its incorporation of the "future"
  - Better in terms of **less noise**
  - Better in terms of **shorter reaction times to state changes**

References

Sources and Further Readings

These slides follow three sources: the books by Russell and Norvig [1] (chapter 15), Bishop [2] (chapter 13), and the videos by mathematicalmonk [3]. The seminal tutorial by Rabiner [4] is also highly recommendable.

[1] S. Russell, P. Norvig, "Artificial Intelligence: A Modern Approach", 3rd edition, Prentice Hall, 2009. See http://aima.cs.berkeley.edu

[2] C.M. Bischop, "Pattern Recognition and Machine Learning", Springer, 2nd ed., 2007. See http://research.microsoft.com/en-us/um/people/cmbishop/prml

[3] mathematicalmonk, "(ML 14.4-14.12) Hidden Markov models", mathematicalmonk YouTube channel. Online: http://www.youtube.com/user/mathematicalmonk (Dec 2013)

[4] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE, vol. 77, no. 2, 1989.