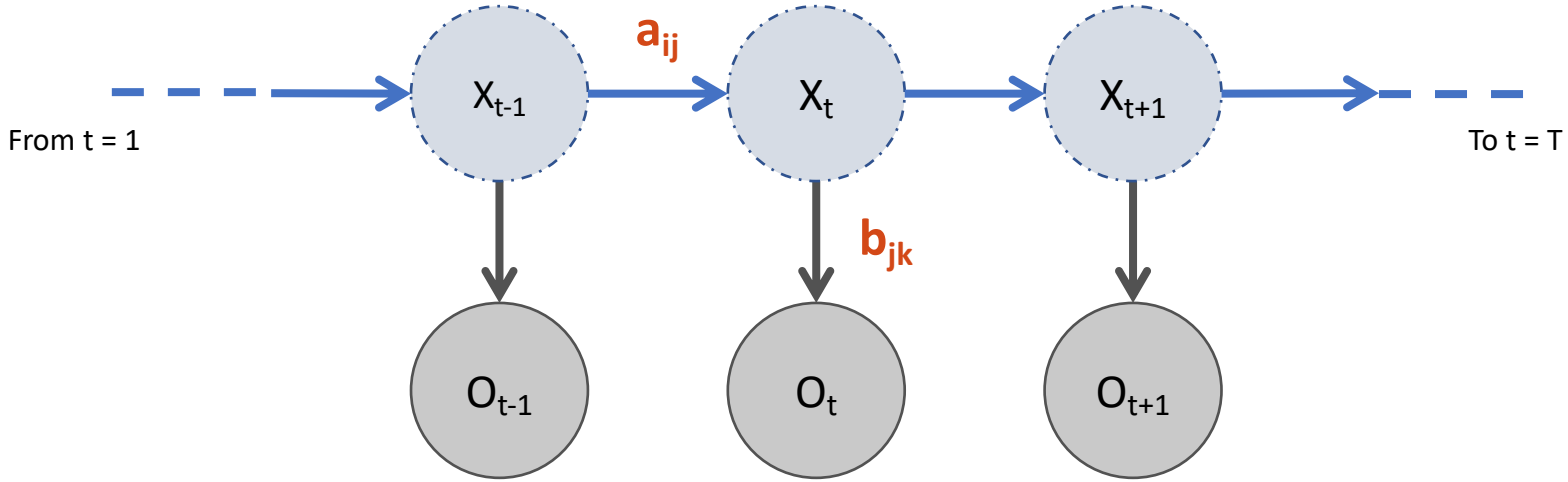
HMM Terminology

Time instants	t *in* {1,2 . . . T}
Hidden States / States / Emitters	X_t
Outputs / Emissions / Observations / Visible States	O_t
All possible states / states set	X_t *in* {1,2 . . . N}
All possible emissions / emissions set	O_t *in* {1,2 . . . K}
Initial state distribution / Initial state probabilities	p_i in q *or* π_i in π
Transition probabilities / State transition probabilities	a_{ij} *in* row-stochastic matrix A
Emission probabilities / Observation probabilities	b_{jk} *in* row-stochastic matrix B



DD2380

Artificial Intelligence

HMMs

Iolanda Leite

Credits

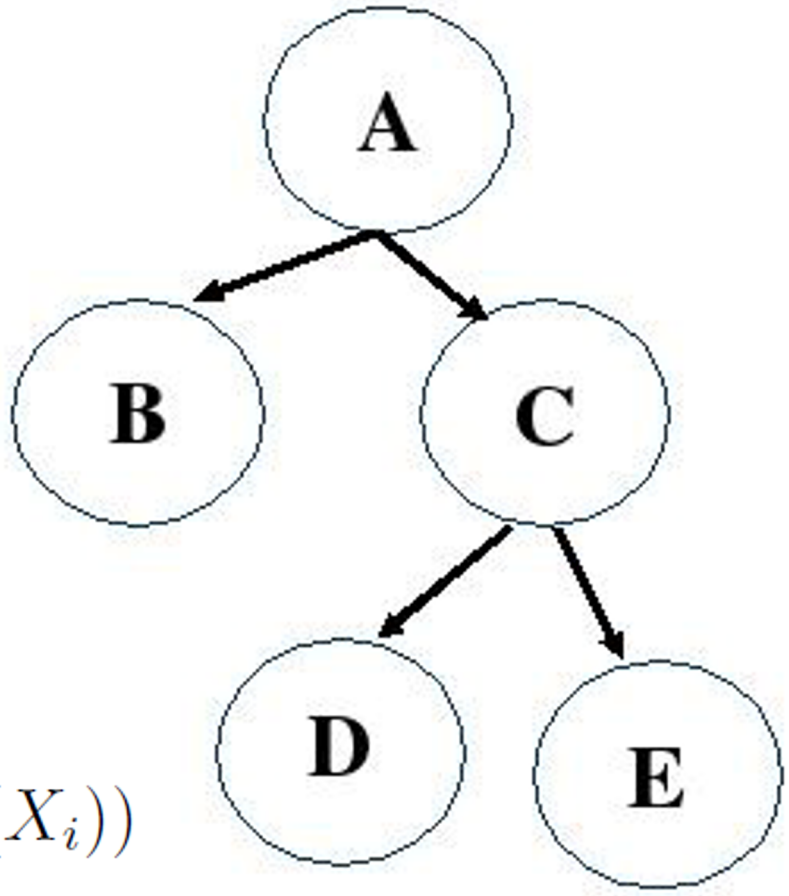
- Original slides from Patric Jenfelt, KTH
- Based partly on material from
  - Francisco Melo, IST Technical University of Lisbon
  - http://ai.berkeley.edu

Reading instructions

- Chapters 13-15 in the book
- Stamp tutorial on HMMs on the course web page

Bayesian network

- Aka Probabilistic **Directed Acyclic** Graphical Model
- Represents joint probability distribution (in a compact manner)
- Arrow → "direct influence over"
- A has direct influence over B
- Each arrow is accompanied with a conditional probability distribution, e.g. p(B|A)



- Factorization

$$p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{i=n} p(X_i | Parents(X_i))$$

Bayesian network

- Compact representation of the joint distribution over a set of variables

- Graphical representation that helps
  - **analyze** prob. Information
  - **structure** prob. Information

- Very hard to gather data to build a model for p(A,B,C,D,E), much easier to look at conditional probabilities such as p(B|A)

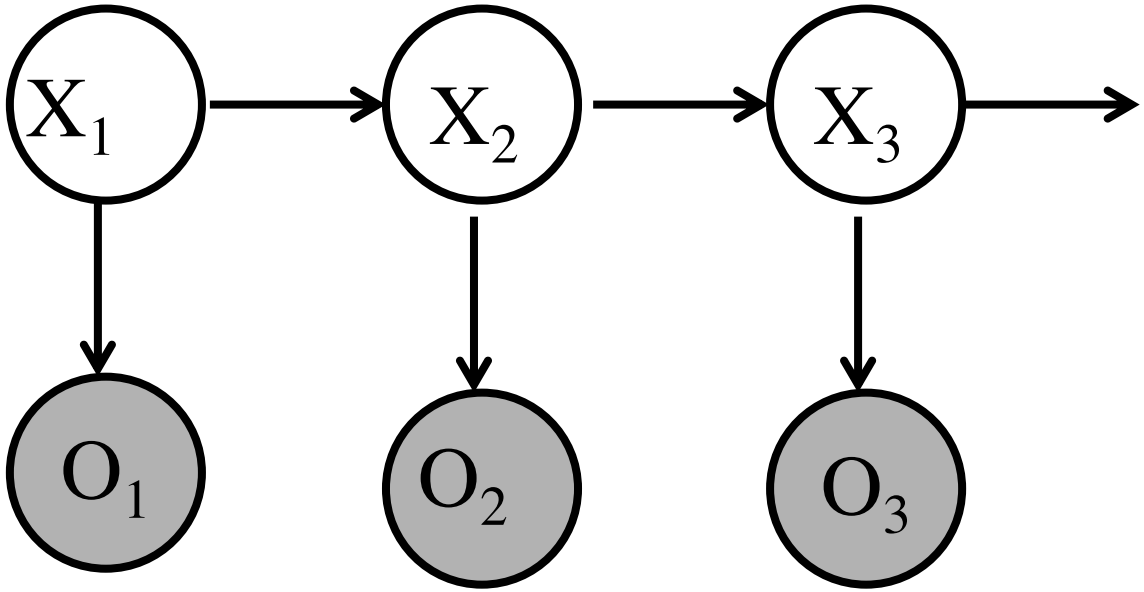
Markov model

- The (first order) Markov assumption

–The distribution $p(X_t)$ depends only on the distribution $p(X_{t-1})$

–The present (current state) can be predicted using local knowledge of the past (state at the previous step)

Hidden Markov Models (HMMs)



Hidden states

Observations
(Often marked as filled)

- State transition model:
- $p(X_t=j|X_{t-1}=i)=A(i,j)=a_{ij}$
- Observation model:
- $p(O_t=j|X_t=i)=b_{ij}$

Elements of discrete HMM

1. Number of states N, $x \in \{1, \ldots, N\}$;

2. Number of events K, $k \in \{1, \ldots, K\}$;

3. Initial-state probabilities,
   $\pi = \{\pi_i\} = \{P(x_1 = i)\}$ for $1 \leq i \leq N$;

4. State-transition probabilities,
   $A = \{a_{ij}\} = \{P(x_t = j | x_{t-1} = i)\}$ for $1 \leq i, j \leq N$;

5. Discrete output probabilities,
   $B = \{b_i(k)\} = \{P(o_t = k | x_t = i)\}$ for $1 \leq i \leq N$ and $1 \leq k \leq K$.

Elements of discrete HMM: π

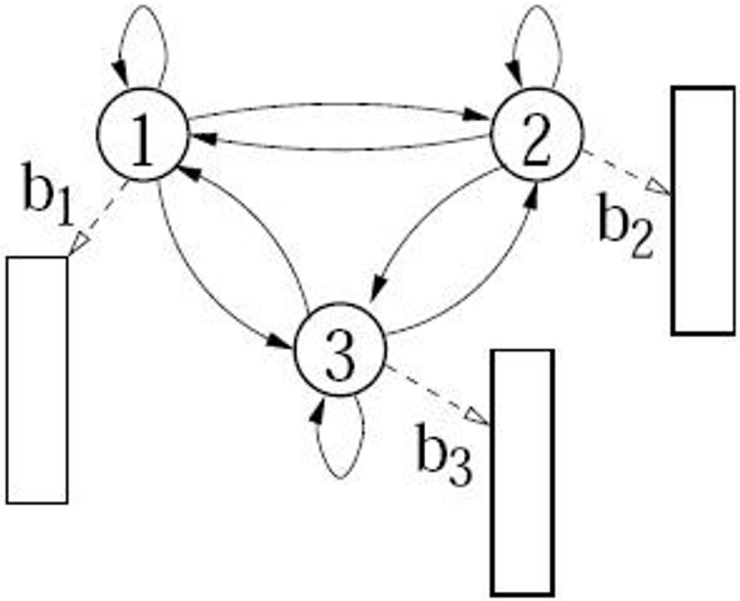
- Initial Distribution : contains the probability of the (hidden) model being in a particular hidden state at time t = 1 (sometimes t=0).
- Often referred to as π

- Ex: π=[0.5 0.2 0.3], i.e.,

$p(X_1=1)=0.5$

$p(X_1=2)=0.2$

$p(X_1=3)=0.3$



Elements of discrete HMM: A

- State transition matrix : holding the probability of transitioning from one hidden state to another hidden state.
- Ex: a_{21} gives $p(X_{t+1}=1|X_t=2)$, i.e. probability to transition from state 2 to state 1

	$X_{t+1}=1$	$X_{t+1}=2$	...	$X_{t+1}=N$
$X_t=1$	a_{11}	a_{12}	...	a_{1N}
$X_t=2$	a_{21}	a_{22}	...	a_{2N}
...	...	...	...	...
$X_t=N$	a_{N1}	a_{N2}	...	a_{NN}

Elements of discrete HMM: B

- Output matrix : containing the probability of observing a particular measurement given that the hidden model is in a particular hidden state.

	$O_t=1$	$O_t=2$	...	$O_t=K$
$X_t=1$	$b_1(1)$	$b_1(2)$	...	$b_1(K)$
$X_t=2$	$b_2(1)$	$b_2(2)$	...	$b_2(K)$
...	...	...	...	...
$X_t=N$	$b_N(1)$	$b_N()$	...	$b_N(K)$

- $b_j(O_t)$ is the probability to observe O_t in state j

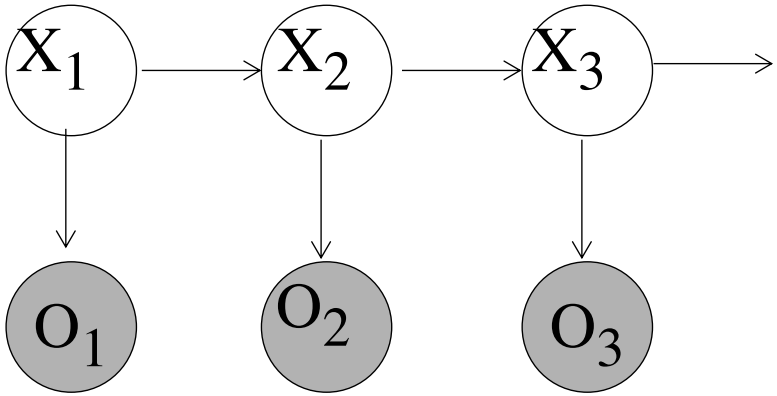
Connect BN and HMMs

- The two diagrams below show different things

BN

Arrows: Dependency between variables

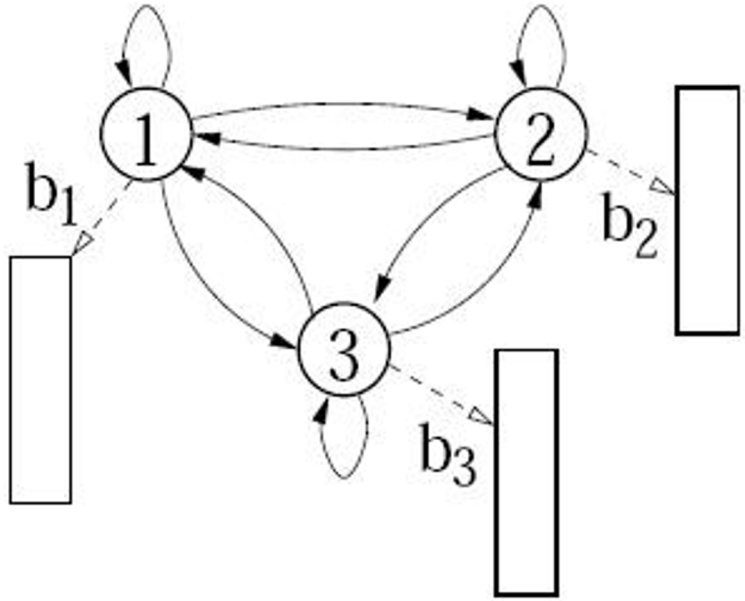
Circles: Variables



HMM

Arrows: State transitions and observation probs

Circles: The different states



Example: Umbrella world

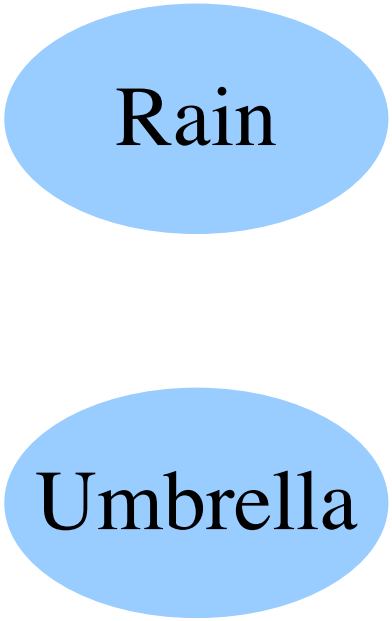
- A person tries to infer the weather outside (rain={true,false}) by observing if a certain person has an umbrella={true,false} that day.

- Draw Bayesian network!
  - What are the variables?
  - Which cause which?




Example: Umbrella world

- A person tries to infer the weather outside (rain={true,false}) by observing if a certain person has an umbrella={true,false} that day.
- Draw Bayesian network



Here, we cannot directly observe if it rains. This is **hidden** to us. We observe the umbrella and infer if it rains or not. Rain causes the person to use an umbrella

Example: Umbrella world

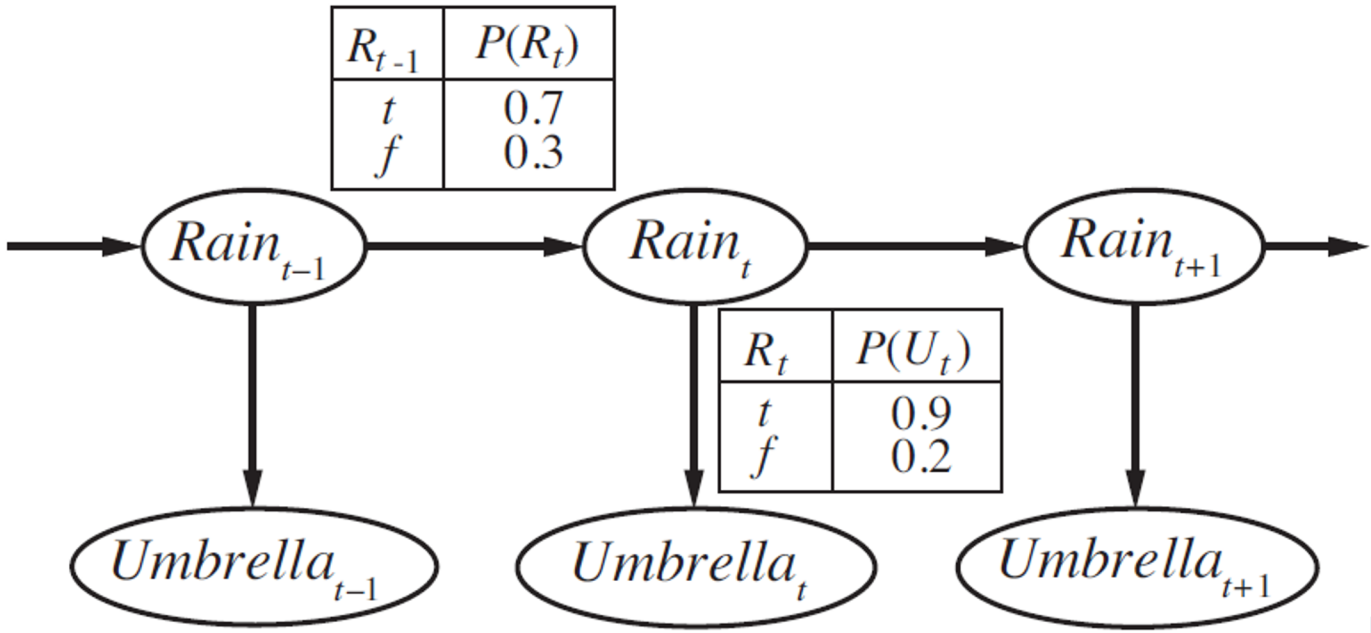
- A person tries to infer the weather outside (rain={true,false}) **every day** by observing if a certain person has an umbrella={true,false} that day.

- Draw the Bayesian network that corresponds to the **time sequence** (the person makes observations over consecutive days).

- What additional assumptions did you make?

Example: Umbrella world

- At each time step the state can be either Rain=true or Rain=false. The observation depends directly on the state.
- Additional assumption: The next state depends only on the previous state (first order Markov assumption).



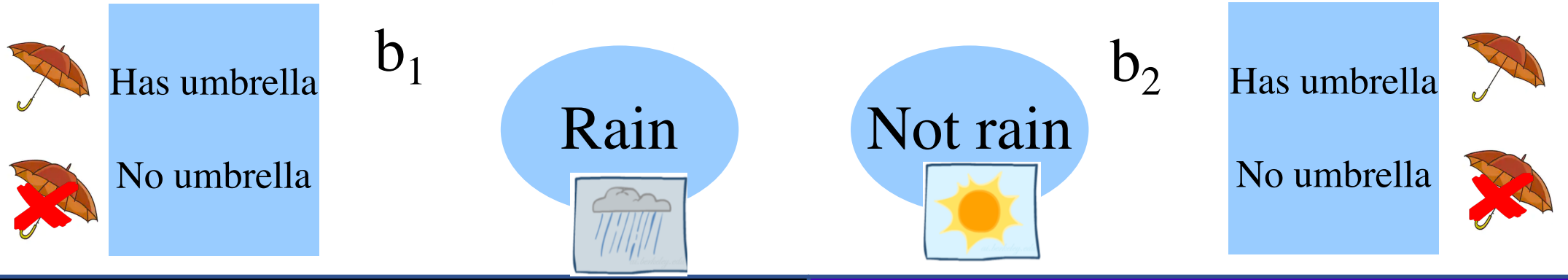
Example: Umbrella world

- A person tries to infer the weather outside (rain={true,false}) **every day** by observing if a certain person has an umbrella={true,false} that day. Formulate as an HMM.

–How many states? Which?
–How many observations? Which?

Example: Umbrella world

- A person tries to infer the weather outside (rain={true,false}) **every day** by observing if a certain person has an umbrella={true,false} that day. Formulate as an HMM.
  - States: {Rain=true, Rain=false}
  - Observations: {Umbrella=true, Umbrella=false}



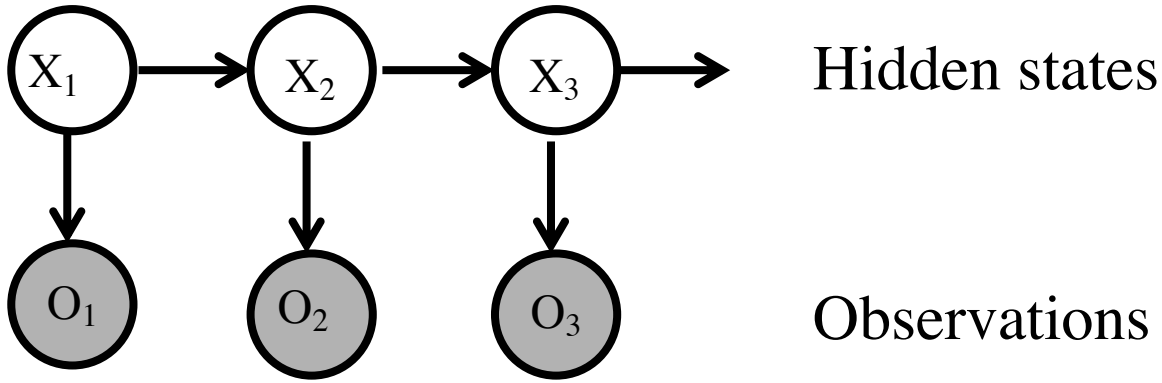
What is an HMM, again...

- It is a **model** (not necessarily a perfect one) to describe a system / data from a system

- It can be used to (e.g.)
  - Generate predictions about how the system will behave. Ex: stock market
  - Analyze if a sequence of measurements match a certain model, e.g., did the person say "Bayesian" (i.e. match our model for how that sounds) or "Bearnaise" (match that model)?
  - Learn something about a system by learning the model parameters.

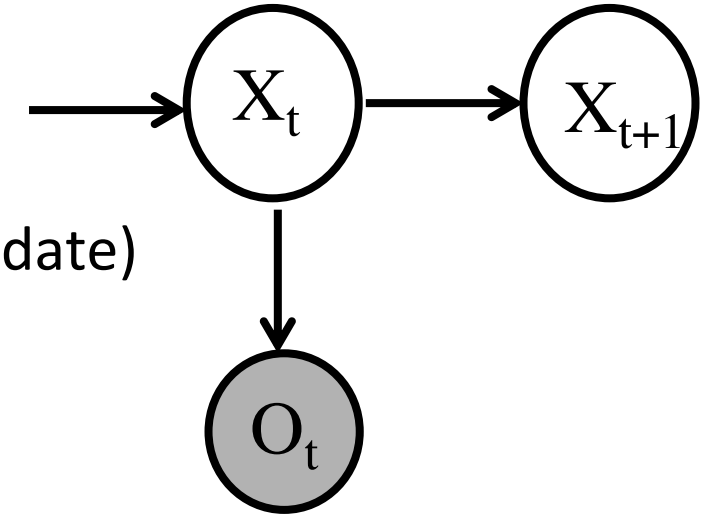
Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process: future depends on past via the present
  - Current observation independent of all else given current state



- Does this mean that evidence variables are guaranteed to be independent?
  - [No, they tend to correlated by the hidden state]

Prediction in HMMs



- Assume we have current belief P(X | evidence to date)

$$p(X_t \mid O_{1:t})$$

- Then, after one time step passes:

$$p(X_{t+1} \mid O_{1:t}) = \{sum\ rule\} = \sum_{X_t} p\,(X_{t+1}, X_t \mid O_{1:t})$$

$$= \{product\ rule\} = \sum_{X_t} p\,(X_{t+1} \mid X_t\,, O_{1:t})\; p(X_t \mid O_{1:t})$$

P(A,B|C) = P(A|B,C) P(B|C)

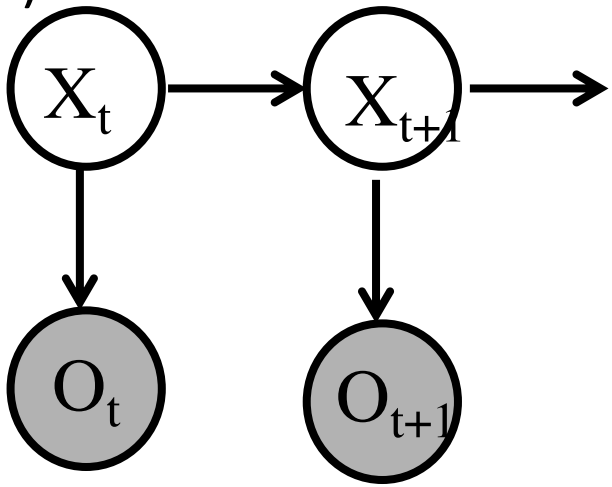
$$= \{O_{1:t}\ cond.indep\ of X_{t+1}\ given\ X_t\} = \sum_{X_t} p\,(X_{t+1} \mid X_t\,)\; p(X_t \mid O_{1:t})$$

- Basic idea: beliefs get "pushed" through the transitions

Measurements/observations in HMMs

- Assume we have current belief P(X | previous evidence):

$$p(X_{t+1} \mid O_{1:t})$$



- Then, after evidence comes in:

$$p(X_{t+1} \mid O_{1:t+1}) = \{\text{"split" } O\} = p(X_{t+1} \mid O_{t+1}, O_{1:t})$$

$$= \{Bayes\ rule\} = \frac{p(O_{t+1} \mid X_{t+1}, O_{1:t})\, p(X_{t+1} \mid O_{1:t})}{\sum_{X_{t+1}} p(O_{t+1} \mid X_{t+1}, O_{1:t})\, p(X_{t+1} \mid O_{1:t})}$$

$$= \{O_{t+1} \text{ cond. indep of } O_{1:t} \text{ given } X_{t+1}\} = \frac{p(O_{t+1} \mid X_{t+1})\, p(X_{t+1} \mid O_{1:t})}{\sum_{X_{t+1}} p(O_{t+1} \mid X_{t+1})\, p(X_{t+1} \mid O_{1:t})}$$

Bayes rule

$$p(Y|X) = p(X|Y)\, p(Y) \,/\, \sum_{Y} p(X|Y)p(Y)$$

Look at
Ex: Usain Bolt

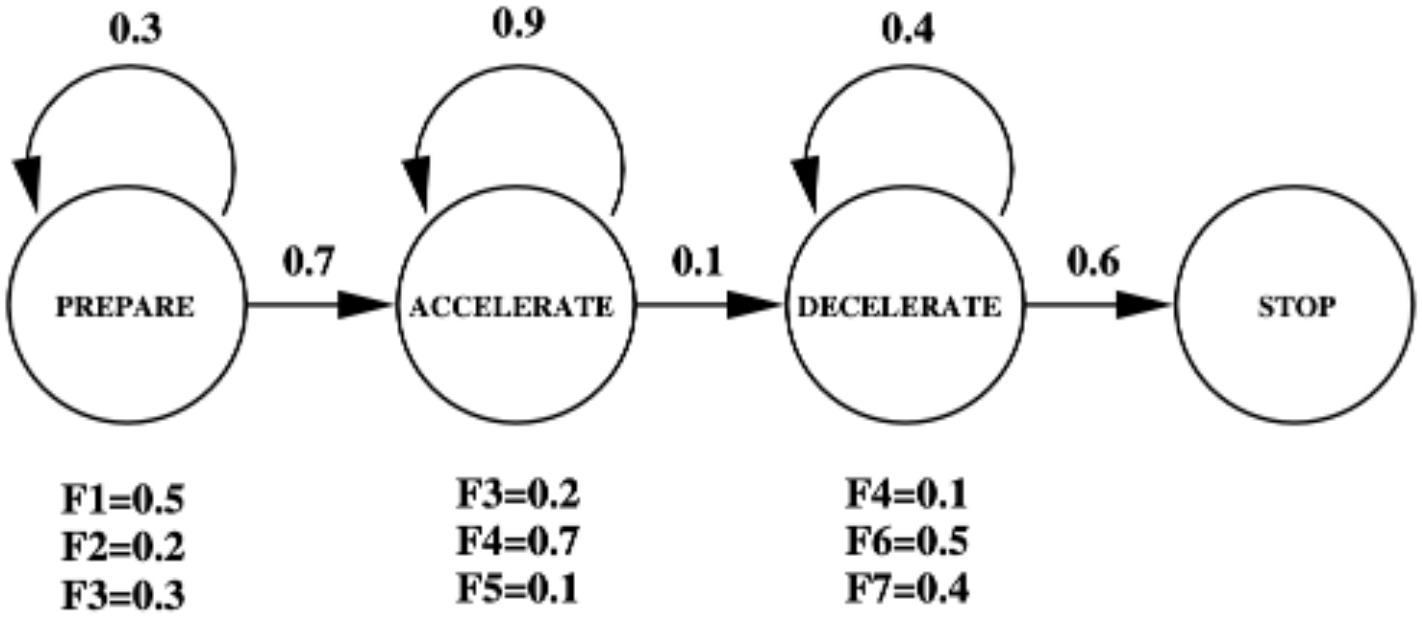
What is A,B,π?
Answer Q1-4

HMM example: Usain Bolt

- You have developed an automatic video annotation system for annotating recorded running sequences of Usain Bolt. The system takes images from a video stream of a running sequence as an input, it extracts some visual data and annotates each image as:

  - Usain is preparing for running
  - Usain runs/accelerates
  - Usain decelerates

HMM example: Usain Bolt

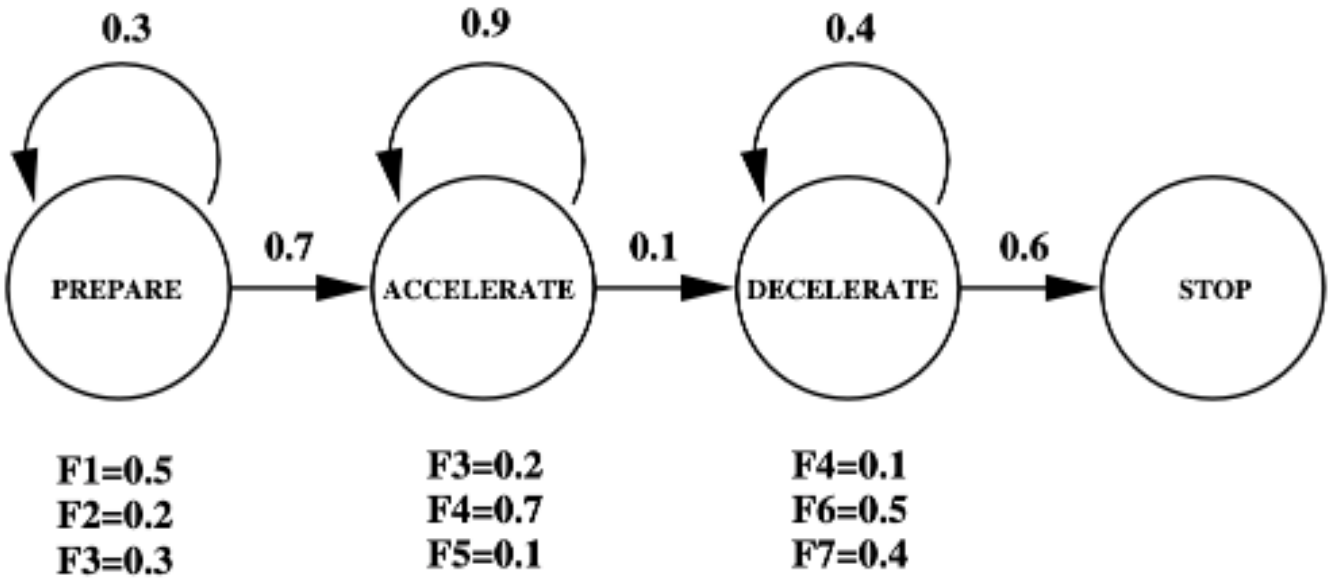


- F1...7 represent seven ways of classifying each video frame (i.e. seven different possible observation outcomes that might have to do with Usain's posture) and their probabilities of being observed in each state. If an observation is not mentioned under a state it means it has probability zero of being observed there.
- STOP is a non-emitting terminating state.
- The system divides a video sequence in several video frames and each of them is classified to one of the F1...F7.

Ex: Usain Bolt

- What does A, B and π look like?

(4th state non-emitting / silent)



$$A = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.5 & 0.2 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.7 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0.5 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\pi = [1 \quad 0 \quad 0 \quad 0]$$

p(X_2=ACCELERATE)?

- Start with π and propagate probability one step with A!
- $\pi=X_1=[1\ 0\ 0\ 0] \rightarrow X_2=\pi A=[0.3\ 0.7\ 0\ 0]$
- $\rightarrow$ p(X_2=ACCELERATE)=0.7

$$A = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.5 & 0.2 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.7 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0.5 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\pi = [1 \quad 0 \quad 0 \quad 0]$$

$p(O_2=F2)$?

- We know the distribution for X_2
- $p(X_2)=[0.3\ 0.7\ 0\ 0]$
- and we know how probable each measurement is in each state. Use sum rule!!

$$p(0_2 = F_2) = \sum_{i=1}^{N} p\,(O_2 = F_2 \mid X_2 = i)p(X_2 = i)$$
$$= 0.2 \cdot 0.3 + 0 \cdot 0.7 + 0 \cdot 0 + 0 \cdot 0 = 0.06$$

$$B = \begin{bmatrix} 0.5 & \boxed{0.2} & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.7 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0.5 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

F2

$p(X_2=ACCELERATE \mid O_2=F_2)$?

- Use Bayes rule

$$p(X_2 = A \mid O_2 = F_2) = \frac{p(O_2 = F_2 \mid X_2 = A)p(X_2 = A)}{p(O_2 = F_2)} = \frac{0 \cdot 0.7}{0.06} = 0$$

$p(X_2)$=[0.3 0.7 0 0]

$$B = \begin{bmatrix} 0.5 & 0.2 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.7 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0.5 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

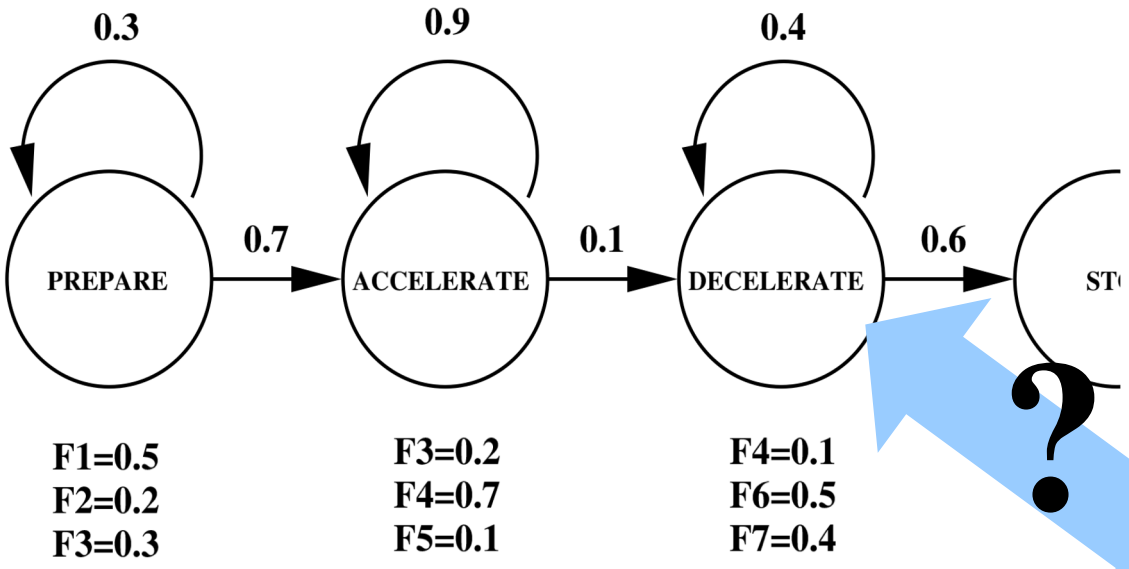
F2

$p(0_2 = F_2) = 0.06$

- Could we have seen this immediately?

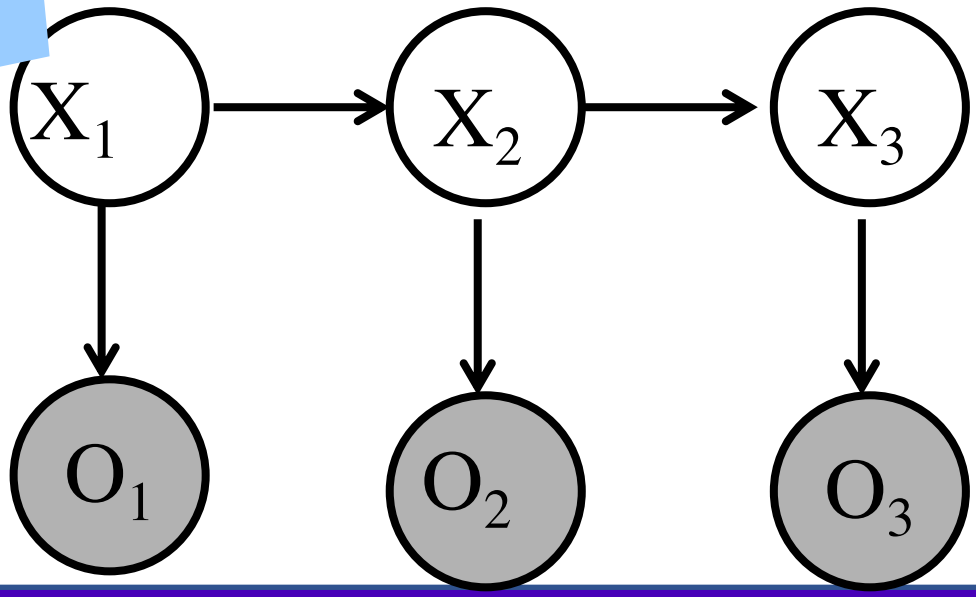
–YES!! Cannot measure F2 in Accelerate state, i.e., p=0

Ex: Usain Bolt

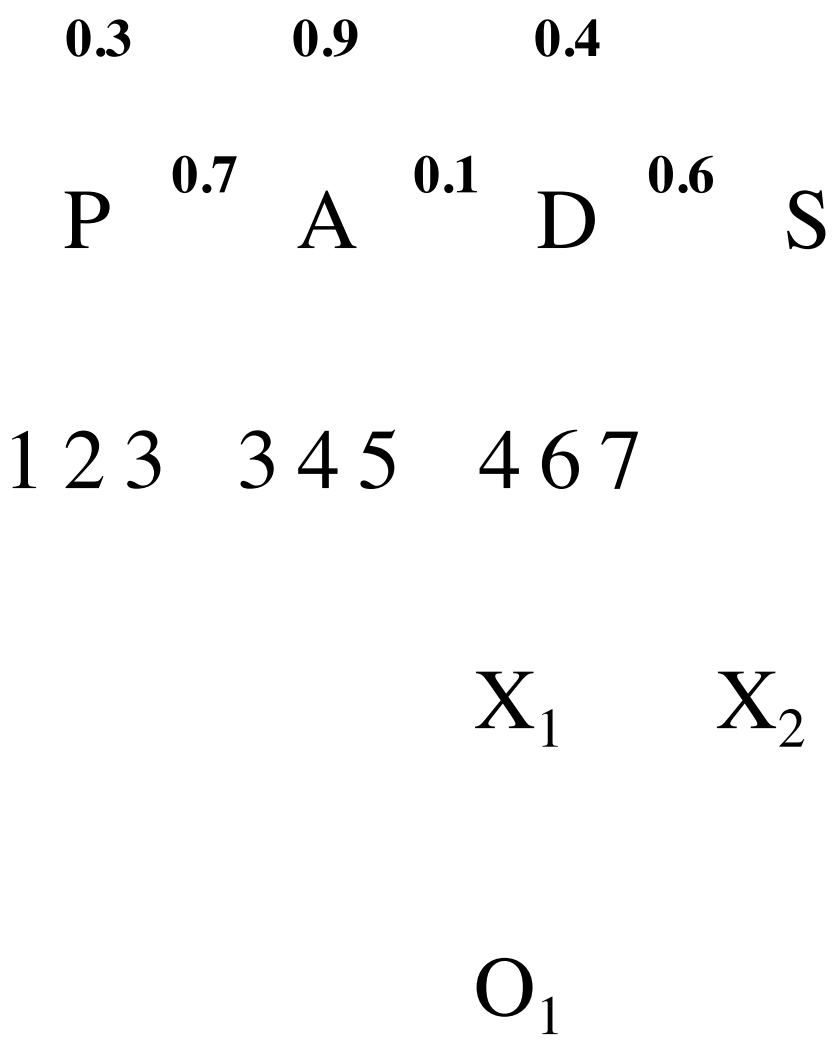


?

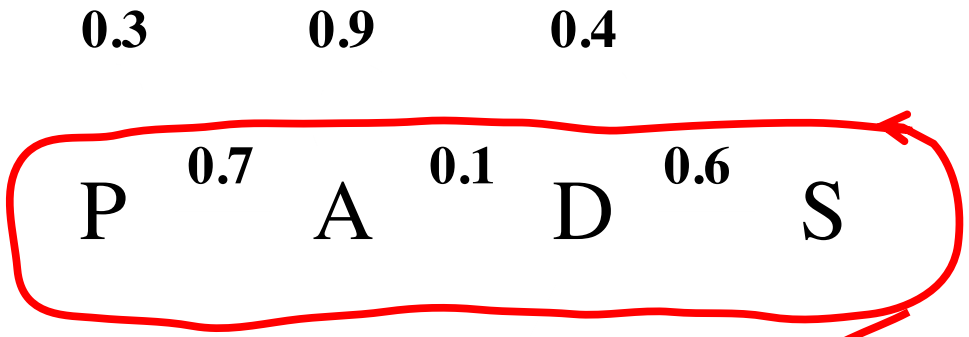
Let's look at the connection between the two views



Let's have a closer look!



The states



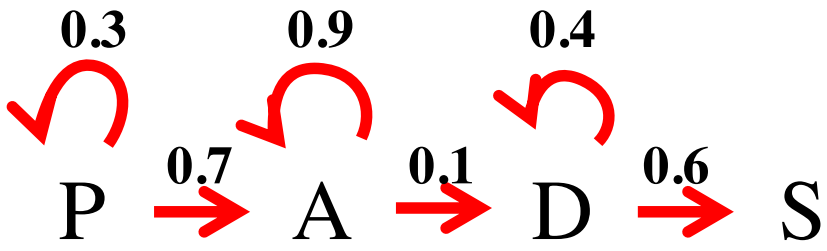
1 2 3 3 4 5 4 6 7

The states that variables $X_1, X_2, \ldots$ can be in (remember discrete)

X_1 X_2

O_1

The state transitions



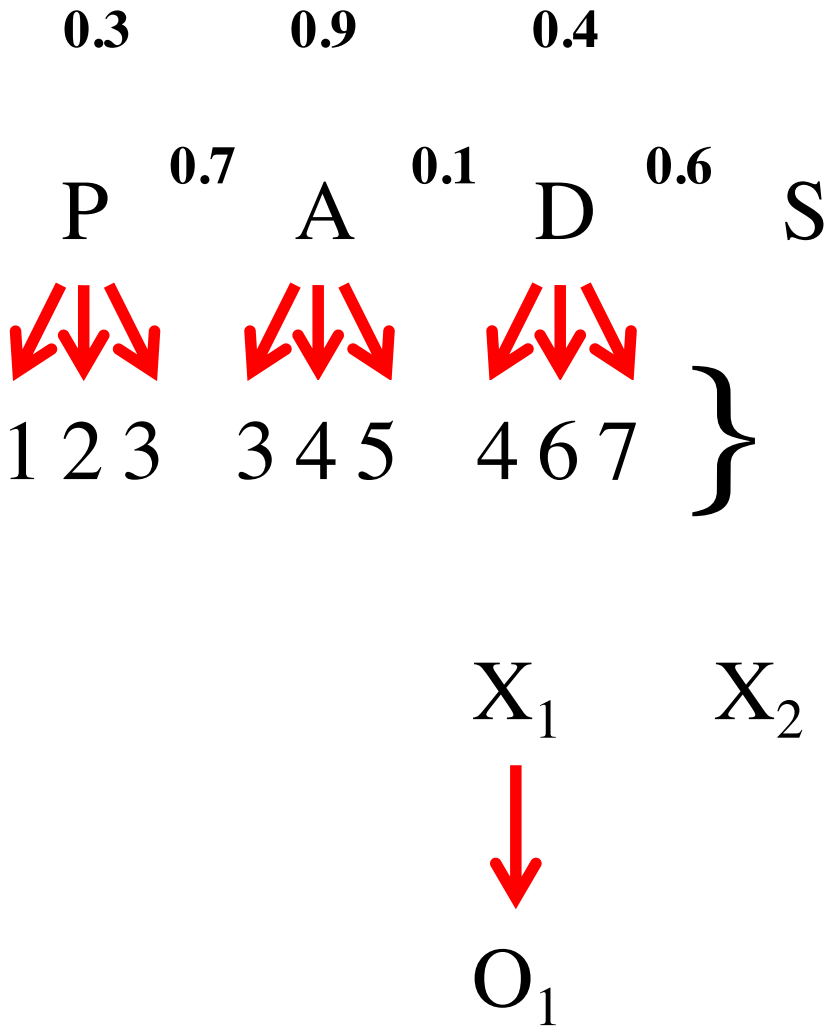
1 2 3 3 4 5 4 6 7

$X_1 \rightarrow X_2$

O_1

Red lines describe $p(x_{k+1}|x_k)$
i.e., the state transitions.
Captured with the A matrix (top)
and $p(x_{k+1}|x_k)$ (bottom)

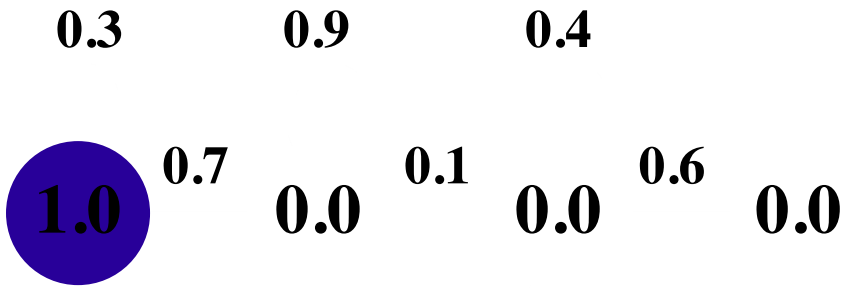
The observation/emission model



Different possible observations, (aka "symbols")

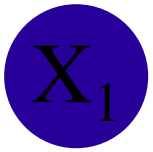
Red lines describe $p(O_k|X_k)$ i.e., observation model. Captured by matrix B (top) and $p(O_k|X_k)$ (bottom).

Initial state π=[1 0 0 0]



1 2 3 3 4 5 4 6 7

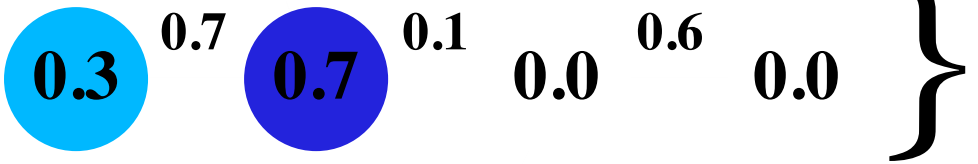
The darker the higher probability



O_1

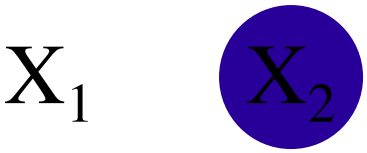
Prediction for X_2 ($= \pi$ A)

0.3 0.9 0.4



} Must sum to 1 i.e., the system is in some state!!

1 2 3 3 4 5 4 6 7



O_1 O_2

Prediction for X_3 ($=X_2A$)

0.3 0.9 0.4

0.09 0.7 0.84 0.1 0.07 0.6 0.0 } Must sum to 1 i.e., the system is in some state!!

1 2 3 3 4 5 4 6 7

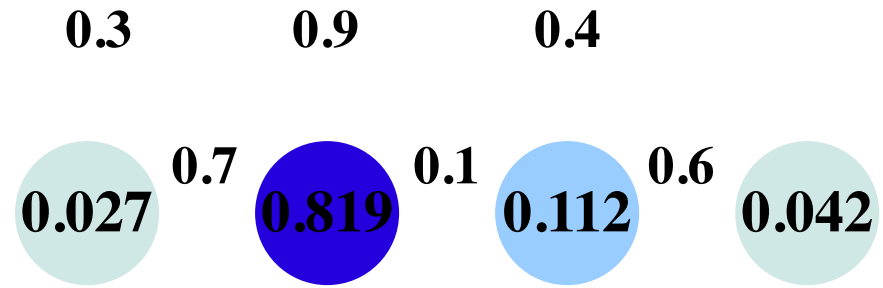
X_1 X_2 X_3

O_1 O_2 O_3

$$X_2 = [0.3 \quad 0.7 \quad 0 \quad 0]$$

$$A = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Prediction for X_4



1 2 3 3 4 5 4 6 7

Note: We know for sure that we are at t=4 but not in which state we are. This is what we often try to estimate!

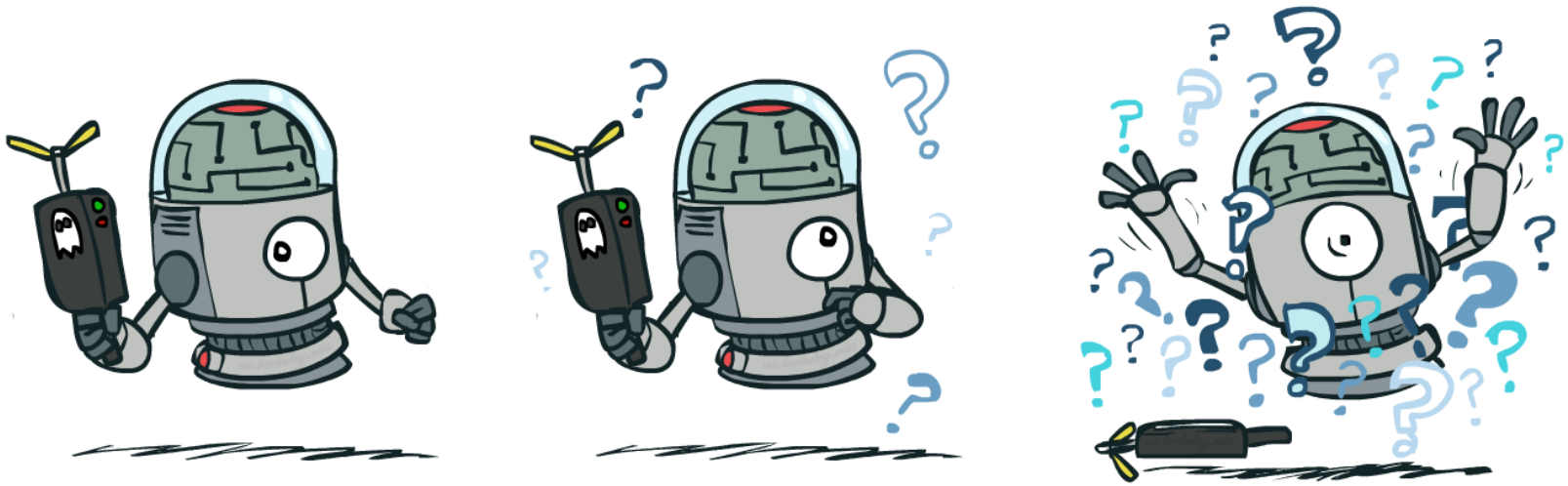
X_1 X_2 X_3 X_4

O_1 O_2 O_3 O_4

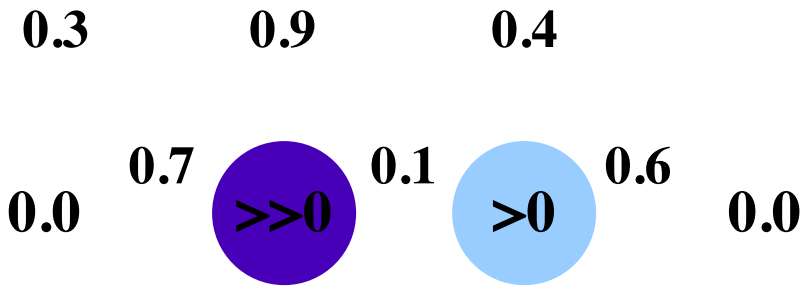
Measurements?

- Without measurements we are doing pure prediction
  - can do that ahead of time!

- Measurements give clues to what state we are in
  - beliefs get reweighted, uncertainty "decreases"

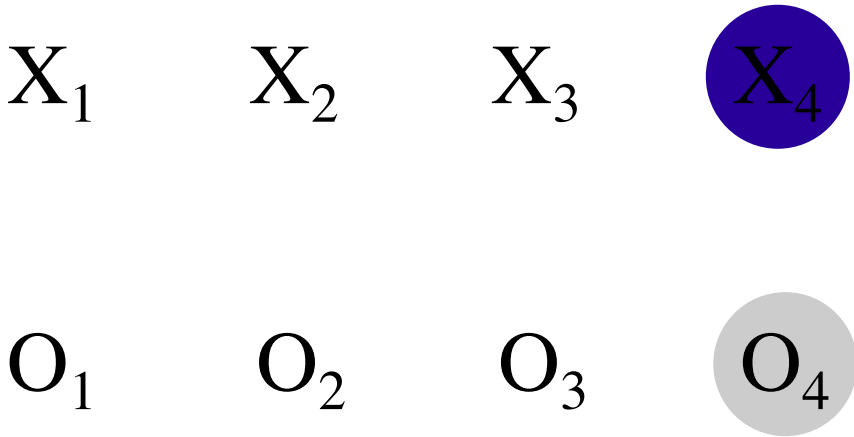


Assume we measure $O_4 = 4$ (i.e. obs 4 at t=4)



1 2 3 3 4 5 4 6 7

Weight with p(O|X)
→ not in state 1 for sure because we cannot measure 4 there



Observed (grey)

How to calculate $p(x_4|O_4)$?

- We have $p(X_4)$ from the prediction
- For each state j we can calculate $p(X_4=j|O_4)$
-

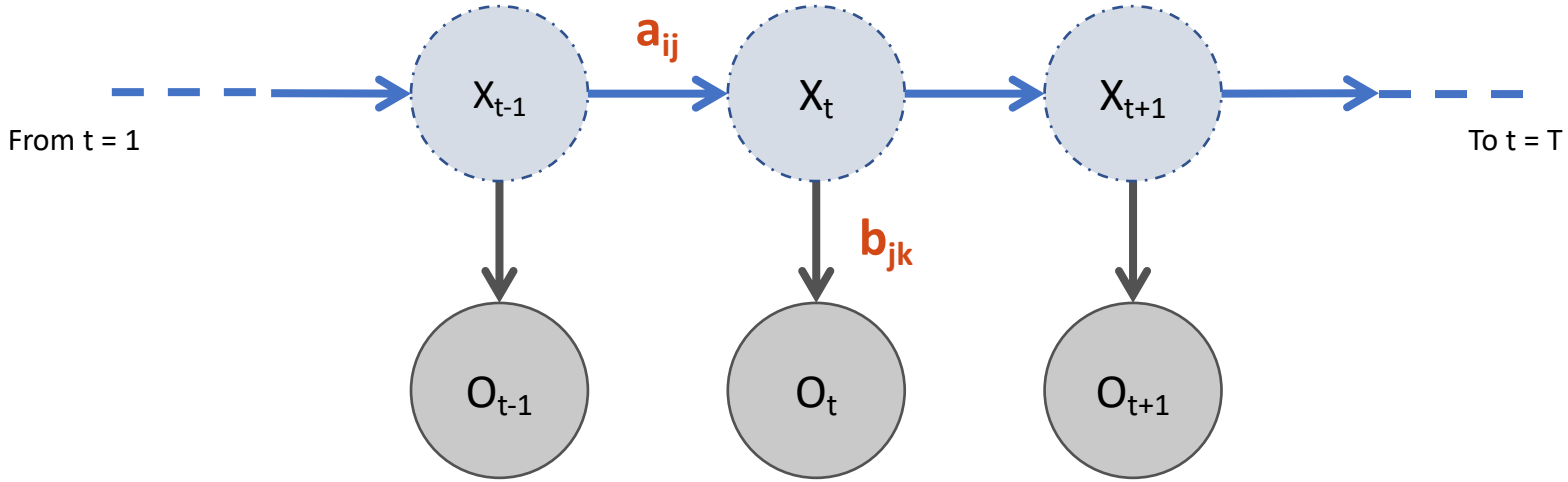
$$\begin{aligned} p(X_4 = j \mid O_4) &= \{Bayesrule\} \\ &= \frac{p(O_4 \mid X_4 = j)p(X_4 = j)}{p(0_4)} \\ &= \frac{b_j(4)p(X_4 = j)}{p(0_4)} \\ &= \eta b_j(4)p(X_4 = j) \end{aligned}$$

- Calculate η by normalization so that the above terms sum to 1 for all j.

$$p(X_4 = \mathrm{P} \mid O_4 = 4) + p(X_4 = \mathrm{A} \mid O_4 = 4) + p(X_4 = \mathrm{D} \mid O_4 = 4) + p(X_4 = \mathrm{S} \mid O_4 = 4) = 1$$

HMM Terminology

Time instants	t *in* {1,2 . . . T}
Hidden States / States / Emitters	X_t
Outputs / Emissions / Observations / Visible States	O_t
All possible states / states set	X_t *in* {1,2 . . . N}
All possible emissions / emissions set	O_t *in* {1,2 . . . K}
Initial state distribution / Initial state probabilities	p_i in q *or* π_i in π
Transition probabilities / State transition probabilities	a_{ij} *in* row-stochastic matrix A
Emission probabilities / Observation probabilities	b_{jk} *in* row-stochastic matrix B



Three problems solved with HMMs

1. **Evaluation/**filtering: Compute likelihood $p(O_{1:t}|\lambda)$ of observation sequence ($O_{1:T}=\{O_1,O_2,\ldots,O_t,\ldots,O_T\}$) given λ

   Forward algorithm

2. **Decoding/**smoothing: Most likely state sequence $X^*_{1:T}$ given $O_{1:T}$ and λ

   Viterbi algorithm

3. **Learning**: Estimate model parameters $\Lambda=\{\lambda\}$ given $O_{1:T}$ such that $p(O_{1:T}|\lambda)$ is maximized
   $\rightarrow$ A and B matrices and π

   Baum-Welch algorithm

Three problems solved with HMMs cont'd

- Note:

–I will display formulas for completeness but not derive them and not spend time on explaining most them

–I will gloss over the details and target a conceptual understanding

–Go back later for more details

- HMM tutorial sessions (Lektion)
- HMM tutorials by Mark Stamp
- Course book

1. Compute likelihood $p(O_{1:t}|\lambda)$ of observation sequence given λ

- Motivating examples

–Character recognition:

- Have models for each character
- Draw a character and **compare it to models of different characters**
- Pick model that fits best → Recognized char

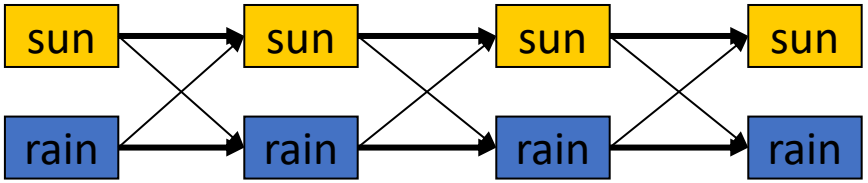
–A1: Identify fish species

- Model swimming patterns of known fish, **compare new fish to the models**

1. Compute likelihood $p(O_{1:t}|\lambda)$ of observation sequence given λ

- **Given:**
  - A, B, π
  - Emission sequence **O** = $\{O_1, O_2 \dots O_T\}$

- **Unknown:**
  - Hidden state sequence **X** = $\{X_1, X_2 \dots X_T\}$ that actually produced **O**.



- **To Find:**

$X_1 \quad X_2 \quad \cdots \quad X_T$

  - Probability that the given sequence **O** occurred **regardless of which X produced the sequence.**

Likelihood of $p(O_{1:T}|\lambda)$

$$p(O_{1:T} \mid \lambda) = \{sum\ rule\} = \sum_{X_{1:T}} p\,(O_{1:T}, X_{1:T} \mid \lambda)$$

$$= \{product\ rule\} = \sum_{X_{1:T}} p\,(O_{1:T} \mid X_{1:T}, \lambda) p(X_{1:T} \mid \lambda)$$

$$= \sum_{X_{1:T}} \underbrace{\pi_{X_1} a_{X_1 X_2} a_{X_2 X_3} \ldots a_{X_{T-1} X_T}}_{p(X_{1:T}|\lambda)} \underbrace{b_{X_1}(O_1) b_{X_2}(O_3) \ldots b_{X_T}(O_T)}_{p(O_{1:T}|X_{1:T},\lambda)}$$

- Note that we are summing over all possible permutations of $X_{1:T}$
- Evaluating this requires $O(2TN^T)$ multiplications
- Can be formulated recursively using the forward (and backward) algorithm

Forward algorithm (aka α-pass)

- Introduce: $\alpha_t(i) = p(O_{1:t}, X_t = i \mid \lambda) \forall t = 1, \ldots, T$

  $\alpha_t(i)$ is the probability of observing a partial sequence of observables O_1, … O_t AND at time t, being in state i

- Initialize as: $\alpha_1(i) = \pi_i b_i(O_1)$

- For 2≤t≤T: $\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}\right] b_i(O_t)$

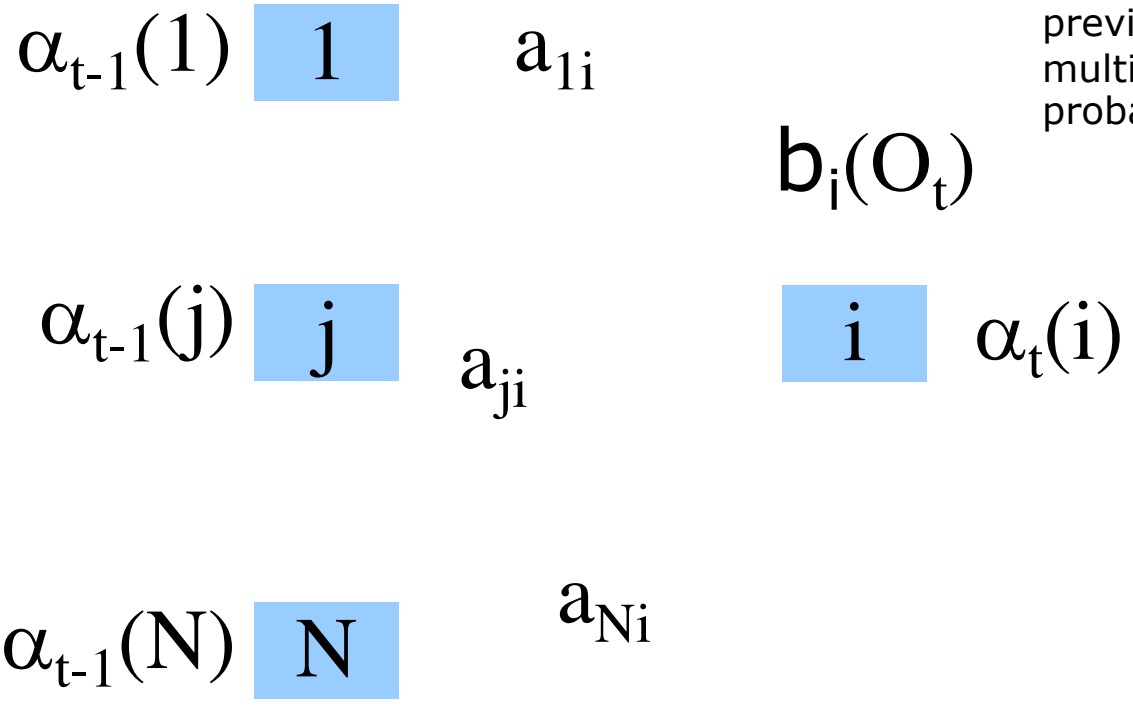
One step in the forward algorithm

$$\alpha_t(i) = p(O_{1:t}, X_t = i \mid \lambda) \forall t = 1, \ldots, T$$

$$\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}\right] b_i(O_t)$$

Weight (observation probability)

Prediction (sum of the previous partial probabilities multiplied by the transition probabilities)



Forward algorithm (aka α-pass)

- Introduce: $\alpha_t(i) = p(O_{1:t}, X_t = i \mid \lambda) \forall t = 1, \ldots, T$

  $\alpha_t(i)$ is the probability of observing a partial sequence of observables O_1, … O_t AND at time t, being in state $X_t = i$

- Initialize as: $\alpha_1(i) = \pi_i b_i(O_1)$

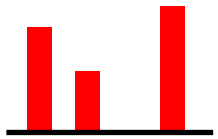
- For 2≤t≤T: $\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}\right] b_i(O_t)$

- Which gives us: $p(O_{1:T} \mid \lambda) = \sum_{i=1}^{N} p(O_{1:T}, X_t = i \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$

- → a recursive way to calculate likelihood with only N^2T multiplications (compared to $2TN^T$)

Forward algorithm intuition

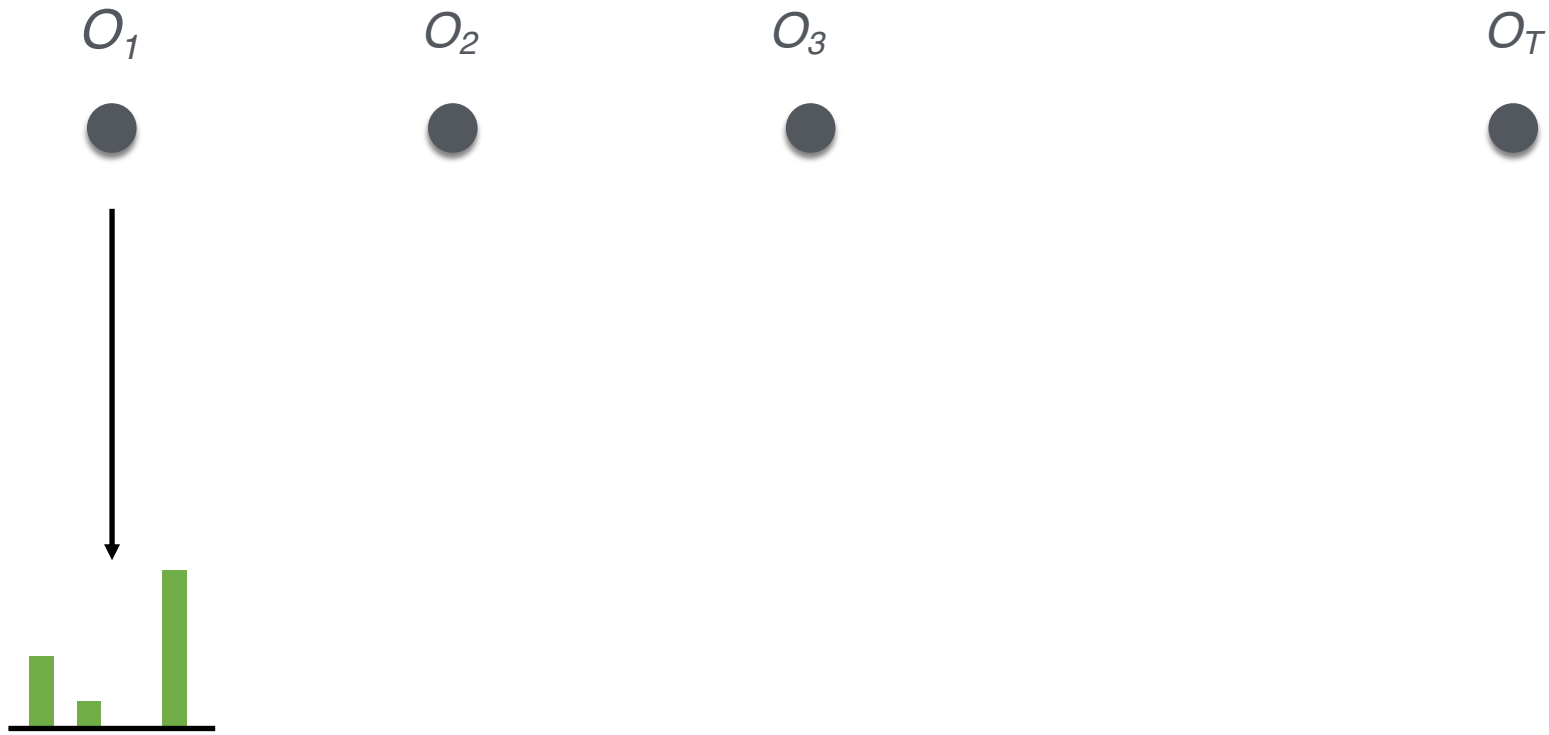
O_1 O_2 O_3 O_T



Initial distribution

π_i

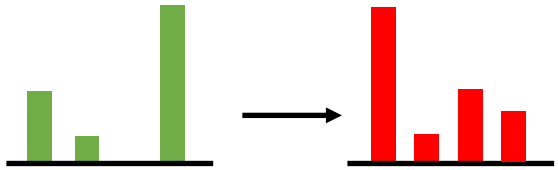
Forward algorithm intuition



$$\alpha_1(i) = \pi_i b_i(O_1)$$

Forward algorithm intuition

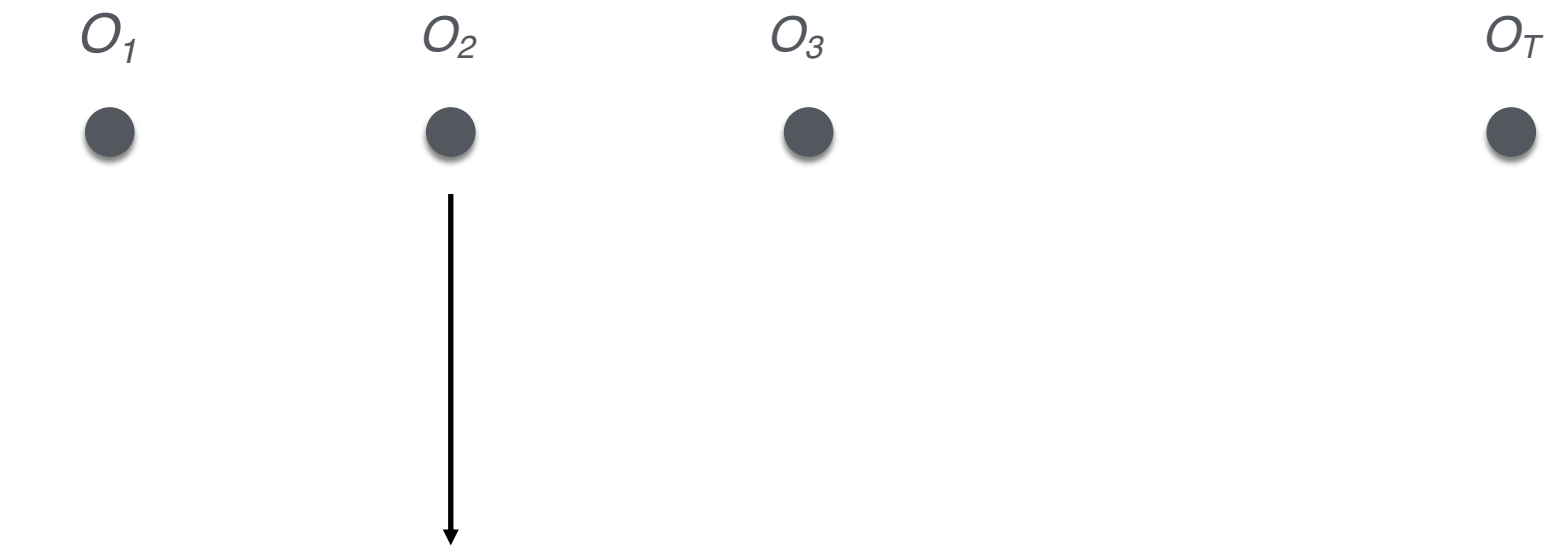
O_1 O_2 O_3 O_T



Predicted distribution

$$\left[\sum_{j=1}^{N} \alpha_1(j) a_{ji}\right]$$

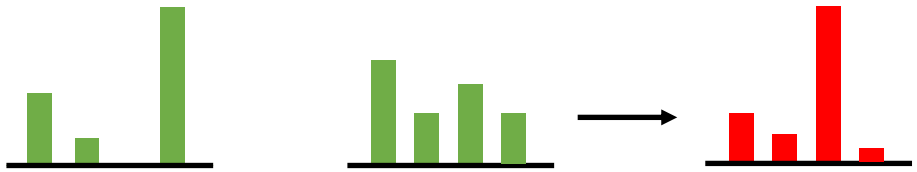
Forward algorithm intuition



"Improved" distribution

$$\left[\sum_{j=1}^{N} \alpha_1(j) a_{ji}\right] b_i(O_2)$$

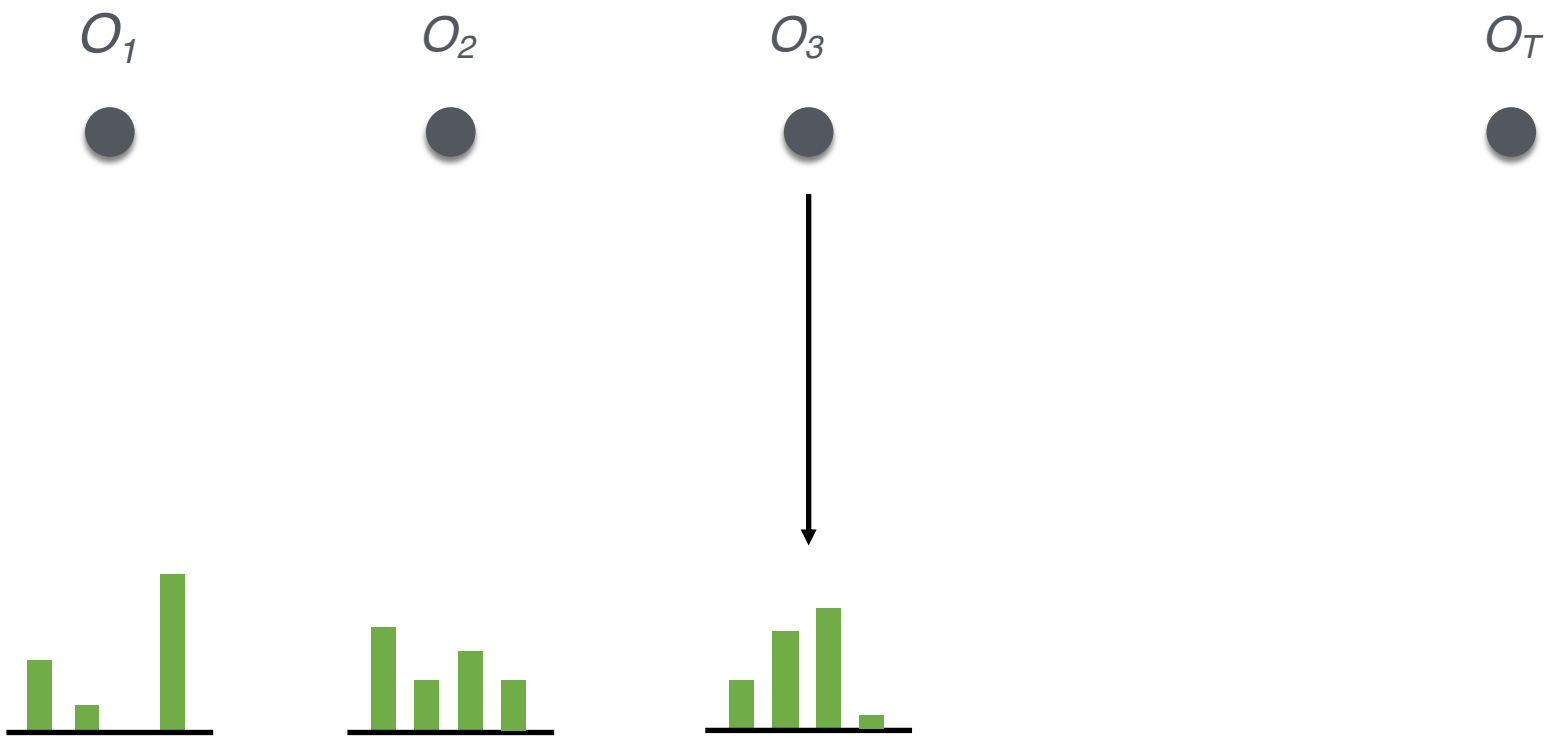
Forward algorithm intuition



Predicted distribution

$$\left[\sum_{j=1}^{N} \alpha_2(j) a_{ji}\right]$$

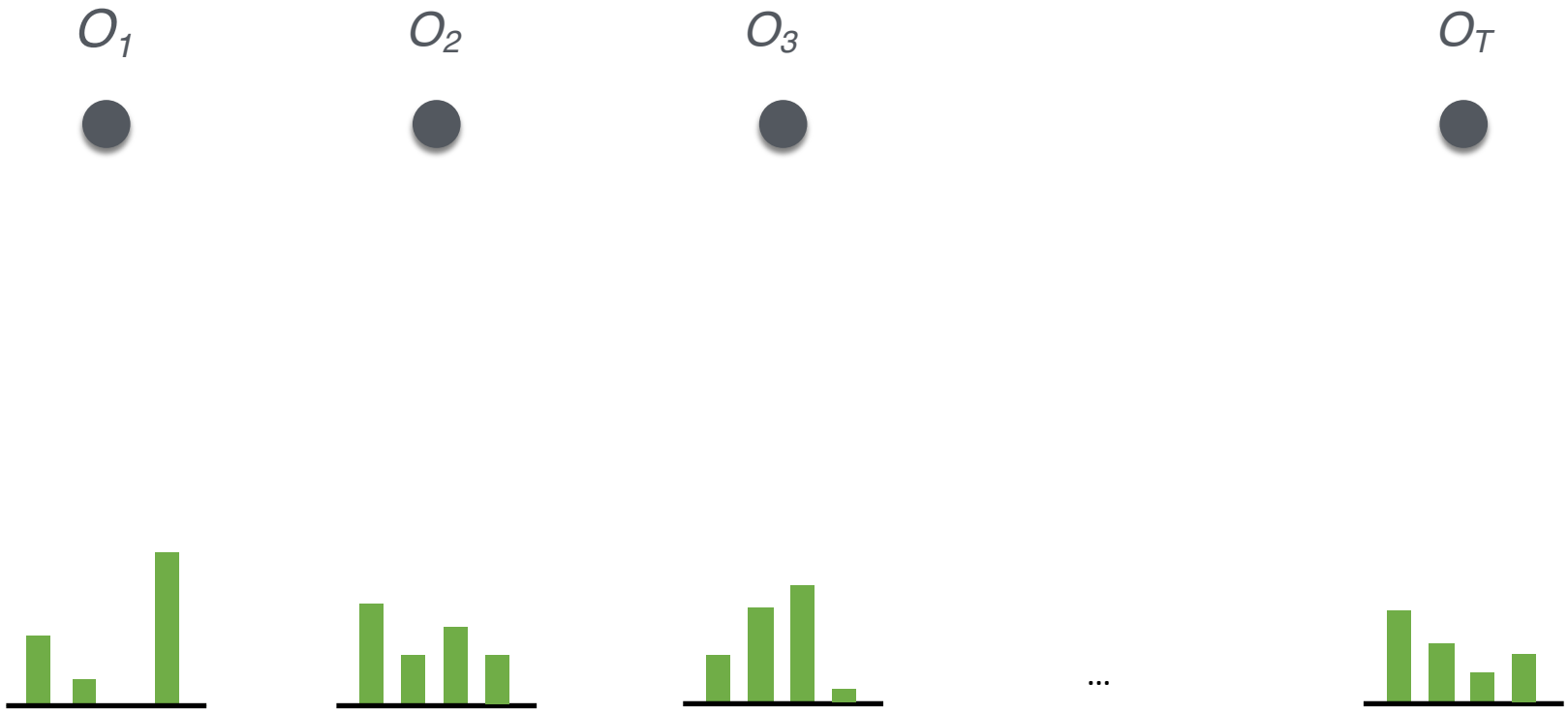
Forward algorithm intuition



"Improved" distribution

$$\left[\sum_{j=1}^{N} \alpha_2(j) a_{ji}\right] b_i(O_3)$$

Forward algorithm intuition



Backward algorithm (aka β-pass)

- $\beta_t(i)$ = Probability that the model is in the hidden state $X_t(i)$ (i *in* [1,2,…,N])

&& **will generate** the remainder of the emission sequence, from O_{t+1} to O_T, as specified by the emission sequence **O**.

- Introduce: $\beta_t(i) = p(O_{t+1:T} \mid X_t = i, \lambda)$
- Initialize: $\beta_T(i) = 1, \forall i = 1, \ldots, N$
- For t<T: $\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(O_{t+1})\beta_{t+1}(j)$ → How the present relates to the future

Backward algorithm (aka β-pass)

$$p(O_{1:T} \mid \lambda) = \sum_{i=1}^{N} p\,(O_{1:T}, X_1 = i) = \{productrule\}$$

$$= \sum_{i=1}^{N} p\,(X_1 = i)p(0_{1:T} \mid X_1 = i) = \{O_{1:T} = 0_1, O_{2:T}\}$$

$$= \sum_{i=1}^{N} p\,(X_1 = i)p\big(0_1, O_{2:T} \mid X_1 = i\big) = \{0_1 indep.\, of\, O_{2:T}\}$$

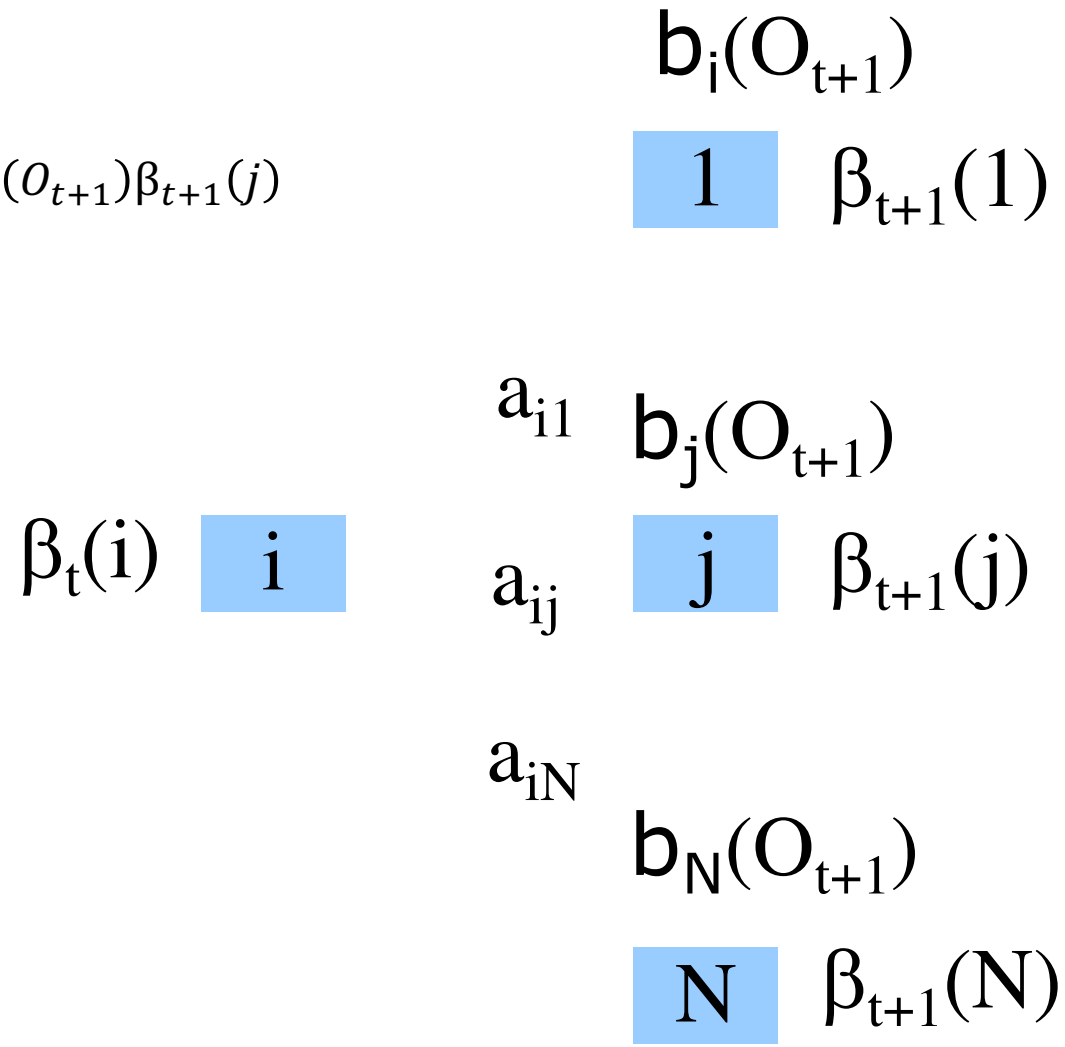
$$= \sum_{i=1}^{N} p\,(X_1 = i)p(0_1 \mid X_1 = i)p(O_{2:T} \mid X_1 = i)$$

$$= \sum_{i=1}^{N} \pi_i\, b_i(O_1)\beta_1(i)$$

- An alternative way to calculate $p(O_{1:T}|\lambda)$

One step in the backward algorithm

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(O_{t+1})\beta_{t+1}(j)$$



Three problems solved with HMMs

1. **Evaluation/**filtering: Compute likelihood $p(O_{1:t}|\lambda)$ of observation sequence ($O_{1:T}=\{O_1,O_2,\ldots,O_t,\ldots,O_T\}$) given λ

   Forward algorithm

2. **Decoding/**smoothing: Most likely state sequence $X^*_{1:T}$ given $O_{1:T}$ and λ

   Viterbi algorithm

3. **Learning**: Estimate model parameters $\Lambda=\{\lambda\}$ given $O_{1:T}$ such that $p(O_{1:T}|\lambda)$ is maximized
   $\rightarrow$ A and B matrices and π

   Baum-Welch algorithm

2. Calc most likely state sequence

- Motivating examples

–Parts-Of-Speech (POS) tagging:

- Given a sentence such as "**I love cats and dogs**"
- Find POS tags

<pronoun><verb><noun><conjunction><noun>

–Speech recognition

- Given a sound recording of spoken words
- Find which words were bring spoken

2. Calc most likely state sequence

- Motivating examples

–Parts-Of-Speech (POS) tagging:

- Given a sentence such as "**I love cats and dogs**"
- Find POS tags

<pronoun><verb><noun><conjunction><noun>

2. Calc most likely state sequence

- Motivating examples
  - Speech recognition
- Given a sound recording of spoken words
- Find which words were bring spoken

"Recognize speech"
"Wreck a nice beach"

sound very similar or identical!

- HMM will return the **most likely** sequence of words (hidden states)

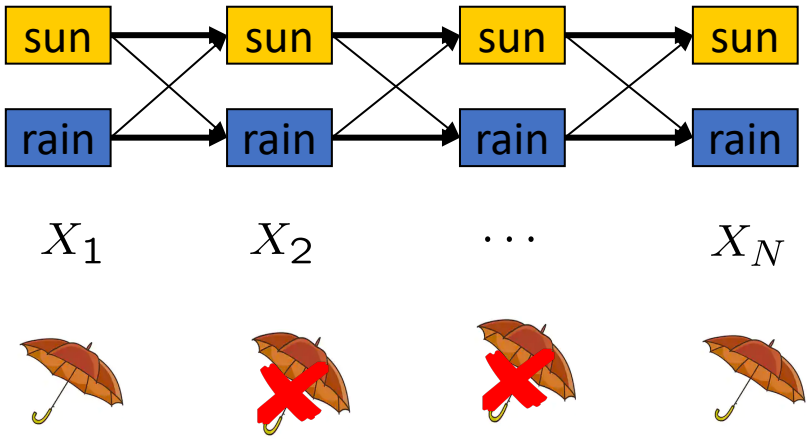
2. Calc most likely state sequence

Given:

- Emission sequence **O** = {O_1, O_2 … O_T}
- A, B, q

To Find:

- Hidden state sequence **X*** = {X_1, X_2 … X_T} that most likely produced **O**.
- Probability of occurrence of **X***



2. Calc most likely state sequence

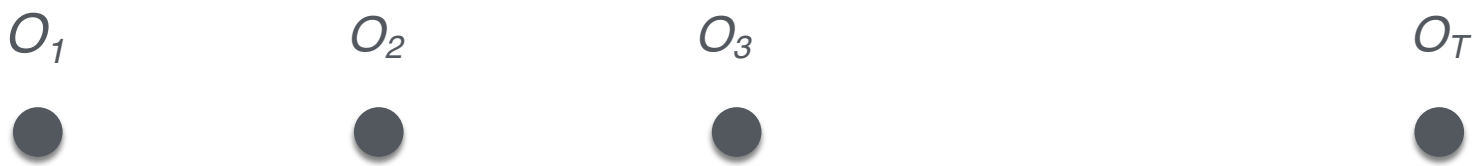
- We can find the most likely sequence by listing all possible sequences and finding the prob of the observed sequence for each of the combinations

$$X_{1:T}^{*} = \underset{X_{1:T}}{argmax}\ p(X_{1:T} \mid O_{1:T}, \lambda)$$

- Cannot solve individually for each time step

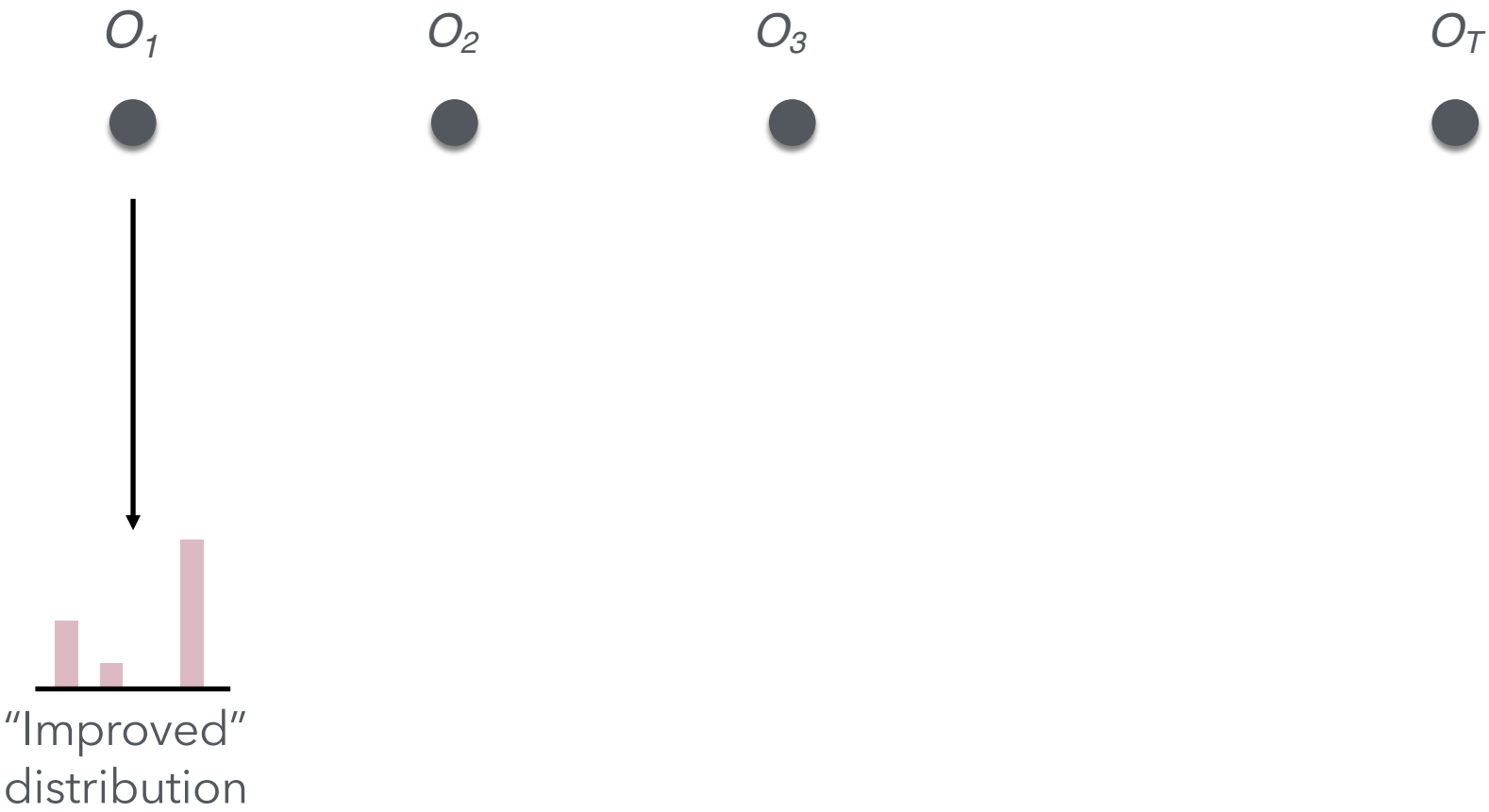
- Need to optimize the sequence not just individual states

2. Calc most likely state sequence

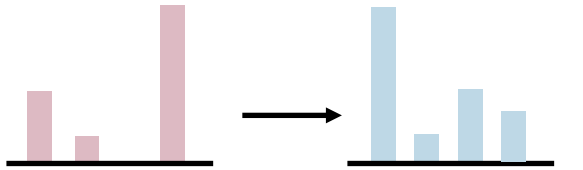


Initial
distribution

2. Calc most likely state sequence

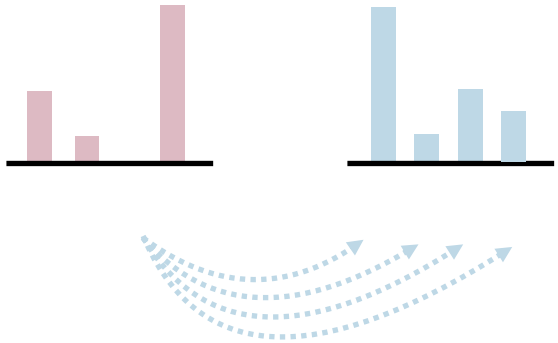


2. Calc most likely state sequence



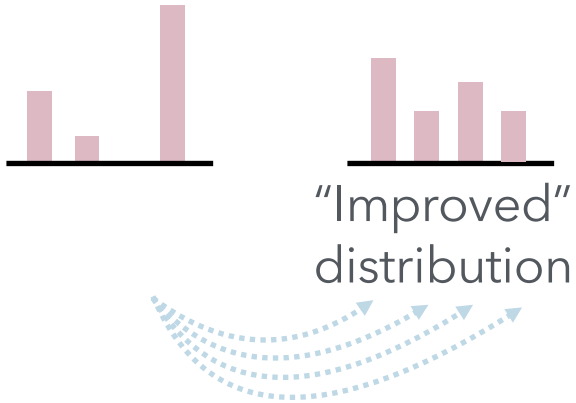
Predicted distribution

2. Calc most likely state sequence



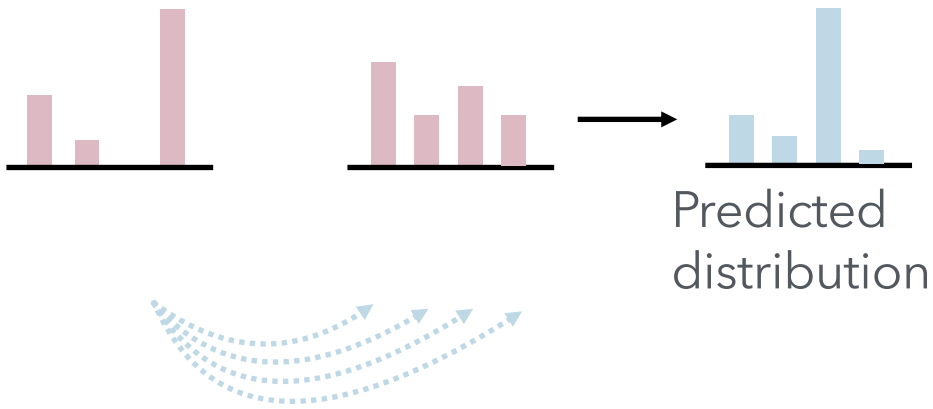
2. Calc most likely state sequence

O_1 O_2 O_3 O_T

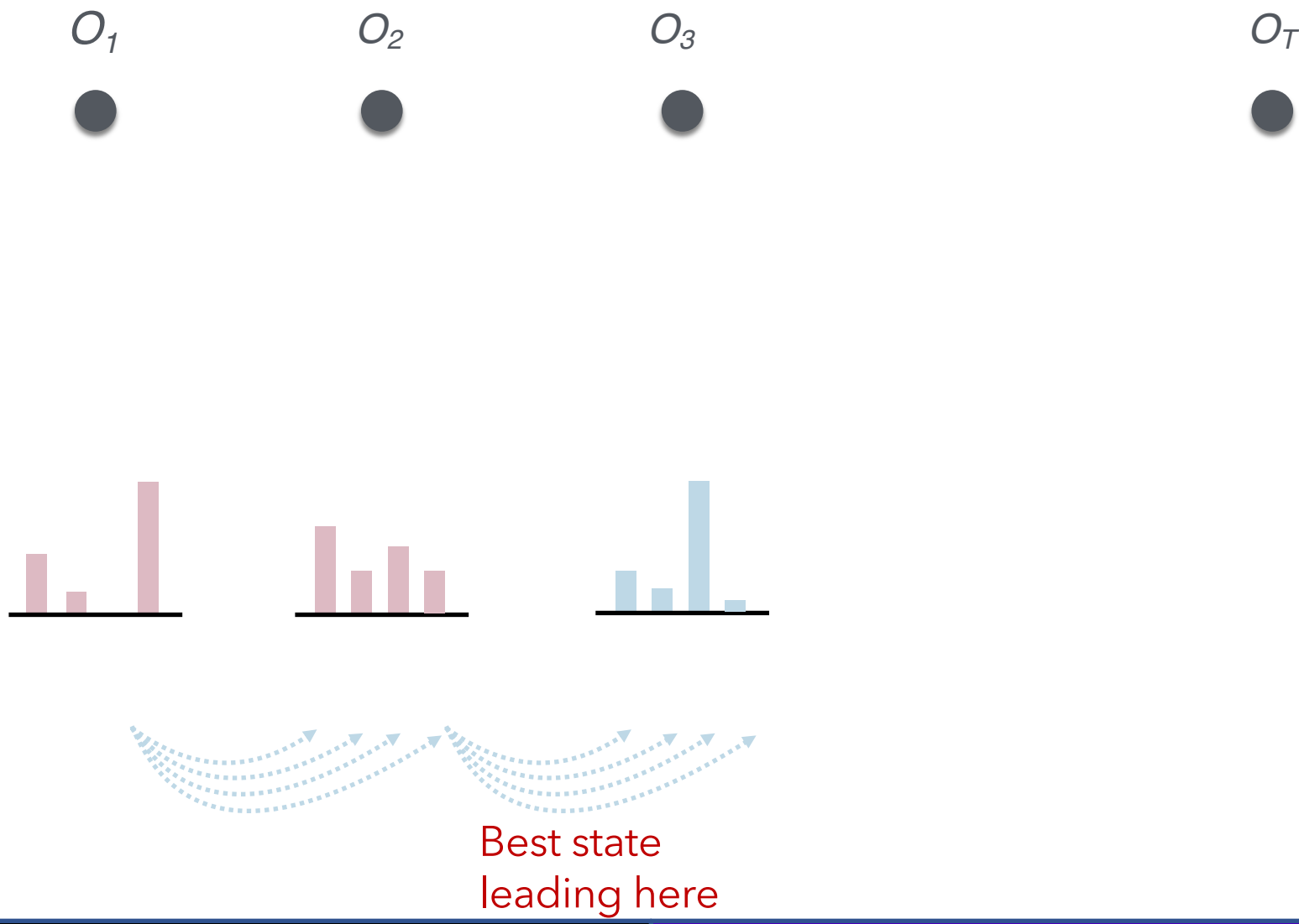


2. Calc most likely state sequence

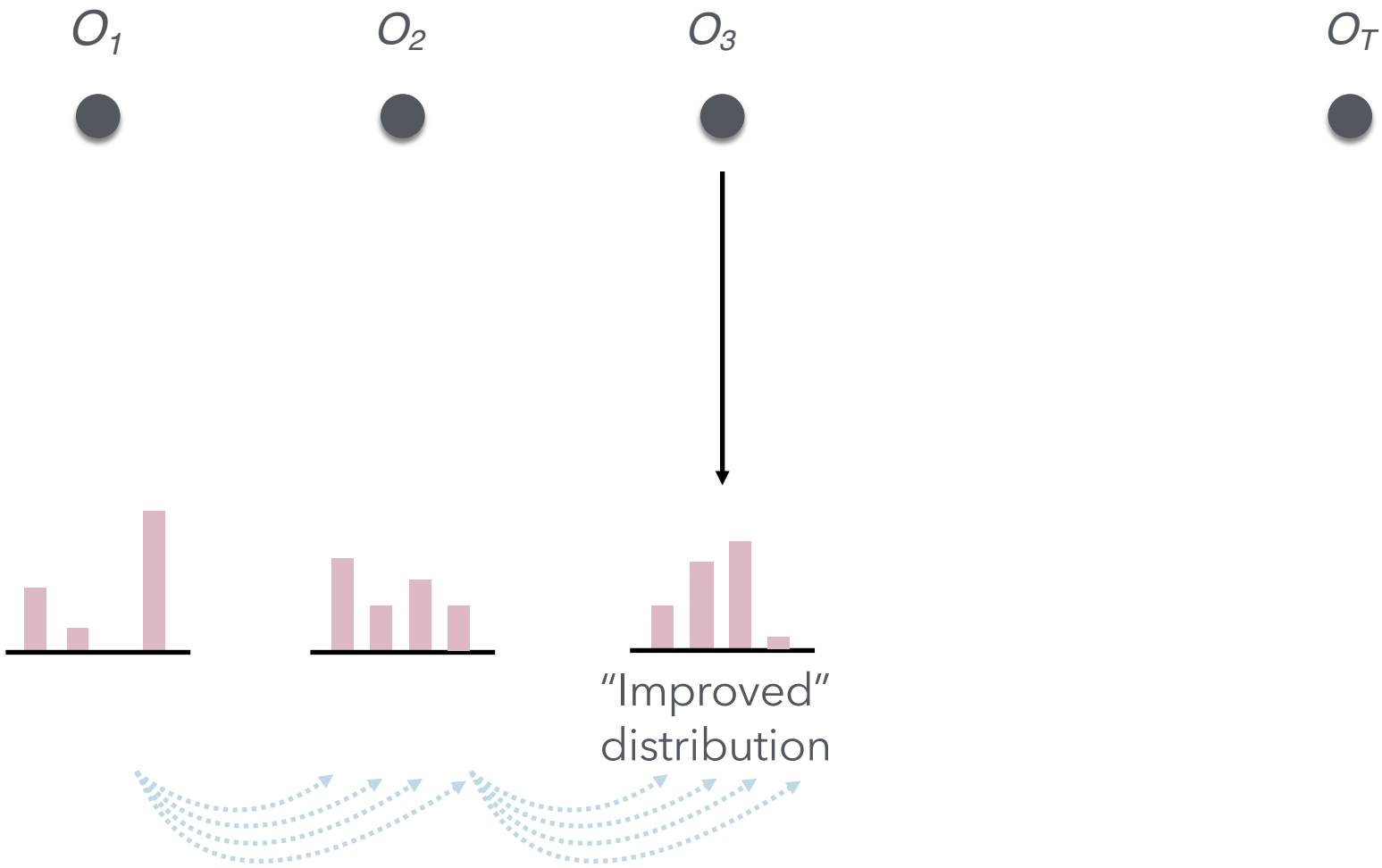
O_1 O_2 O_3 O_T



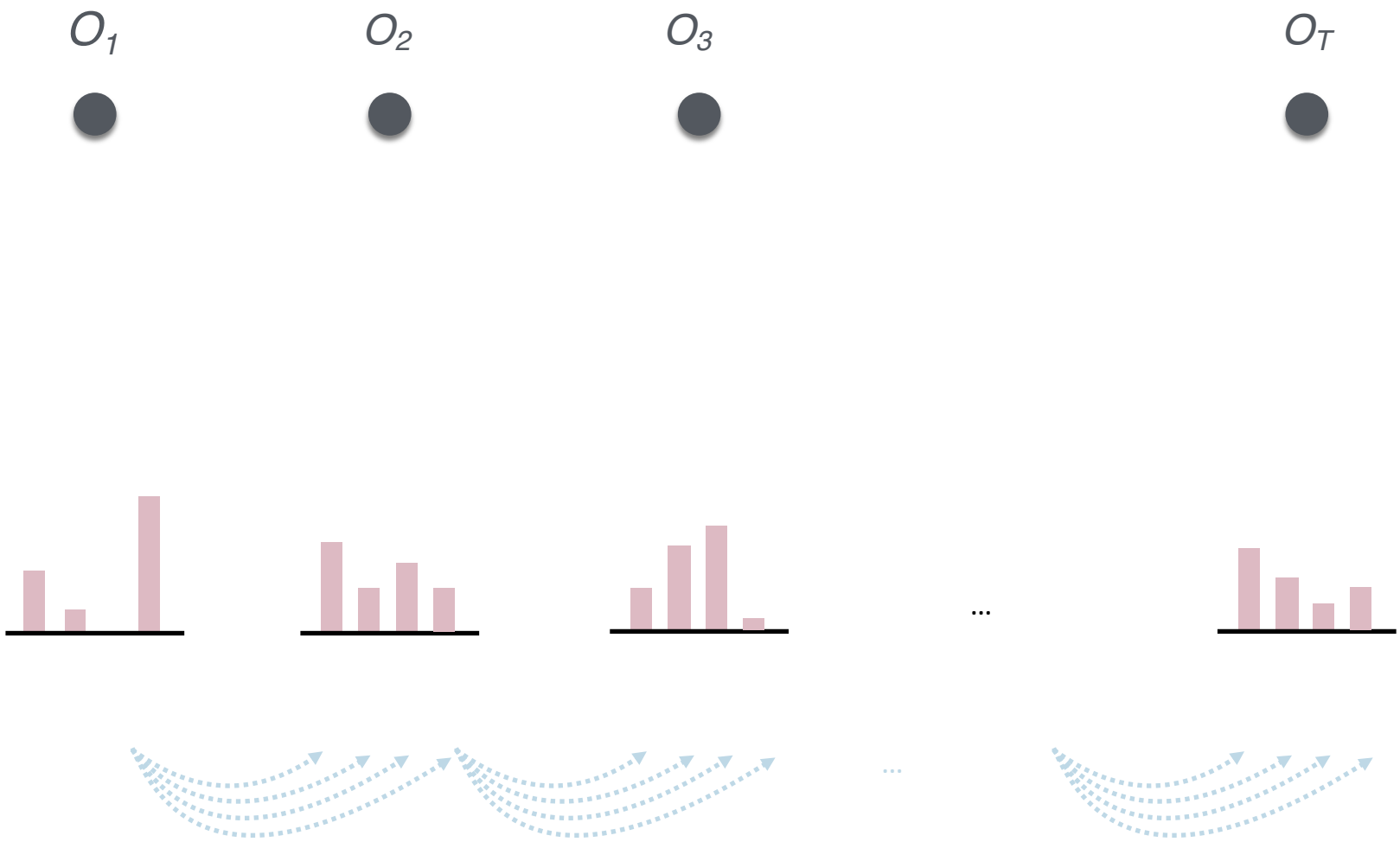
2. Calc most likely state sequence



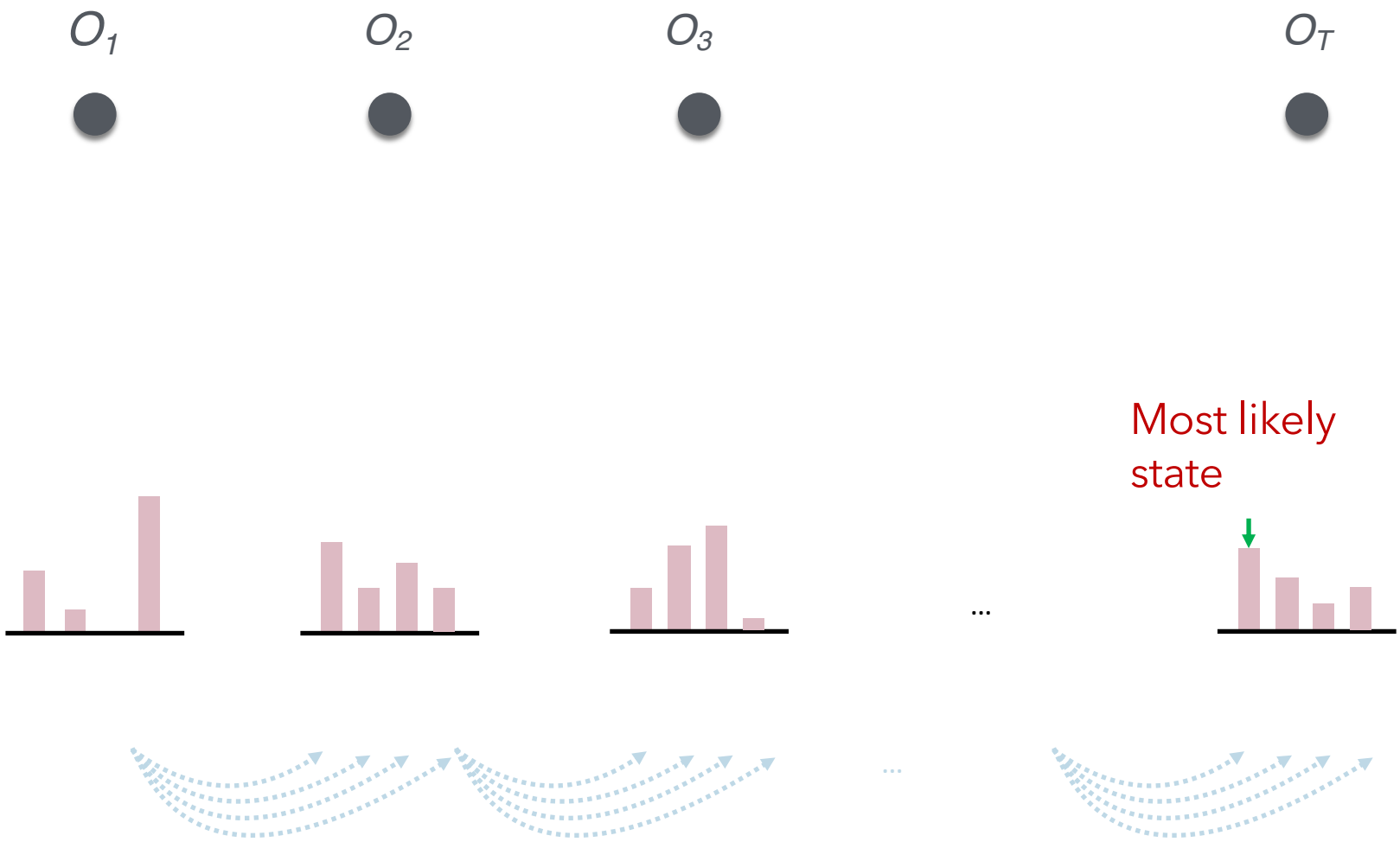
2. Calc most likely state sequence



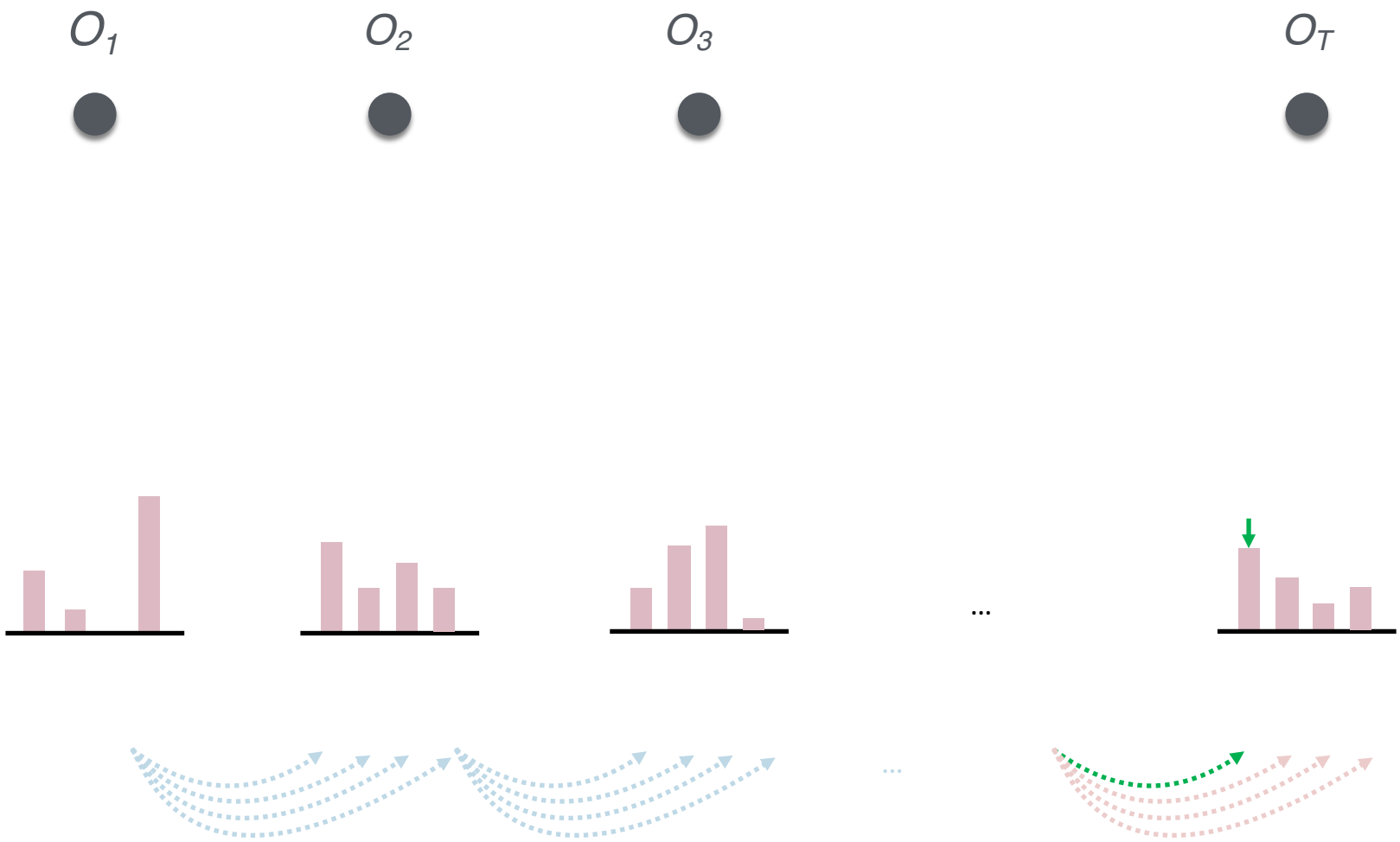
2. Calc most likely state sequence



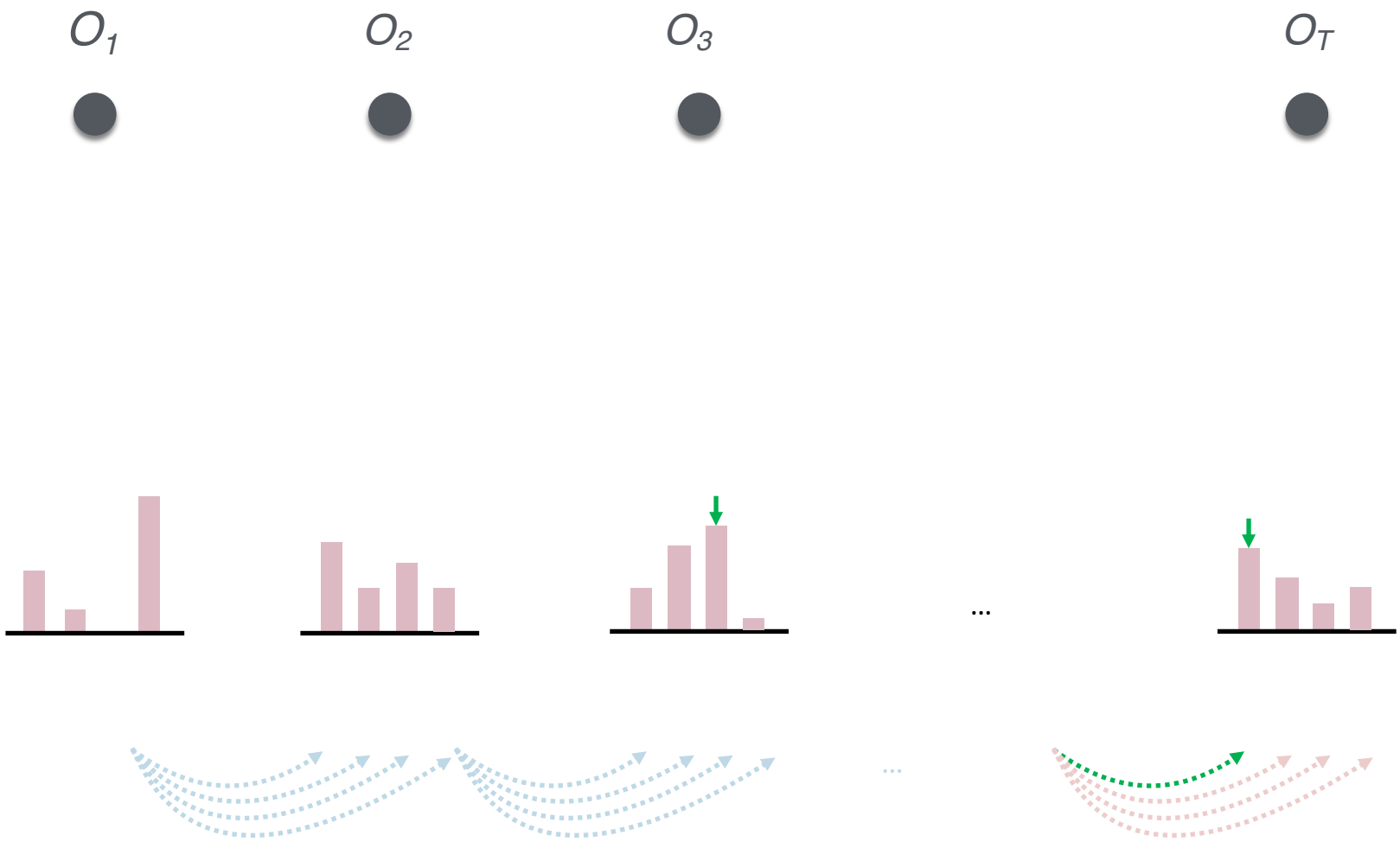
2. Calc most likely state sequence



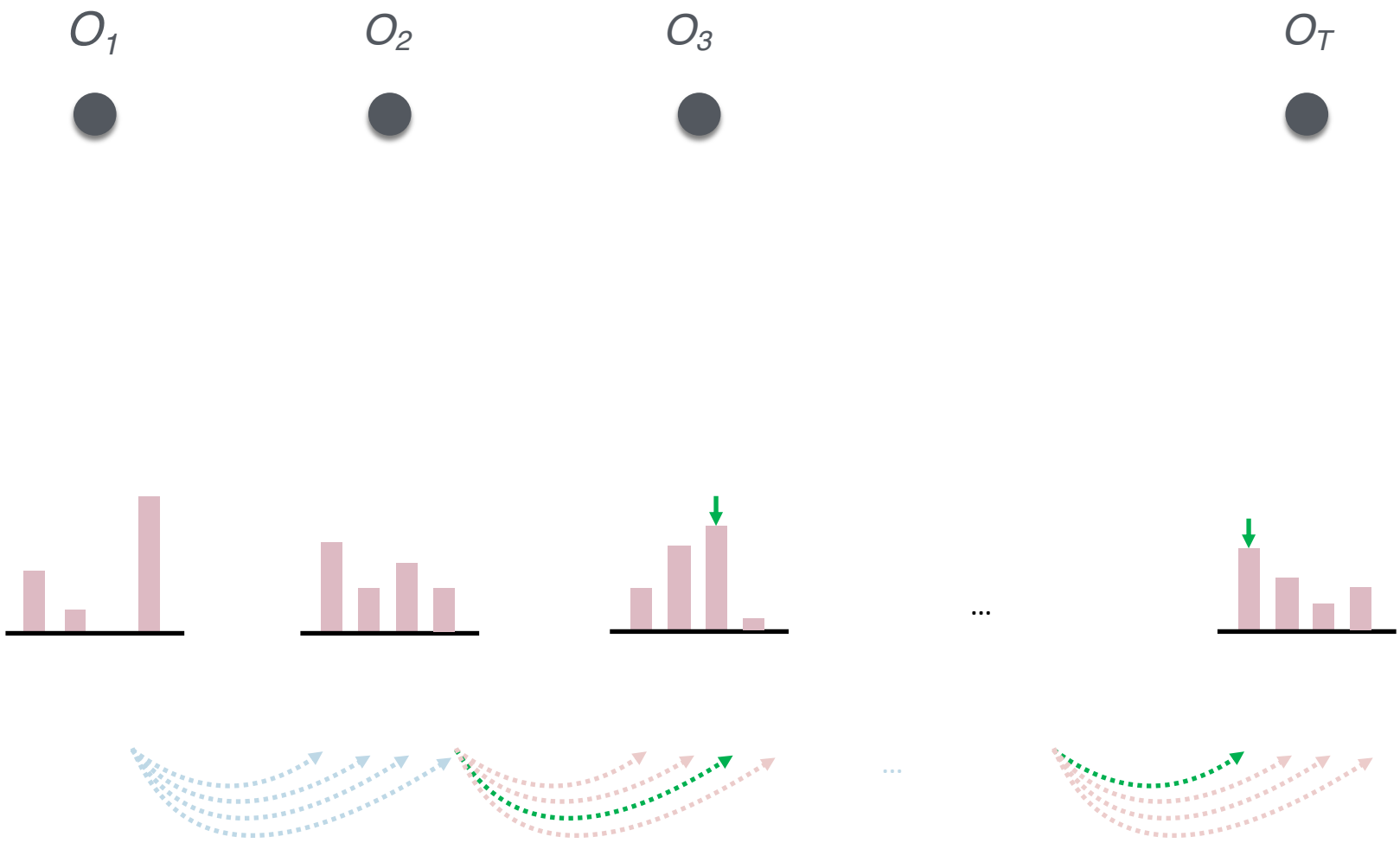
2. Calc most likely state sequence



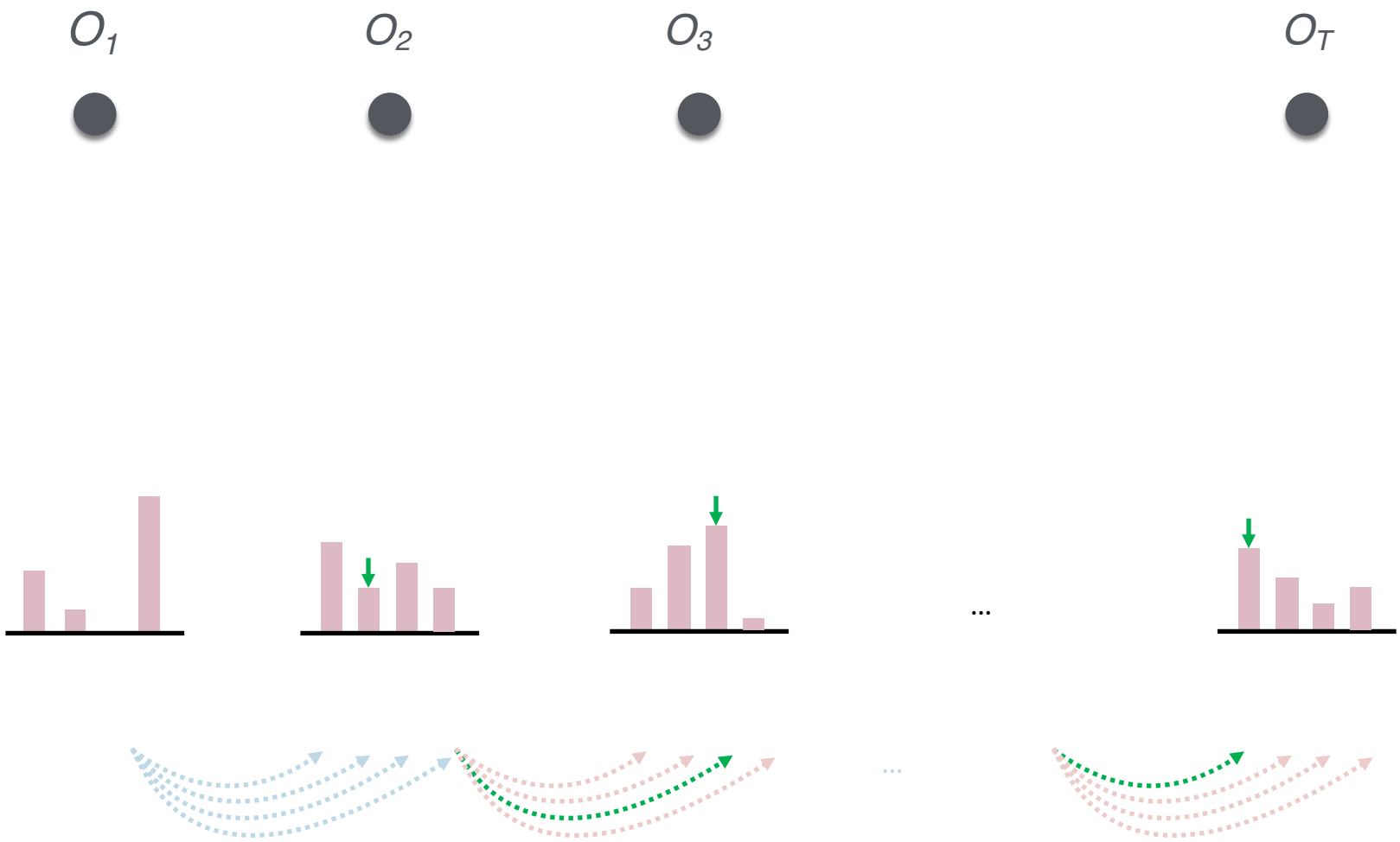
2. Calc most likely state sequence



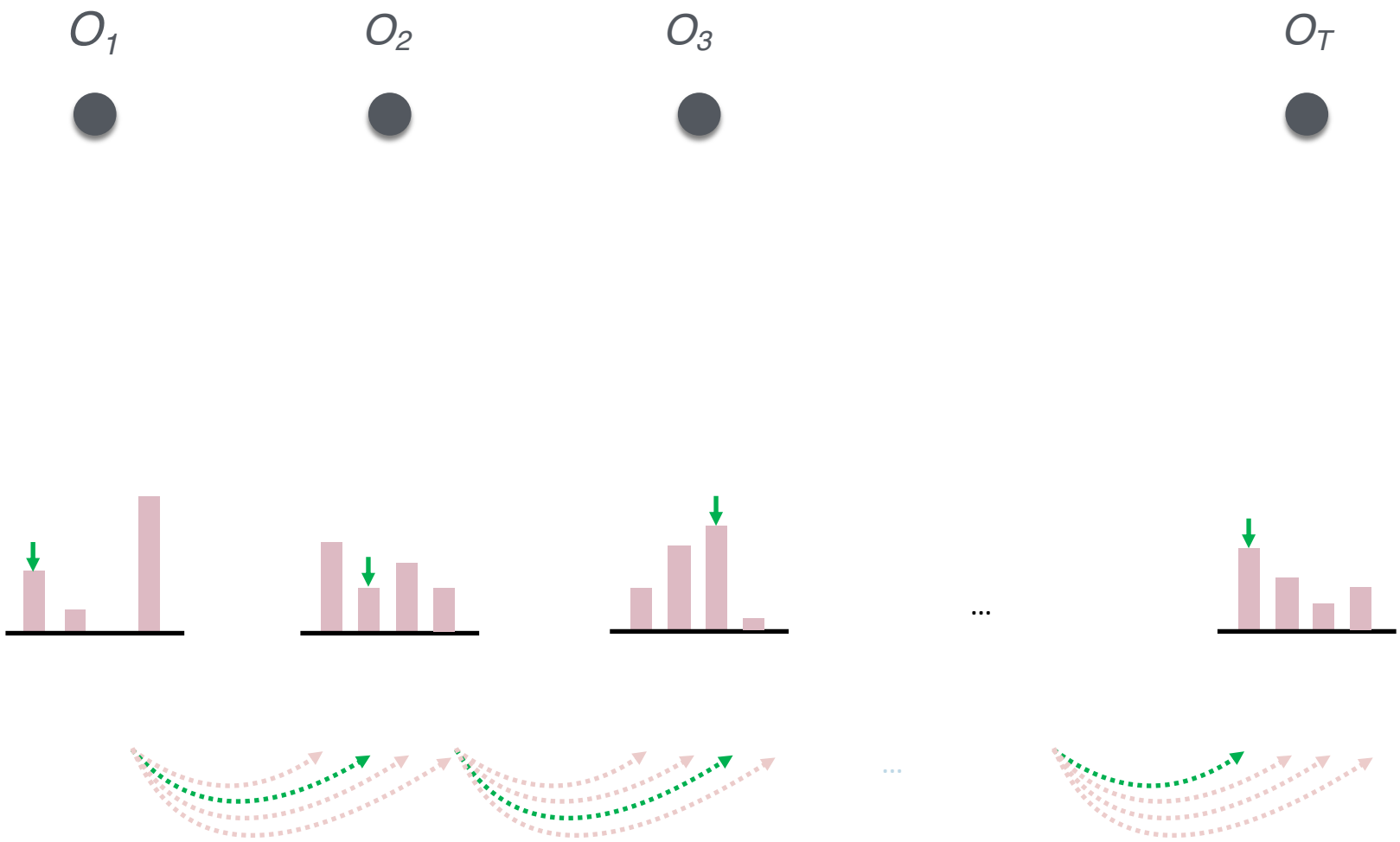
2. Calc most likely state sequence



2. Calc most likely state sequence



2. Calc most likely state sequence



Most likely sequence (Viterbi alg.)

- Solved using dynamic programming (DP) with an algorithm called Viterbi.

–Initialize: $\delta_1(i) = \pi_i b_i(O_1), i = 1, \ldots, N$

–For each t>1: $\delta_t(i) = \max_{j \in \{1,\ldots,N\}} [\delta_{t-1}(j) a_{ji} b_i(O_t)]$

Partial prob of one of the most probable paths to state *i* at time *t*

–Probability of best path: $\max_{j \in \{1,\ldots,N\}} [\delta_{T-1}(j)]$

–Find path by keeping book of preceding states and trace back from highest-scoring final state.

Forward vs. Viterbi algorithm

Forward algorithm computes sums of paths,
Viterbi computes best paths

$$\alpha_t(i) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}\right] b_i(O_t)$$

Forward algorithm (sum)

$$\delta_t(i) = \max_{j \in \{1,\ldots,N\}} \left[\delta_{t-1}(j) a_{ji} b_i(O_t)\right]$$

Viterbi algorithm (Max)

Viterbi cont'd

- Watch out for underflows!!! Multiplying many small numbers (probabilities)

- Better to use

–Initialize: $\delta_0(i) = \log[\pi_i b_i(O_0)], i = 1, \ldots, N$

–For t>1: $\widehat{\delta_t}(i) = \max_{j \in \{1,\ldots,N\}} \left[\widehat{\delta_{t-1}}(j) + \log[a_{ji}] + \log[b_i(O_t)]\right]$

–Probability of best path: $\max_{j \in \{1,\ldots,N\}} [\widehat{\delta_{T-1}}(j)]$

Three problems solved with HMMs

1. **Evaluation/**filtering: Compute likelihood $p(O_{1:t}|\lambda)$ of observation sequence ($O_{1:T}=\{O_1,O_2,\ldots,O_t,\ldots,O_T\}$) given λ

   Forward algorithm

2. **Decoding/**smoothing: Most likely state sequence $X^*_{1:T}$ given $O_{1:T}$ and λ

   Viterbi algorithm

3. **Learning**: Estimate model parameters $\Lambda=\{\lambda\}$ given a set of sequences $O_{1:T}$ such that $p(O_{1:T}|\lambda)$ is maximized
   $\rightarrow$ A and B matrices and π

   Baum-Welch algorithm

3. Estimate model parameters $\Lambda=\{\lambda\}$ given $O_{1:T}$

- Motivating examples:

–Learn a model for a letter for recognizing hand written text

–Learn a model for the word “Bayesian” from audio data

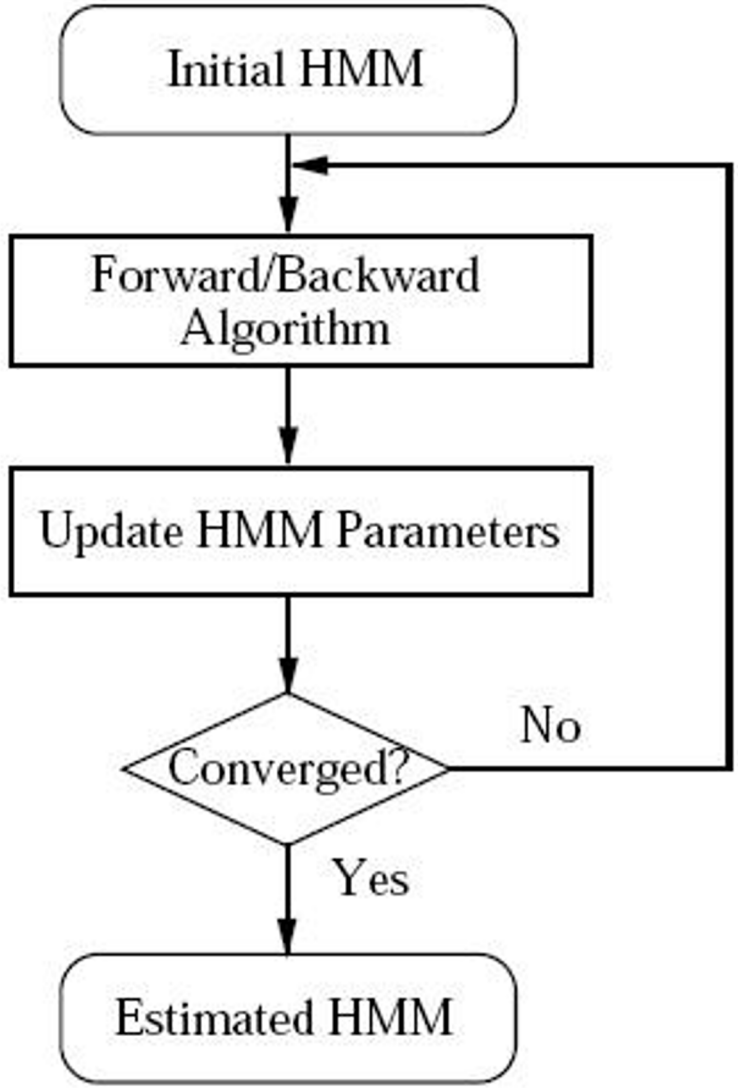
–A1: Learn a model for the movement of fish in the Fishing Derby Game

The Baum-Welch Algorithm

- Given an observation sequence $O_{1:T}$, the number of states, N, and the number of observation outcomes, M.

1. Initialize $\lambda=(A,B,\pi)$
2. Compute $\alpha_t(i)$, $\beta_t(k)$, $\gamma_t(i,j)$ and $\gamma_t(i)$
3. Re-estimate the model $\lambda=(A,B,\pi)$
4. Repeat from 2 until $p(O|\lambda)$ converges

Model estimation with Baum-Welch



GAMMA CALCULATIONS:

1) **Di – Gamma Function**

$$\gamma_t(i,j) = \frac{\alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_T(i)} = p(X_t = i, X_{t+1} = j \mid O_{1:T}, \lambda)$$

Interpretation: Given the entire observation sequence and current estimate of the HMM, what is the probability that at time (t) the hidden state is (X_t=i) && at time (t+1) the hidden state is (X_{t+1}=j)?

2) **Gamma Function** (Marginalizing out X_{t+1})

$$\gamma_t(i) = \sum_{j=1}^{N} \gamma_t(i,j) = p(X_t = i \mid O_{1:T}, \lambda)$$

Interpretation: Given the observation sequence and current estimate of the HMM, what is the probability that at time (t) the hidden state is (X_t=i)?

The function $\gamma_t(i)$

$$\gamma_t(i) = p(X_t = i \mid O_{1:T}, \lambda) = \{productrule\}$$

$$= \frac{p(X_t = i, O_{1:T} \mid \lambda)}{p(O_{1:T} \mid \lambda)} = \frac{p(X_t = i, O_{1:t}, O_{t+1:T} \mid \lambda)}{p(O_{1:T} \mid \lambda)}$$

$$= \{productrule\} = \frac{p(O_{1:t}, X_t = i \mid \lambda) p(O_{t+1:T} \mid X_t = i, O_{1:t}, \lambda)}{p(O_{1:T} \mid \lambda)}$$

$$= \{O_{t+1:T} independent of O_{1:t}\} = \frac{p(X_t = i, O_{1:t} \mid \lambda) p(O_{t+1:T} \mid X_t(i), \lambda)}{p(O_{1:T} \mid \lambda)}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{p(O_{1:T} \mid \lambda)} = \boxed{\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_T(i)}}$$

- Expresses the probability of being in state i at time t given the measurement sequence $O_{1:T}$

The di-gamma function, $\gamma_t(i,j)$

$$\gamma_t(i,j) = p(X_t = i, X_{t+1} = j \mid O_{1:T}, \lambda) = \{productrule\}$$

$$= \frac{p(X_t = i, X_{t+1} = j, O_{1:T} \mid \lambda)}{p(O_{1:T} \mid \lambda)} = \frac{p(X_t = i, X_{t+1} = j, O_{1:t}, O_{t+1:T} \mid \lambda)}{p(O_{1:T} \mid \lambda)}$$

$$= \{productrule\} = \frac{p(O_{1:t}, X_t = i, X_{t+1} = j \mid \lambda) p(O_{t+1:T}, X_t = i, X_{t+1} = j, O_{1:t}, \lambda)}{p(O_{1:T} \mid \lambda)}$$

$$= \{0_{t+1:T} cond.indep.of O_{1:t}\} = \frac{\alpha_t(i) a_{ij} p(O_{t+1:T} \mid X_t = i, X_{t+1} = j, \lambda)}{p(O_{1:T} \mid \lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} p(O_{t+1}, O_{t+2:T} \mid X_t = i, X_{t+1} = j, \lambda)}{p(O_{1:T} \mid \lambda)} = \{allOindep.\}$$

$$= \frac{\alpha_t(i) a_{ij} p(O_{t+1} \mid X_t = i, X_{t+1}, \lambda) p(O_{t+2:T} \mid X_t = i, X_{t+1}, \lambda)}{p(O_{1:T} \mid \lambda)}$$

$$= \{O_{t+1} indep.of X_t \wedge O_{t+2:T} cond.indep.of X_t given X_{t+1}\}$$

$$= \frac{\alpha_t(i) a_{ij} p(O_{t+1} \mid X_{t+1}, \lambda) p(O_{t+2:T} \mid X_{t+1}, \lambda)}{p(O_{1:T} \mid \lambda)} = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{p(O_{1:T} \mid \lambda)}$$

$$= \boxed{\frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_T(i)}}$$

Learn the model, i.e. calculating A,B,π

- Given γ and di-gamma, estimate A,B,π using:

A) Transition estimates

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \forall \; i,j = 1,\ldots,N$$

expected number of transitions from state i to state j

expected number of times in state i (regardless of what we observe)

B) Emission estimates

$$b_j(k) = \frac{\sum_{\substack{t=1,2,\ldots,T-1 \\ O_t=k}} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \forall \quad j = 1,\ldots,N, \quad k = 1,\ldots,K$$

expected number of times in state i and observing k

expected number of times in state i (regardless of what we observe)

q) Initial state probabilities

$$\pi_i = \gamma_1(i) \quad \forall \quad i = 1,\ldots,N$$

expected frequency of being in state I at t = 1

Interpretation of model estimation

- a_{ij} is calculated as the ratio of **all state transitions from i to j** and the **total number of times we are in state i**.

- $b_j(k)$ is calculated as the ratio of the **number of times observation k is given in state i** and the **total number of times we are in state i**.

- Note: there is a "version" of A and B at every time step.

We calculate an "average" over these versions to obtain an estimate of A and B for every iteration of Baum-Welch Algorithm.

To keep in mind

- Baum-Welch is an Expectation-Maximization (EM) algorithm used to train HMM parameters. It *uses* the forward-backward algorithm during each iteration.

- The forward-backward algorithm is just a combination of the forward and backward algorithms: one forward pass, one backward pass.

- On their own, the forward and backward algorithms are used for computing the marginal likelihoods of a sequence of states (not learning).

Model initialization

- Need to make sure that A,B,π are all row stochastic (rows sum to 1)
- Use whatever prior knowledge you have to provide good initial guesses
- If you have no clue, assign the values randomly as

$a_{ij} \approx 1/N$

$b_j(k) \approx 1/M$

$\pi_i \approx 1/N$

Implementation considerations

- Make sure that A,B and π are **not** uniform, i.e. that the values are not exactly 1/N and 1/M resp.
  - Otherwise we are in a local maximum that we cannot get out of and the method will not converge
  - If B is uniform, a measurement gives no info!
- Stop if too many iterations to avoid deadlocks
- Calculating long products of probabilities

→ very small numbers

→ underflow problems

→ use scaling and log likelihoods

Implementation considerations

- How much data is needed?

–Remember that we are estimating A and B by calculating statistics for how frequent transitions/emissions are.

- Little data → bad statistics → bad model

–No general rule, it depends on the problem.

Three problems solved with HMMs

1. **Evaluation/**filtering: Compute likelihood $p(O_{1:t}|\lambda)$ of observation sequence ($O_{1:T}=\{O_1,O_2,\ldots,O_t,\ldots,O_T\}$) given λ

   Forward algorithm

2. **Decoding/**smoothing: Most likely state sequence $X^*_{1:T}$ given $O_{1:T}$ and λ

   Viterbi algorithm

3. **Learning**: Estimate model parameters $\Lambda=\{\lambda\}$ given $O_{1:T}$ such that $p(O_{1:T}|\lambda)$ is maximized
   $\rightarrow$ A and B matrices and π

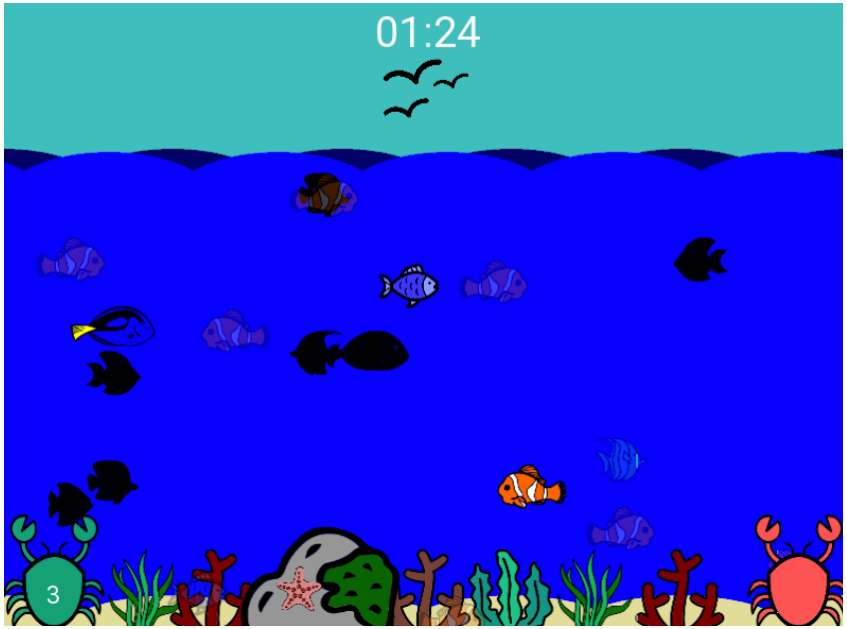
   Baum-Welch algorithm

What's NEXT?

- HMM Tutorials on campus and online – check the schedule on Canvas and do not forget to register.

- A1 is up!
(or will be very soon)

- Lab sessions to
start working on A1.



Motivation A1

- Solve an actual task, i.e., use the AI methods in a context
- Covers
  - Probabilistic reasoning
  - Machine learning
  - Decision making
  - Implementation
  - Testing and evaluation

A1 Assignment

- Work in pairs
- Practice using HMMs
- Have to implement your HMM

–Using an existing implementation not allowed

–No additional Python libraries other than Numpy, but Numpy will likely result in a very slow solution (Kattis uses PyPy to run Python code)