DD2380 Artificial Intelligence

Hidden Markov Models (HMMs)

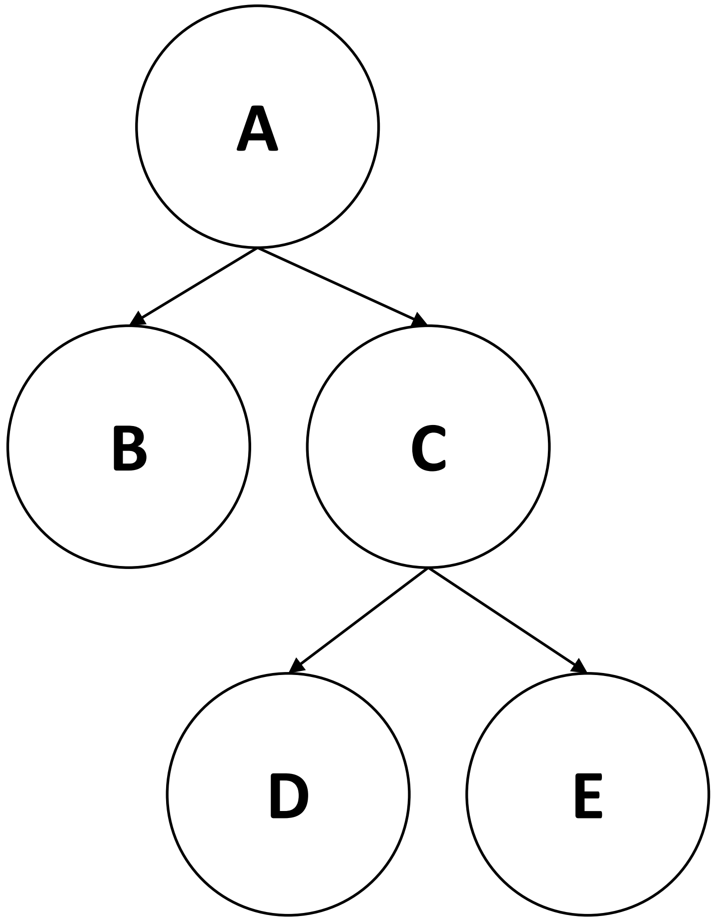
André Pereira (Starts at 15:15)

Reading Instructions

- Chapters 13-15, Russel & Norvig
- Stamp tutorial on HMMs on the course web page

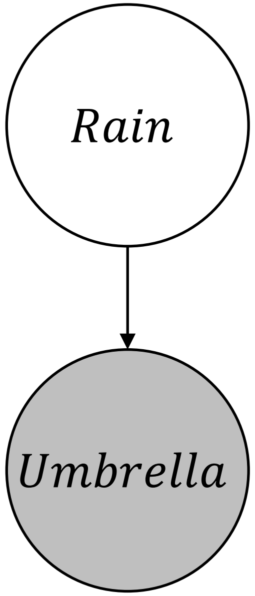
Bayesian Networks

- Aka Probabilistic **Directed Acyclic** Graphical Model
- Represents joint probability distribution (in a compact manner)
  - Helps with analyzing probability information
  - Helps with structuring probability information
- Arrow → "direct influence over"
  - A has direct influence over B
- Each arrow is accompanied with a conditional probability distribution, e.g., $P(B|A)$
- 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 $P(B|A)$
- Factorization
  - $P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{i=n} P(X_i|Parents(X_i))$

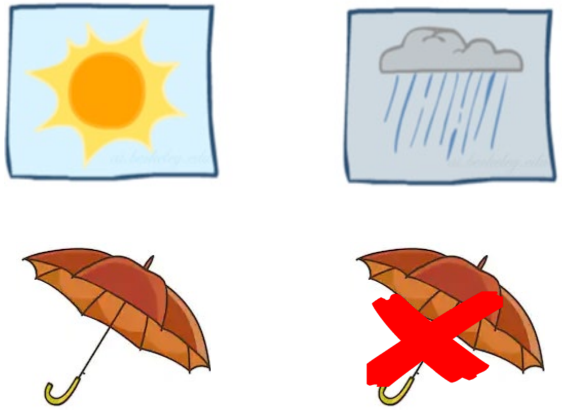


Example: Umbrella world

- A person tries to infer $rain = \{true, false\}$ by observing if a certain person has an $umbrella = \{true, false\}$ that day.
- Draw Bayesian network!
  - What are the variables?
  - Which cause which?



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

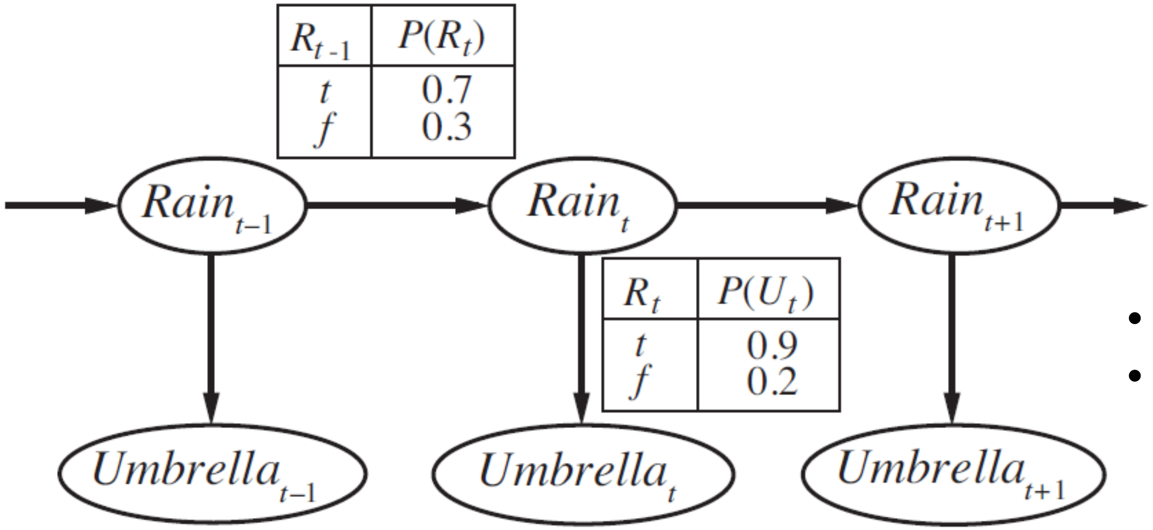


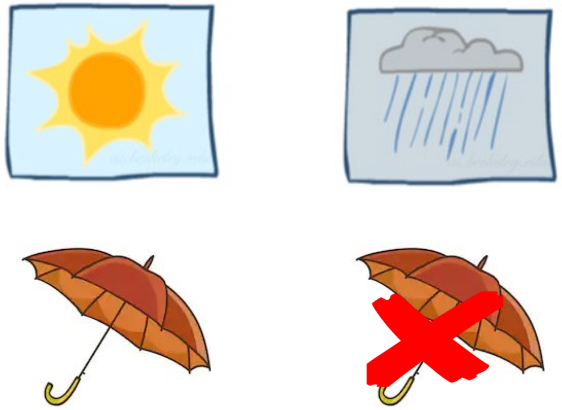
We often cannot directly observe the state!

- Other examples:
  - Cannot observe the weather, only the temperature
    - State = weather, observation = temperature
  - Cannot observe the words spoken, only the sound uttered
    - State = word, observation = sound
  - Cannot observe the position of the car only the laser scanner readings
    - State = position, observation = laser data

Example: Sequential Umbrella world

- A person tries to infer $rain = \{true, false\}$ **every day** by observing if a certain person has an $umbrella = \{true, false\}$ over consecutive days.
- Draw the Bayesian network that corresponds to the **time sequence**
  - What additional assumptions did you make?





- The observation depends directly on the state.
- Additional assumption
  - The next state depends only on the previous state (first order Markov assumption).

Markov Model

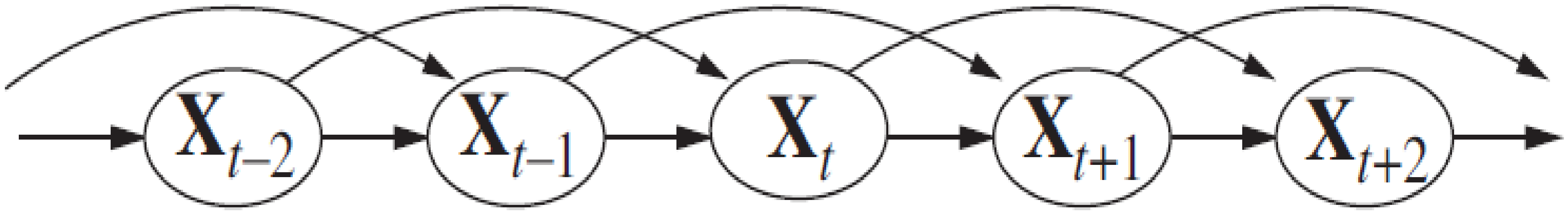
- The (first order) Markov assumption

- The distribution $P(X_t)$ depends only on the distribution $P(X_{t-1})$
  - $p(X_t|X_{t-1}, X_{t-2}, X_{t-3}, \dots) = p(X_t|X_{t-1})$

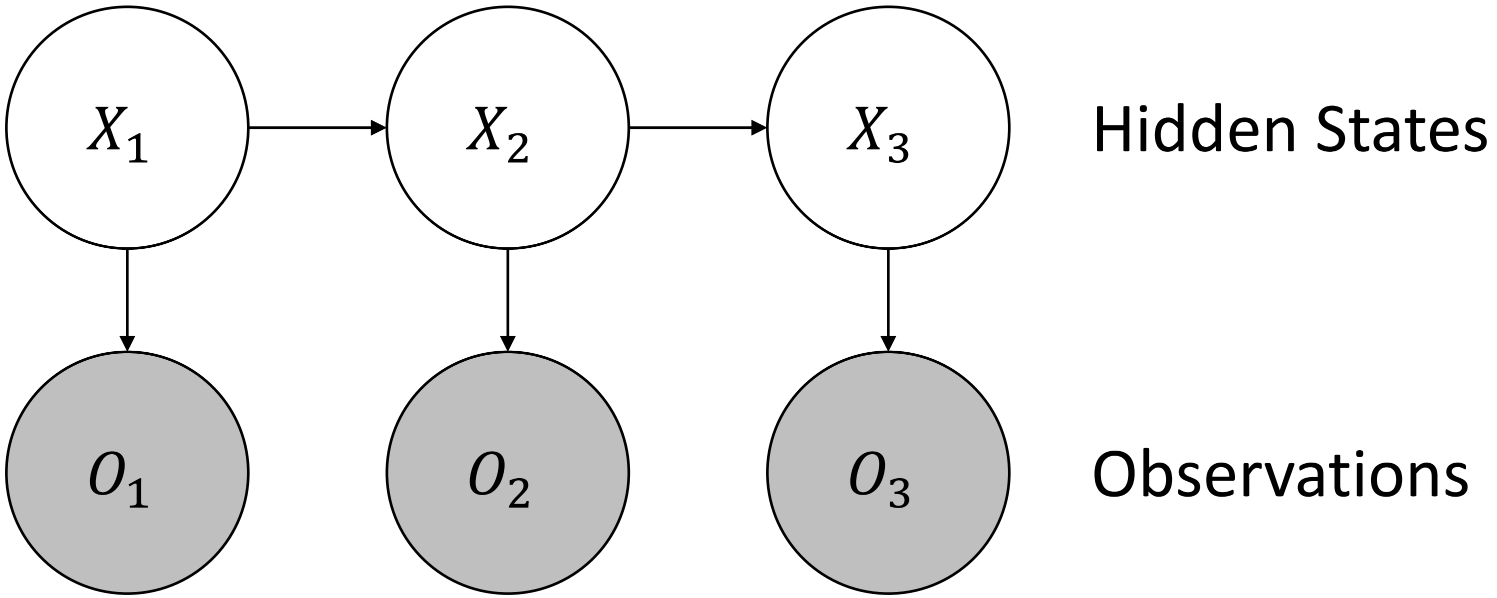
- The present (current state) can be predicted using local knowledge of the past (state at the previous step)

Second-order Markov Model

- State at time k depends on the states at times k-1 and k-2

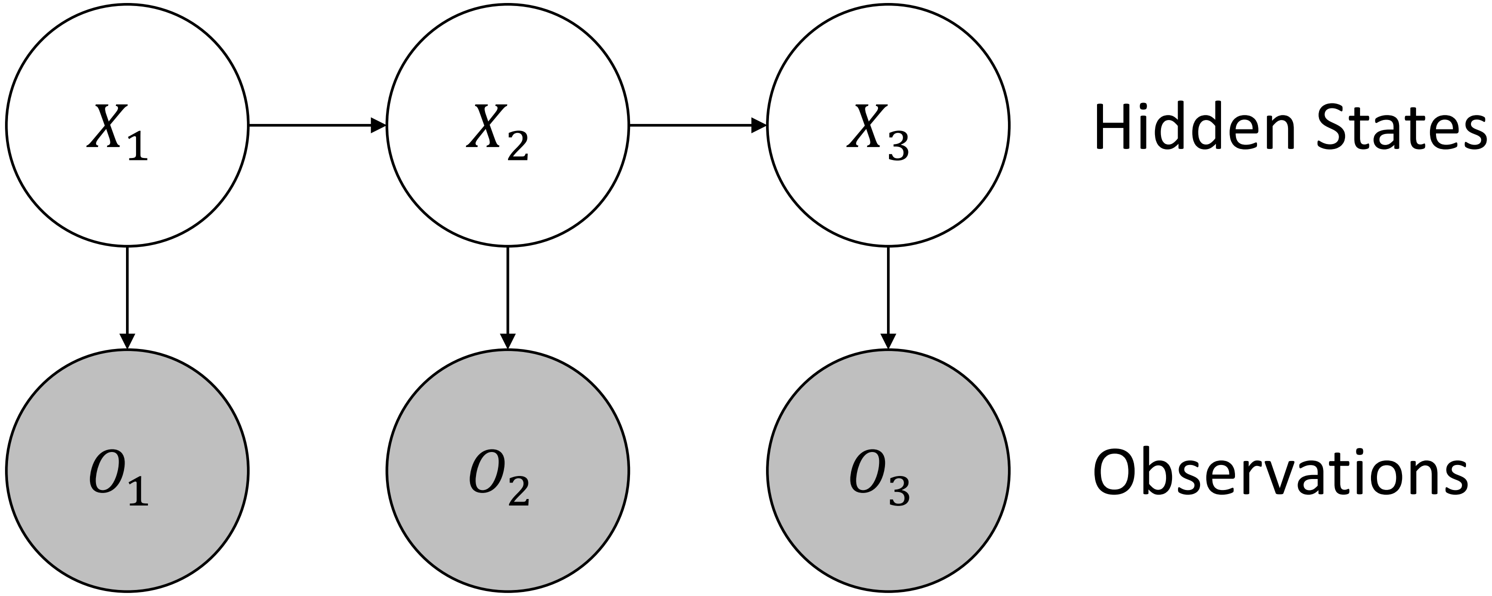


Hidden Markov Models (HMMs)



- Two important (conditional) independence properties:
  - Markov hidden process: future depends on the past via the present
  - Current observation independent of all else given current state

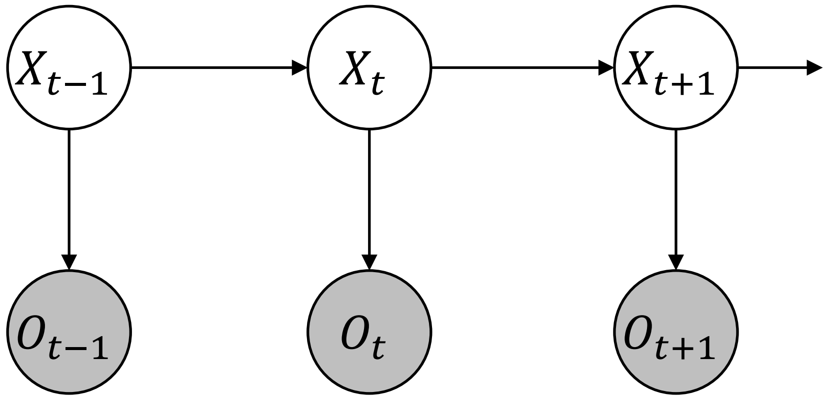
Hidden Markov Models (HMMs)

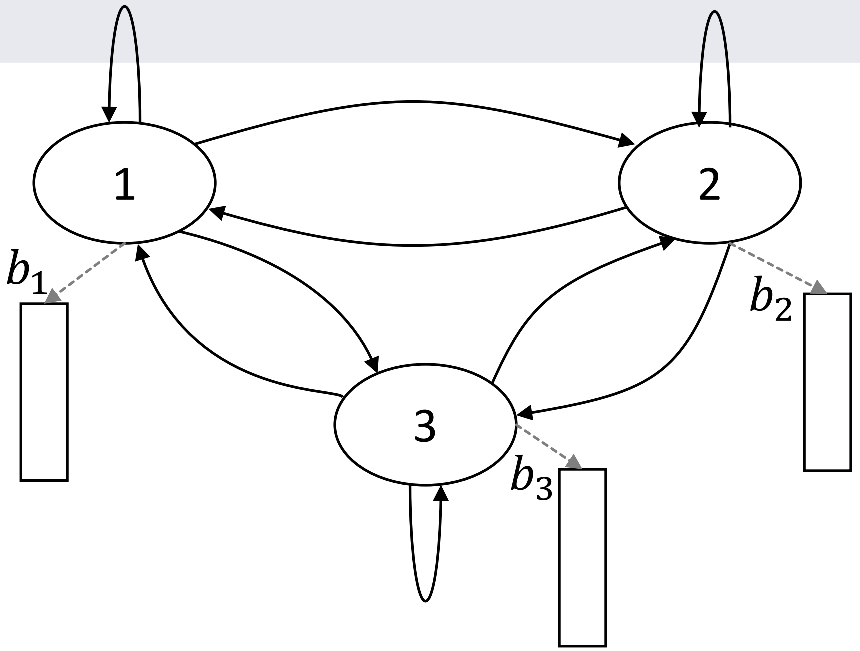


- State transition model: $\mathrm{P}(X_t = j | X_{t-1} = i) = A(i, j) = a_{ij}$
- Observation model: $\mathrm{P}(O_t = j | X_t = i) = b_{ij}$

BN vs State Machine Representation

	Bayesian Network Notation	State Machine Representation
Arrows	Dependency between variables	State transitions and observation probs
Circles	Variables (states at each time step)	The different states



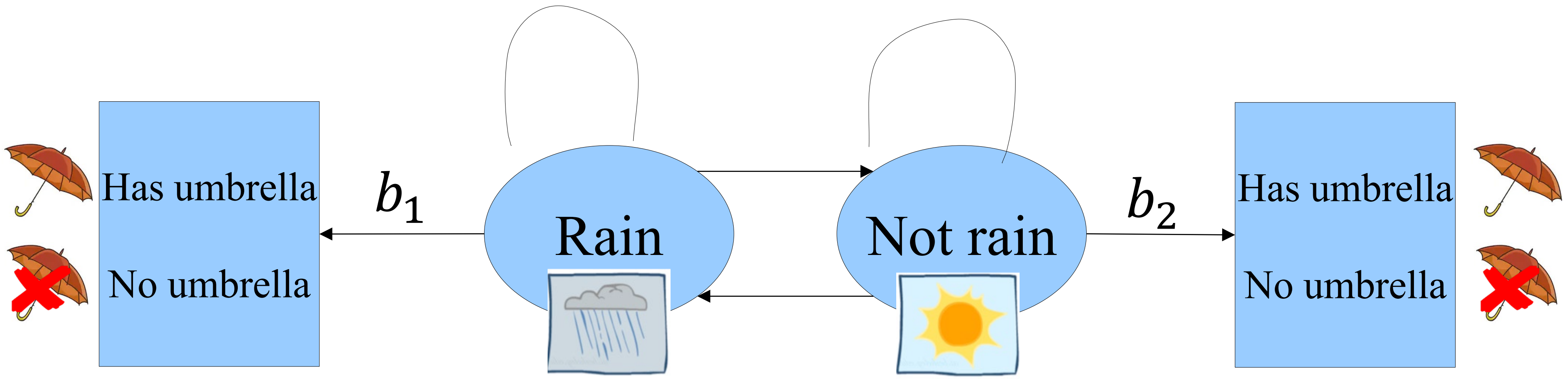


Example: Sequential Umbrella world

- A person tries to infer $rain = \{true, false\}$ **every day** by observing if a certain person has an $umbrella = \{true, false\}$ over consecutive days.
- Formulate using a State Machine Representation
  - How many states? Which?

    Rain = true, Rain = False
  - How many observations? Which?

    Umbrella = true, Umbrella = false



Elements of HMM

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

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

3. Initial State Probabilities
   $\pi = \{\pi_i\} = \{P(x_1 = i)\}$ for $1 \leq i \leq N$

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

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

Elements of HMMs, $\boldsymbol{\pi}$

- 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 $\boldsymbol{\pi}$

- Example:

$\pi = [0.5\ 0.2\ 0.3]$, i.e.,
$p(X_1 = 1) = 0.5$
$p(X_1 = 2) = 0.2$
$p(X_1 = 3) = 0.3$

Elements of HMMs, A

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

$$
\begin{bmatrix}
 & X_{t+1} = 1 & X_{t+1} = 2 & \dots & X_{t+1} = N \\
X_t = 1 & a_{11} & a_{12} & \dots & a_{1N} \\
X_t = 2 & a_{21} & a_{22} & \dots & a_{2N} \\
\dots & \dots & \dots & \dots & \dots \\
X_t = N & a_{N1} & a_{N2} & \dots & a_{NN}
\end{bmatrix}
$$

Elements of HMMs, B

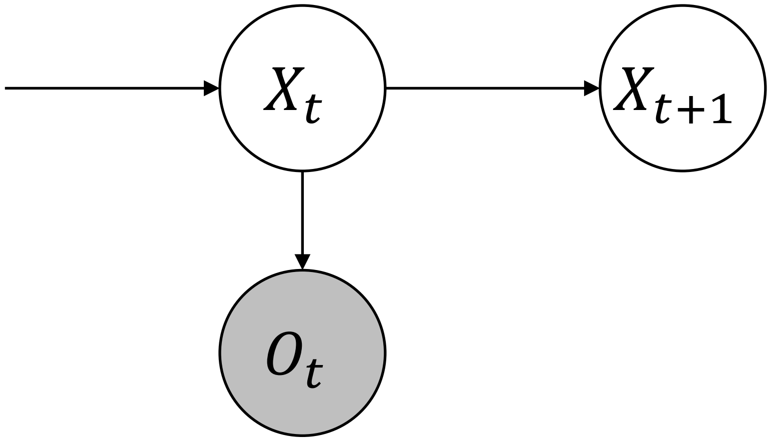
- Output matrix B
  - probability of observing a particular measurement given that the hidden model is in a particular hidden state.
- $b_i(O_t = j)$ is the probability to observe j in state i

$$
\begin{bmatrix}
 & O_t = 1 & O_t = 2 & \dots & O_t = K \\
X_t = 1 & b_1(1) & b_1(2) & \dots & b_1(K) \\
X_t = 2 & b_2(1) & b_2(2) & \dots & b_2(K) \\
\dots & \dots & \dots & \dots & \dots \\
X_t = N & b_N(1) & b_N(2) & \dots & b_N(K)
\end{bmatrix}
$$

The model - λ

- λ sometimes just called M, is the model, i.e.,
- $\lambda = (A, B, \pi)$
  - A: state transition matrix
  - B: observation matrix, output matrix, emission probabilities, emissions
  - π: initial state distribution
- A, B and π are row-stochastic matrices (their rows sum to 1)

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:

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

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

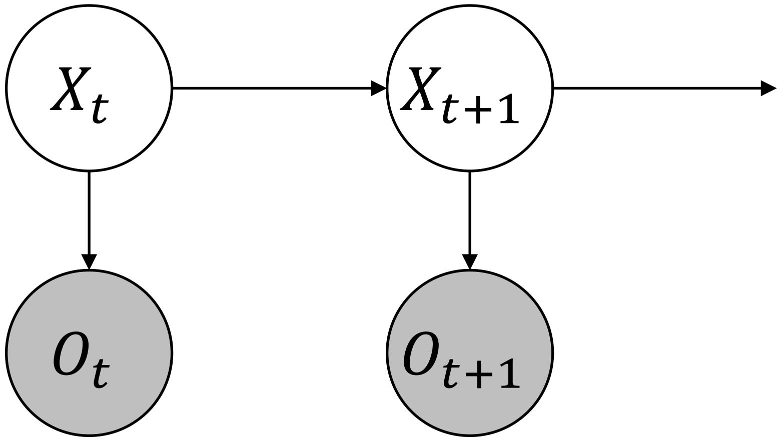
$$= \{product\ rule\} = \sum_{X_t} p\,(X_{t+1} \mid X_t\,, O_{1:t})\ p(X_t \mid O_{1:t})$$

$$P(A,B) = P(A \mid B)P(B)$$
$$P(A,B \mid C) = P(A \mid B,C)P(B|C)$$

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

- Basic idea: beliefs get “pushed” through the transitions

Measurements/observations



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

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

- Then, after evidence comes in:

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

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

$$P(B|A) = \frac{P(A|B)P(B)}{\sum_B P(A|B)P(B)}$$

$$P(B|A,C) = \frac{P(A|B,C)P(B|C)}{\sum_B P(A|B,C)P(B|C)}$$

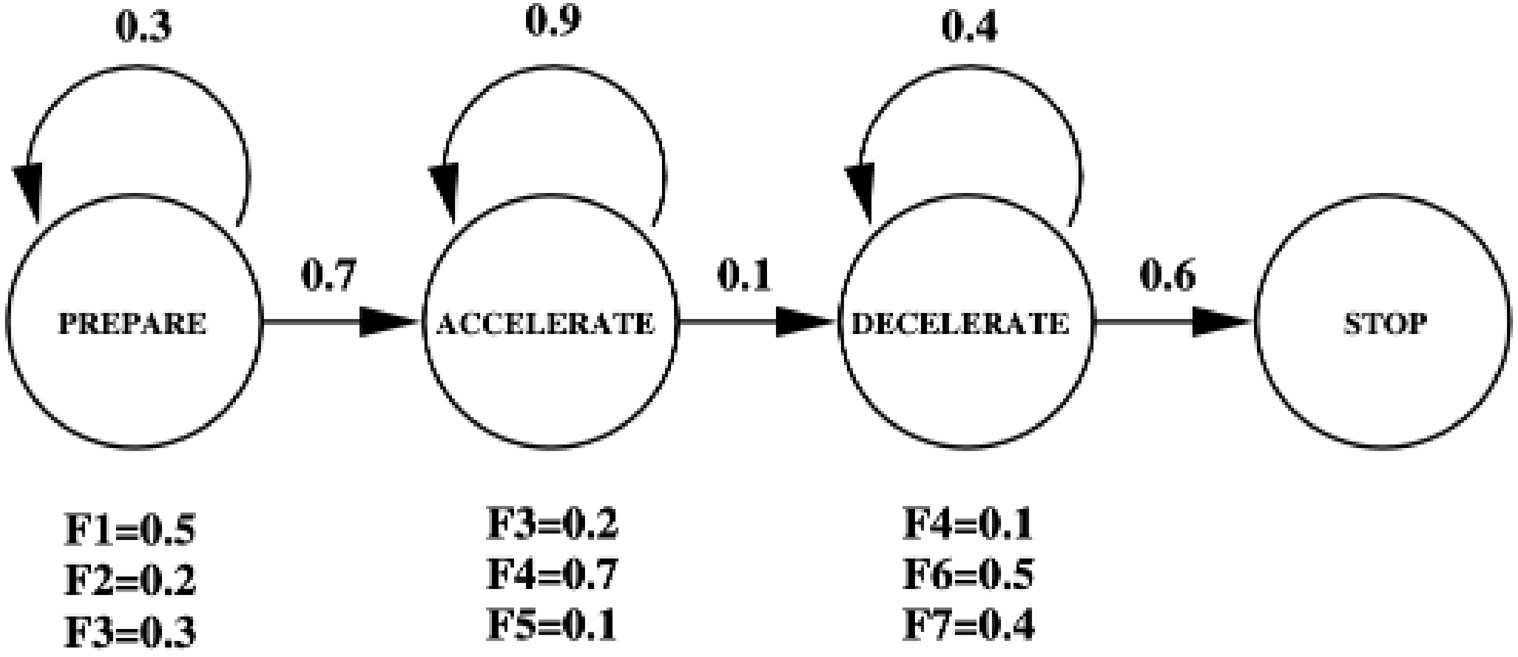
$$= \{\ O_{t+1} \text{ cond. } indep \text{ of } O_{1:t}\ given\ X_{t+1}\} = \frac{p(O_{t+1} \mid X_{t+1})\, p(X_{t+1} \mid O_{1:t})}{\sum_{X_{t+1}} p\,(O_{t+1} \mid X_{t+1}\,)\, p(X_{t+1} \mid O_{1:t})}$$

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

Example: Usain Bolt

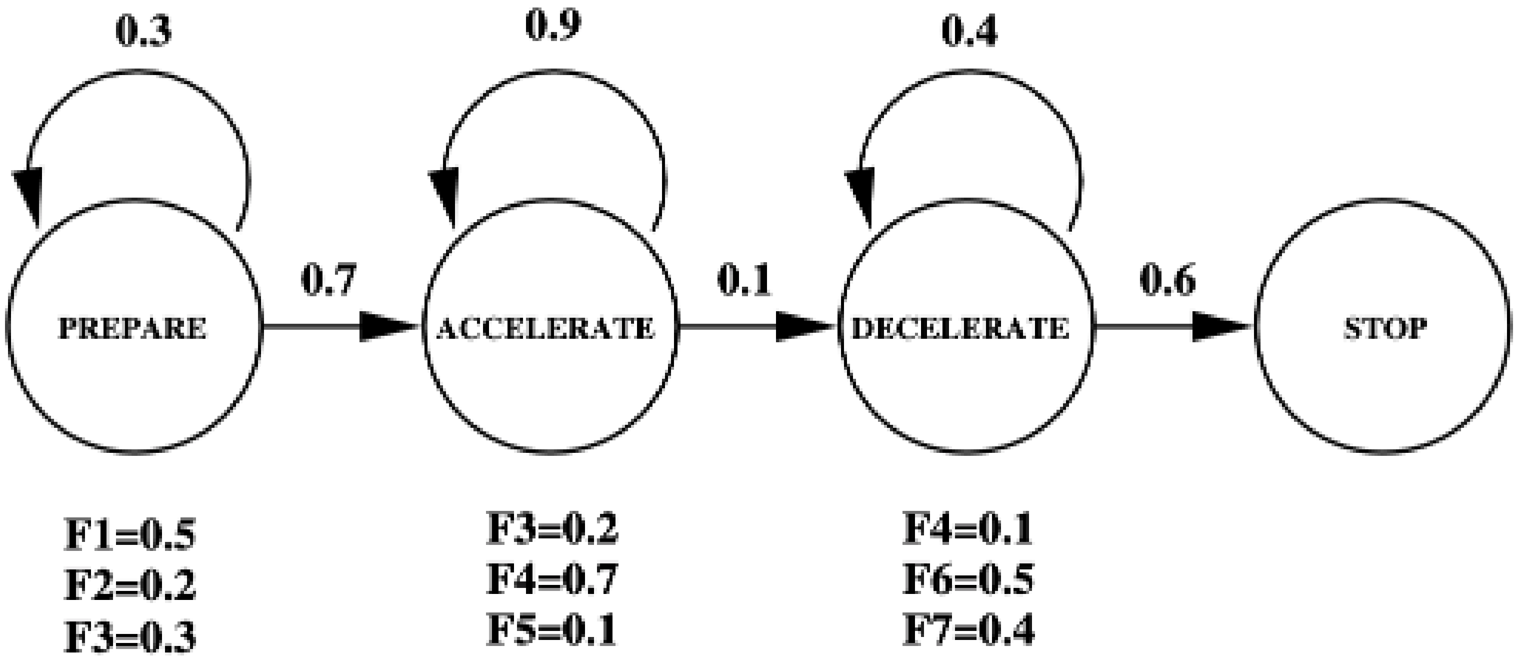
- F1 to 7 represent 7 ways of classifying frames
  - 7 possible observation outcomes that depict Usain's posture and probabilities of observing in each state
  - Observations not mentioned under a state have probability 0 of being observed there.
- STOP is a non-emitting terminating state.
- Video sequences are divided in several video frames classified as F1 to F7.



Example: Usain Bolt

- What does $A, B, and\ \pi$ look like?

$$A = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.2 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.7 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0.5 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \pi = [1 \quad 0 \quad 0 \quad 0]$$



Example: Usain Bolt

- Given:

$$A = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.2 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.7 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0.5 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \pi = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

- How to calculate $\mathrm{P}(X_2 = ACCELERATE)$?
  - Start with π and propagate probability one step with A
    - $\pi = X_1 = [1\ 0\ 0\ 0] \ \rightarrow \ X_2 = \pi A = [0.3\ 0.7\ 0\ 0]$
    - $p(X_2 = ACCELERATE) \ = \ 0.7$

Example: Usain Bolt

- Given:

$$A = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{array}{c} \begin{array}{ccccccc} & F_2 & & & & & \end{array} \\ \begin{bmatrix} 0.5 & \boxed{0.2} & 0.3 & 0 & 0 & 0 & 0 \\ 0 & \boxed{0} & 0.2 & 0.7 & 0.1 & 0 & 0 \\ 0 & \boxed{0} & 0 & 0.1 & 0 & 0.5 & 0.4 \\ 0 & \boxed{0} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad \pi = [1 \quad 0 \quad 0 \quad 0]$$

- How to calculate $\mathrm{P}(O_2 = F2)$?
  - **We know** $\mathbf{P(X_2) = [0.3\ 0.7\ 0\ 0]}$
  - And we know how probable each measurement is in each state from B. Use sum rule.

$$p(0_2 = F_2) = \sum_{i=1}^{N} p\,(O_2 = F_2 \mid X_2 = i)p(X_2 = i)$$

$$= 0.2 \cdot 0.3 + 0 \cdot 0.7 + 0 \cdot 0 + 0 \cdot 0 = 0.06$$

Example: Usain Bolt

- Given:

$$A = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{array}{c} \begin{array}{ccccccc} & F_2 & & & & & \end{array} \\ \begin{bmatrix} 0.5 & 0.2 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & \boxed{0} & 0.2 & 0.7 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0.5 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad \pi = [1 \quad 0 \quad 0 \quad 0]$$

- How to calculate $\mathrm{P}(X_2 = ACCELERATE | O_2 = F_2)$?
  - From previous slides $p(0_2 = F_2) = 0.06$ and $p(X_2 = ACCELERATE) = 0.7$
  - Use Bayes rule

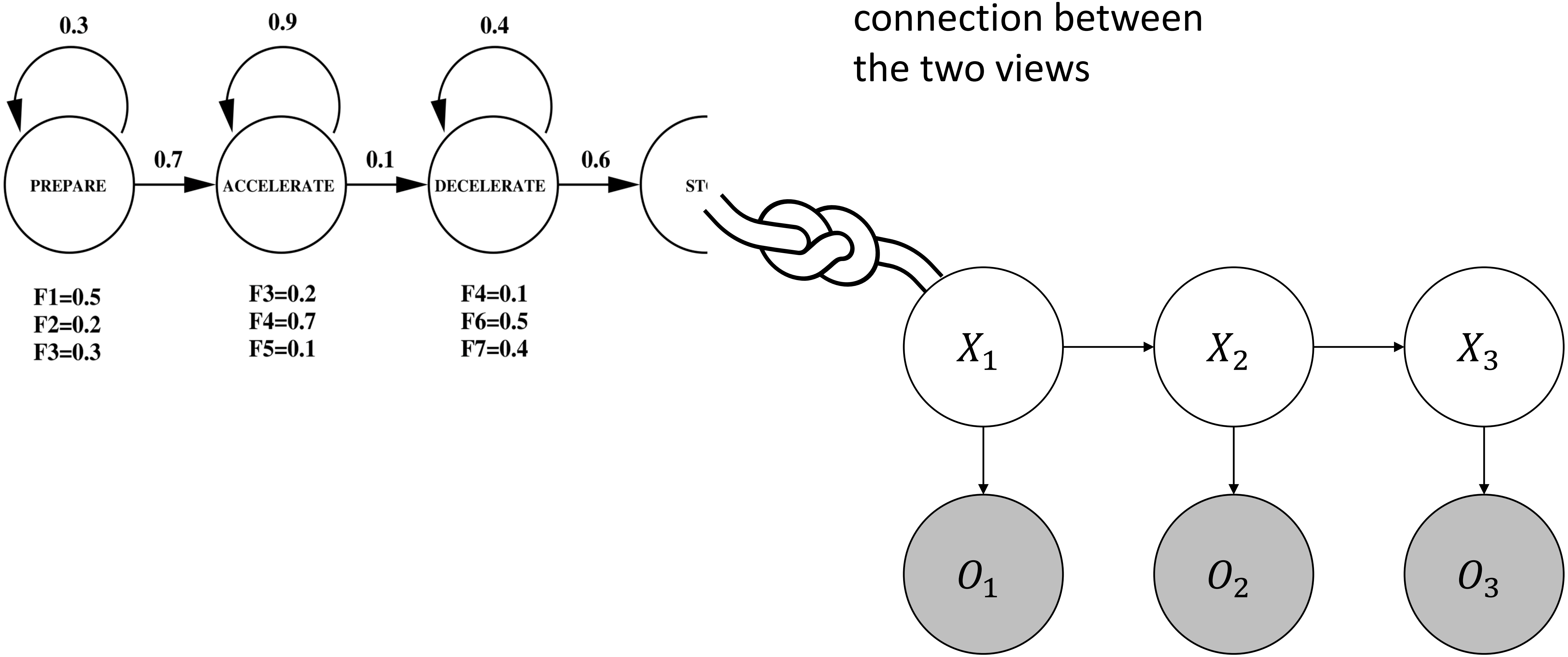
$$p(X_2 = ACCELERATE \mid O_2 = F_2) = \frac{p(O_2 = F_2 \mid X_2 = ACCELERATE)p(X_2 = ACCELERATE)}{p(O_2 = F_2)}$$

$$= \frac{0 \cdot 0.7}{0.06} = 0$$

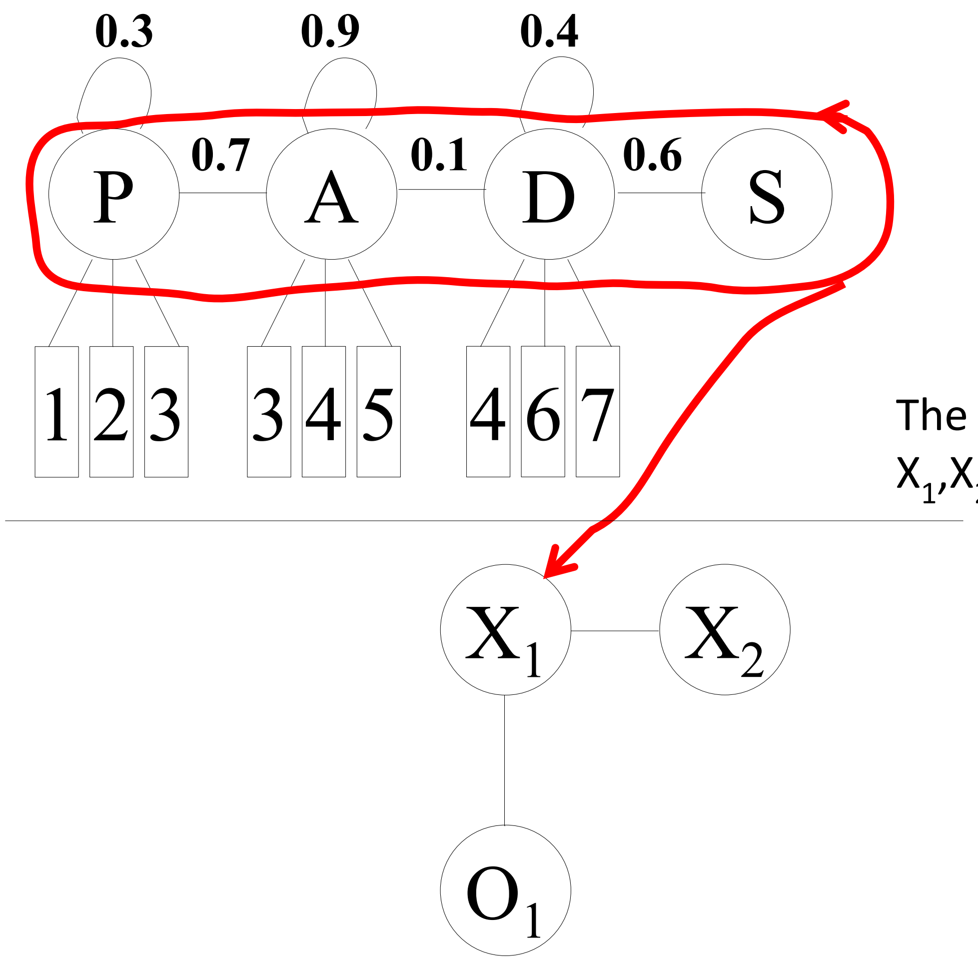
- Could we have seen this immediately?
  - YES. Cannot measure F2 in Accelerate state, i.e., p=0

Example: Usain Bolt

Let's look at the connection between the two views

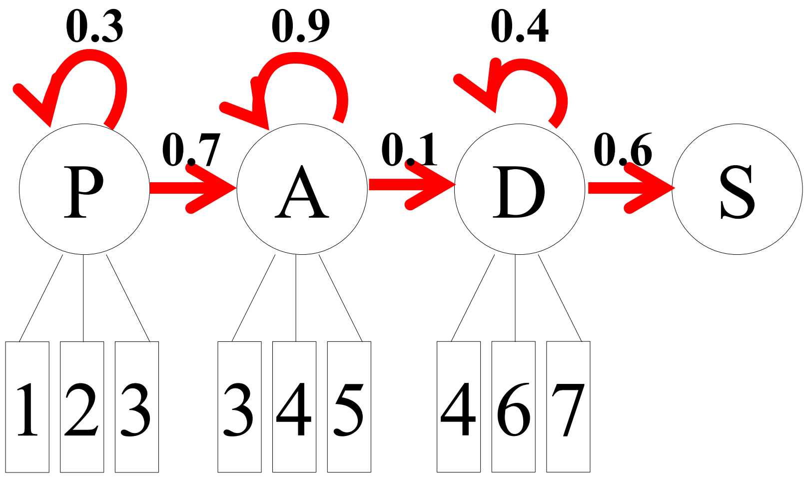


The states

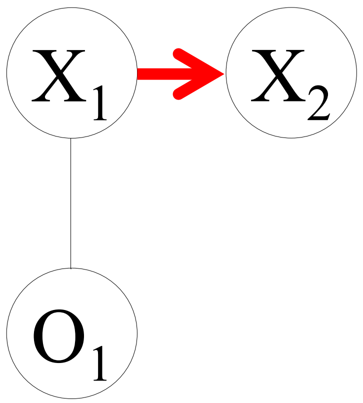


The states that variables X_1,X_2,... can be in (remember discrete)

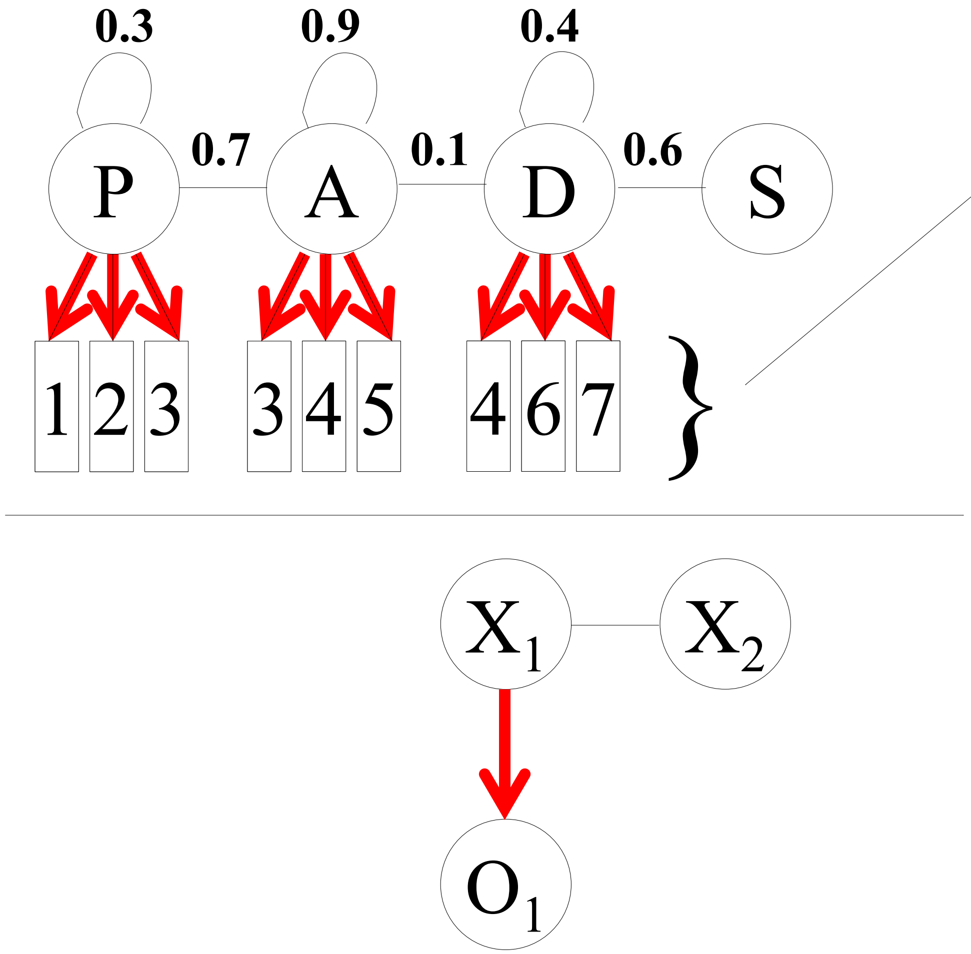
The state transitions



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)



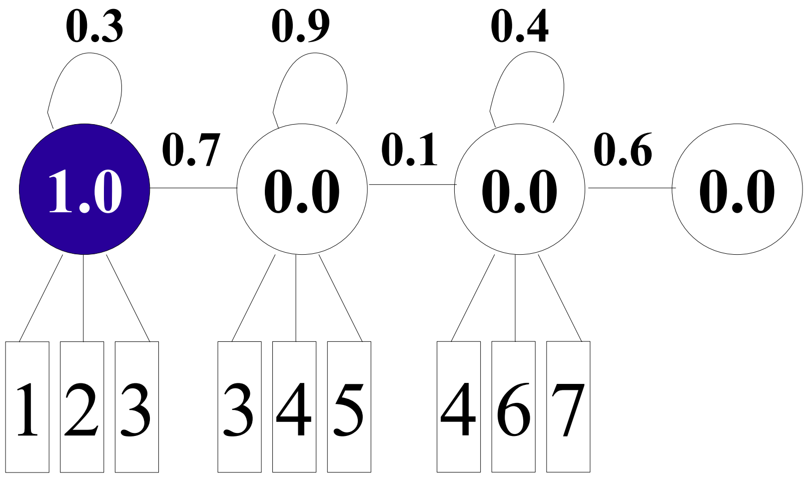
The observation/emission model



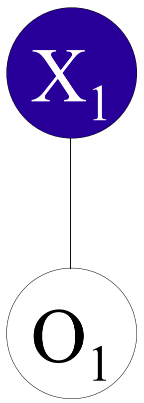
Different possible observations,
(aka “symbols”)

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]



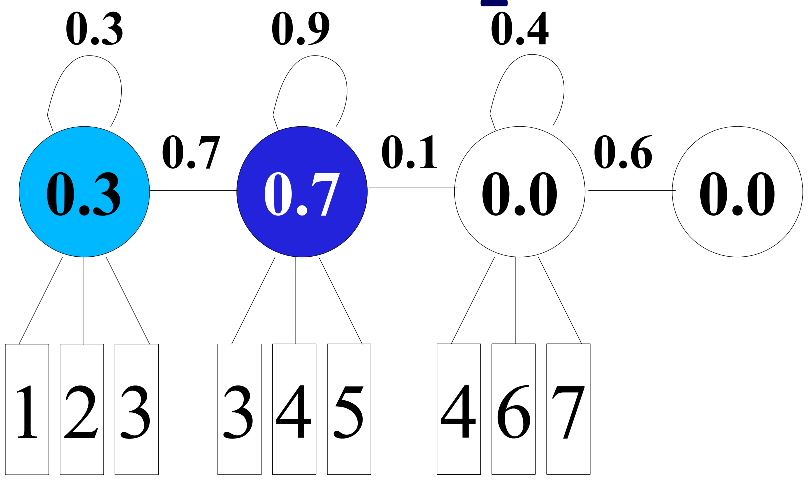
The darker the higher probability



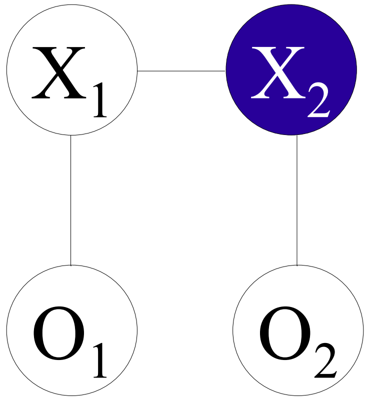
$$\pi = [1 \quad 0 \quad 0 \quad 0]$$

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

Prediction for X_2 (= π A)



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

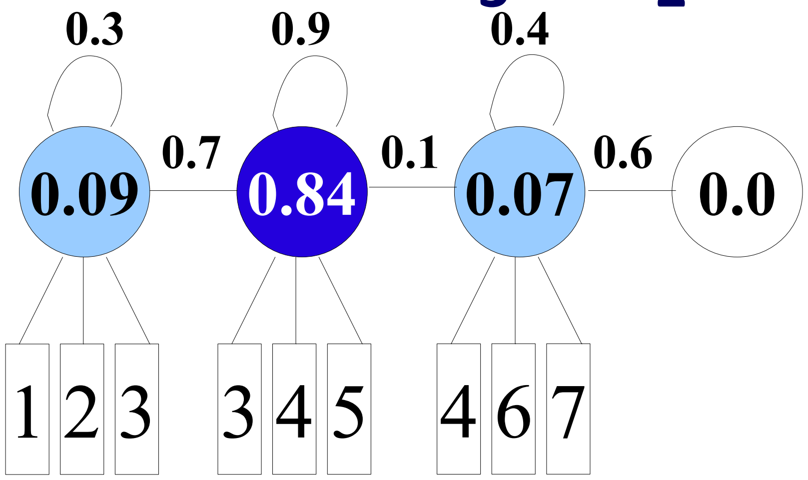


$$\pi = [1 \quad 0 \quad 0 \quad 0]$$

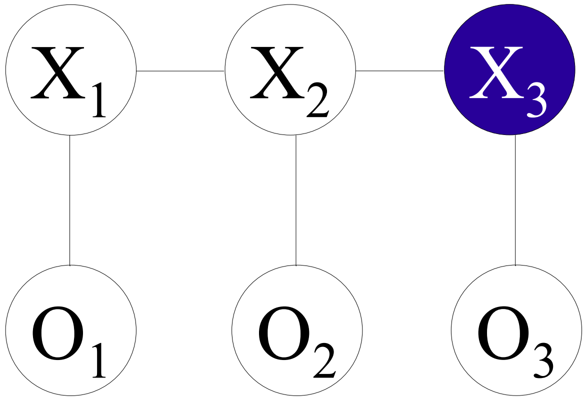
$$A = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{\pi}\mathbf{A} = \mathbf{X_2} = [\mathbf{0.3} \quad \mathbf{0.7} \quad \mathbf{0} \quad \mathbf{0}]$$

Prediction for X_3 (=X_2A)



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

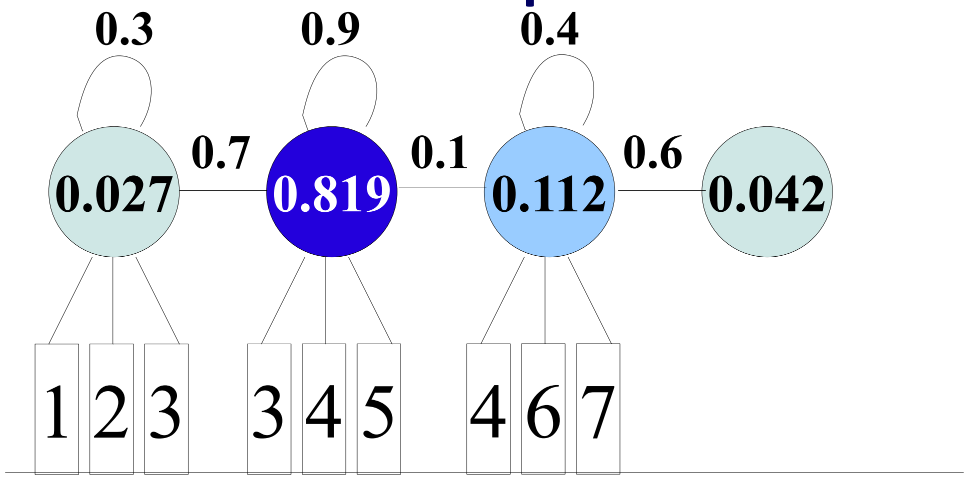


$$X_2 = [0.3 \quad 0.7 \quad 0 \quad 0]$$

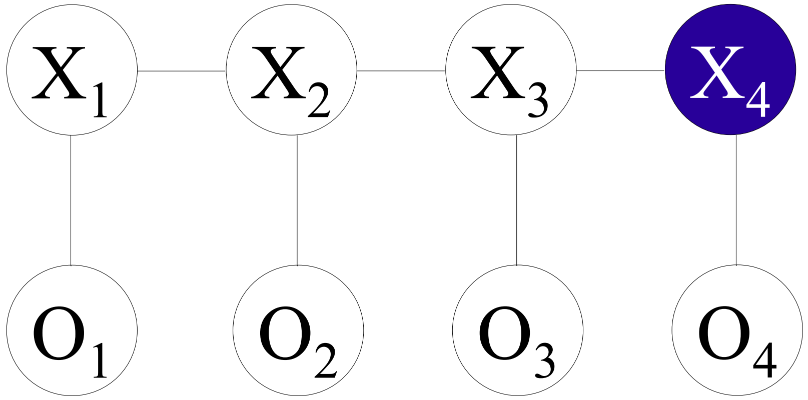
$$A = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{X_2A} = \mathbf{X_3} = [\mathbf{0.09} \quad \mathbf{0.84} \quad \mathbf{0.07} \quad \mathbf{0}]$$

Prediction for X_4



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!



$$X_3 = [0.09 \quad 0.84 \quad 0.07 \quad 0]$$

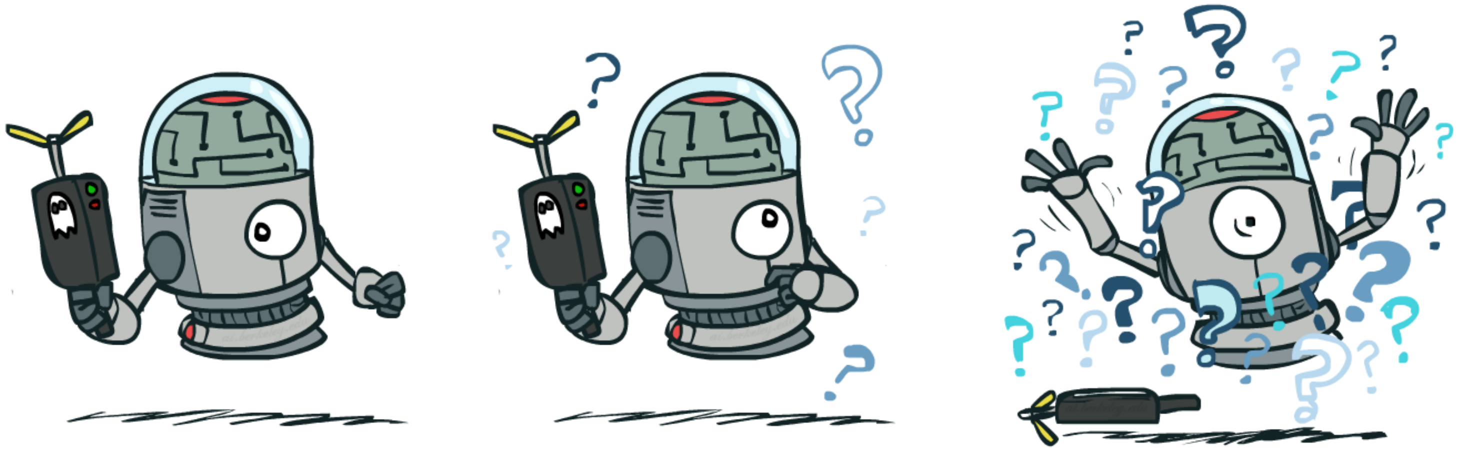
$$A = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{X_3A} = \mathbf{X_4} = [\mathbf{0.027} \quad \mathbf{0.819} \quad \mathbf{0.112} \quad \mathbf{0.042}]$$

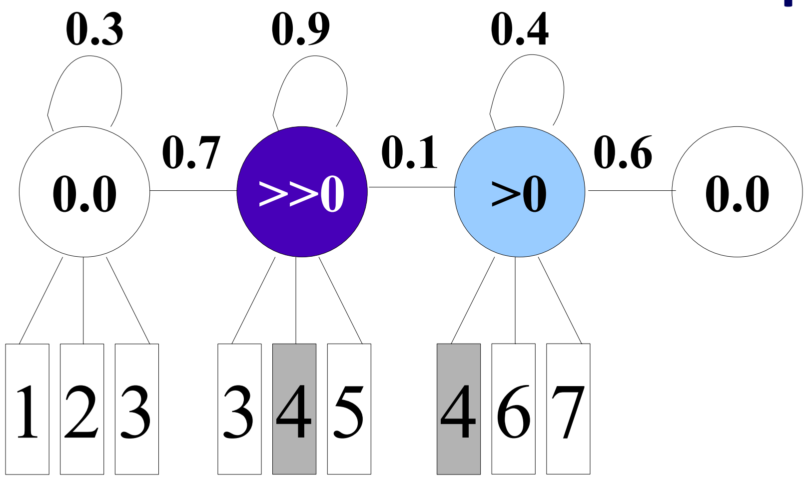
Measurements?

- 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)



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

$$p(X_4 = j \mid O_4) = \{Bayesrule\}$$

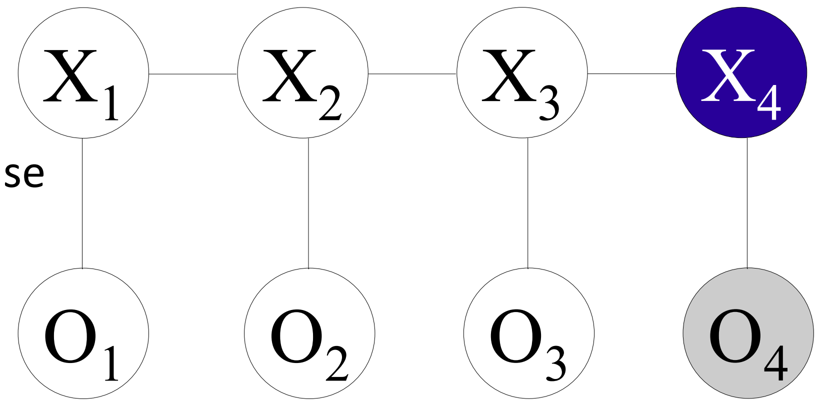
$$= \frac{p(O_4 = 4 \mid X_4 = j)p(X_4 = j)}{p(0_4)}$$

$$= \frac{b_j(4)p(X_4 = j)}{p(0_4)}$$

$$= \eta b_j(4)p(X_4 = j)$$

$$with\ \eta = \frac{1}{\sum_j b_j(4) \times p(X_4 = j)}$$

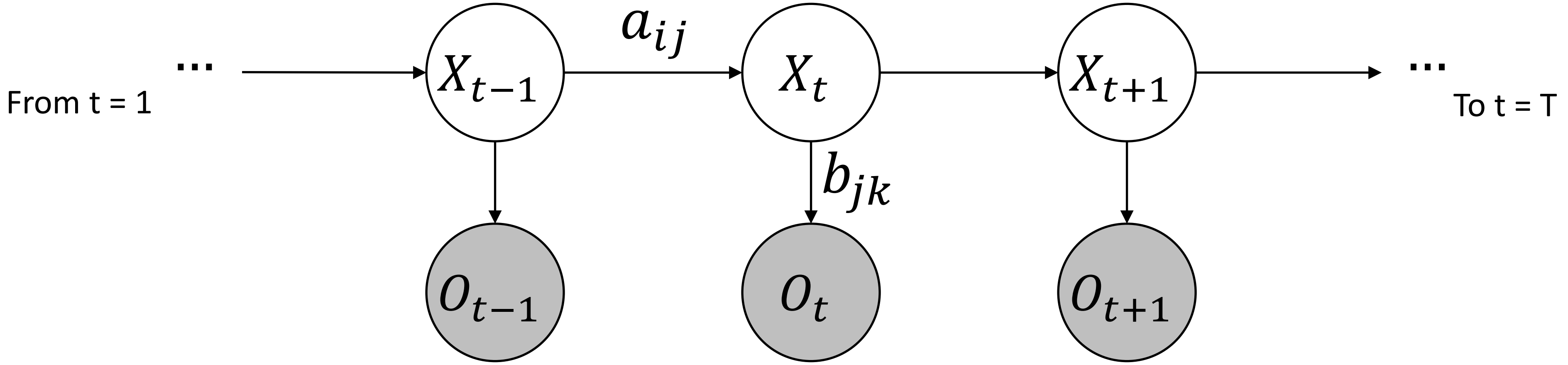
Weight with p(O|X)
→ not in state 1 for sure because we cannot measure 4 there



Observed (grey)

HMM Terminology

Time instants	$t\ in\ \{1,2\ldots T\}$
Hidden States / States / Emitters	X_t
Outputs / Emissions / Observations / Visible States	O_t
All possible states / states set	$X_t\ in\ \{1, 2, \ldots, N\}$
All possible emissions / emissions set	$O_t\ in\ \{1, 2, \ldots, K\}$
Initial state distribution / Initial state probabilities	$p_i\ in\ q\ or\ \pi_i\ in\ \pi$
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
   Backward algorithm

2. **Decoding**: 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

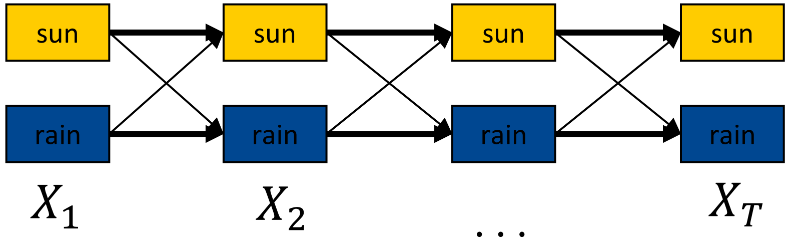
1. Compute likelihood $p(O_{1:t}|\lambda)$ of observed sequence given λ

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

1. Compute likelihood $p(O_{1:t}|\lambda)$ of observed 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:**
  - Probability that the given sequence **O** occurred **regardless of which X produced the sequence.**

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

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

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

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

- 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

Example for Clarity

If you have:

- $N = 2$ states (e.g., A and B),
- $T = 3$ time steps,

then there are $2^3 = 8$ possible sequences, such as:

1. $A \to A \to A$
2. $A \to A \to B$
3. $A \to B \to A$
4. $A \to B \to B$
5. $B \to A \to A$
6. $B \to A \to B$
7. $B \to B \to A$
8. $B \to B \to B$

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 $O_1, \ldots O_t$ AND at time t, being in state i

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

Forward algorithm (aka α-pass)

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

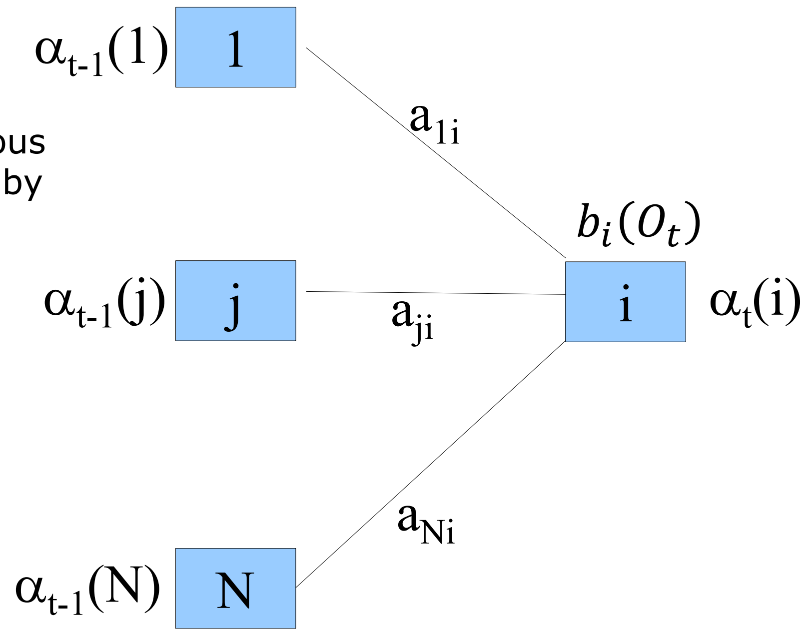
Prediction (sum of the previous partial probabilities multiplied by the transition probabilities)

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

Weight (observation probability)

- For 2≤t≤T:

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



Forward algorithm (aka α-pass)

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

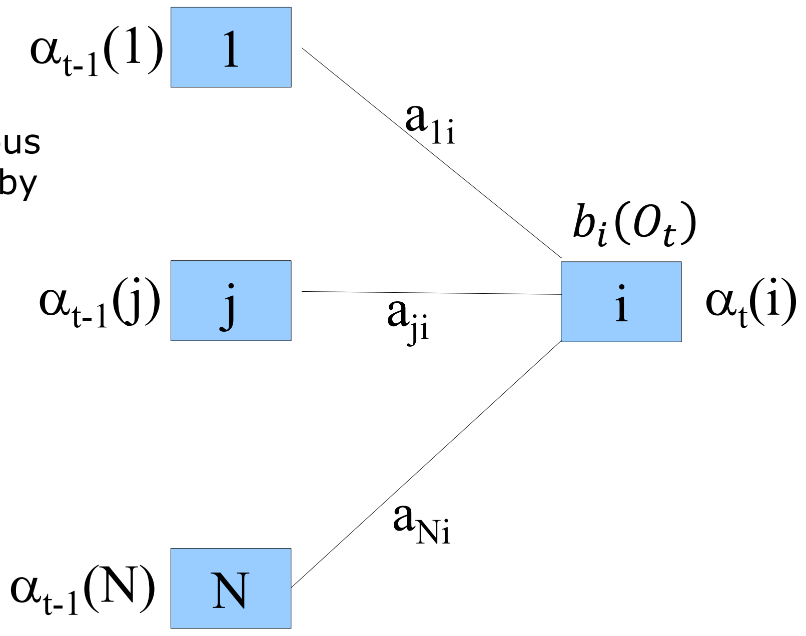
Prediction (sum of the previous partial probabilities multiplied by the transition probabilities)

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

Weight (observation probability)

- For 2≤t≤T:

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

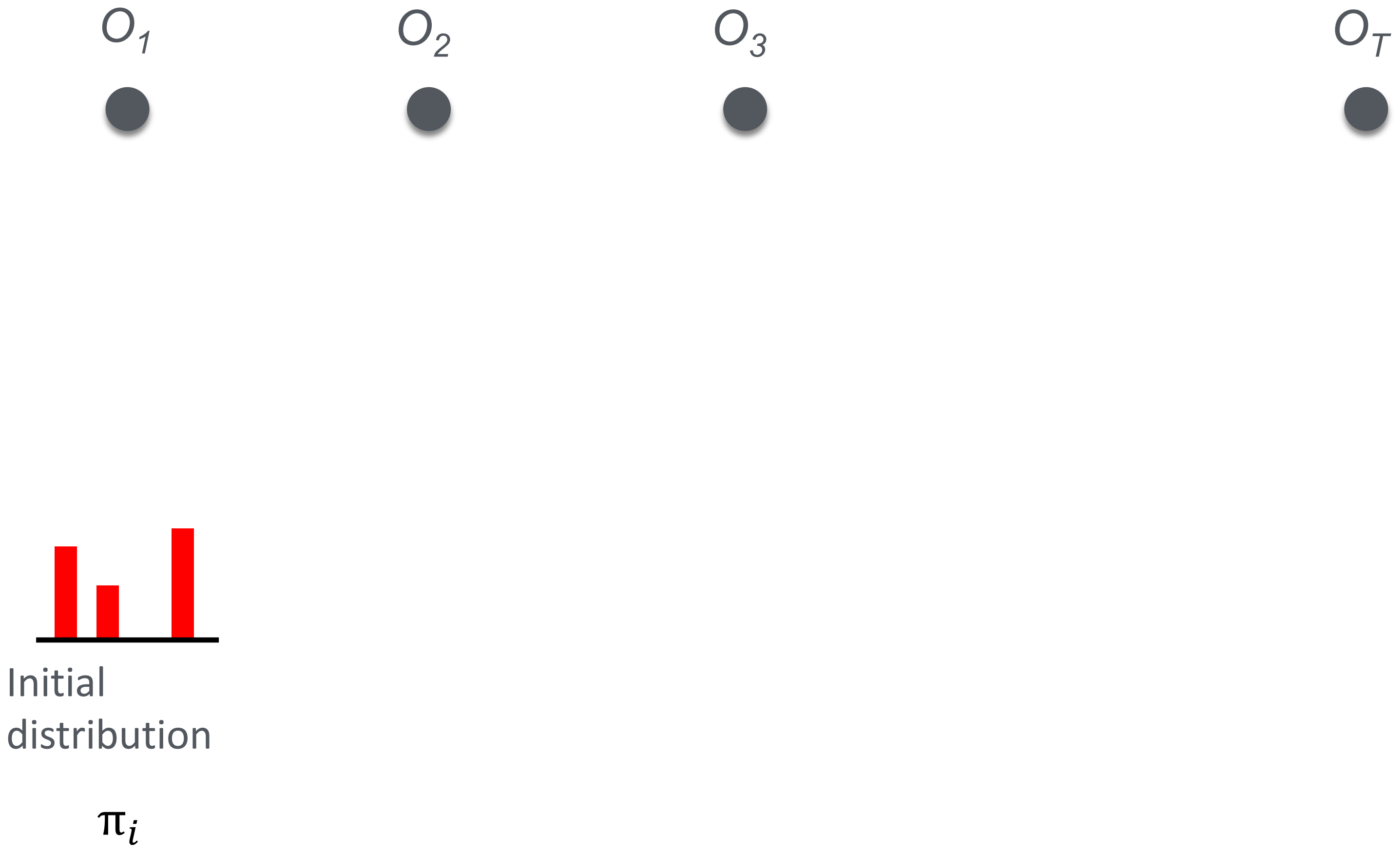


- Which gives us:

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

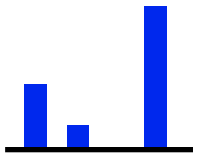
- recursive way to calculate likelihood with only N^2T multiplications (compared to $2TN^T$)

Forward algorithm intuition



Forward algorithm intuition

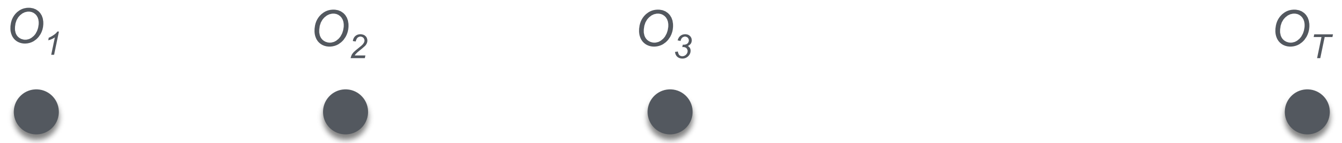
O_1 O_2 O_3 O_T

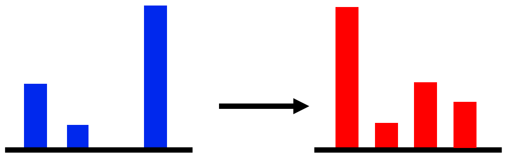


"Improved" distribution

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

Forward algorithm intuition

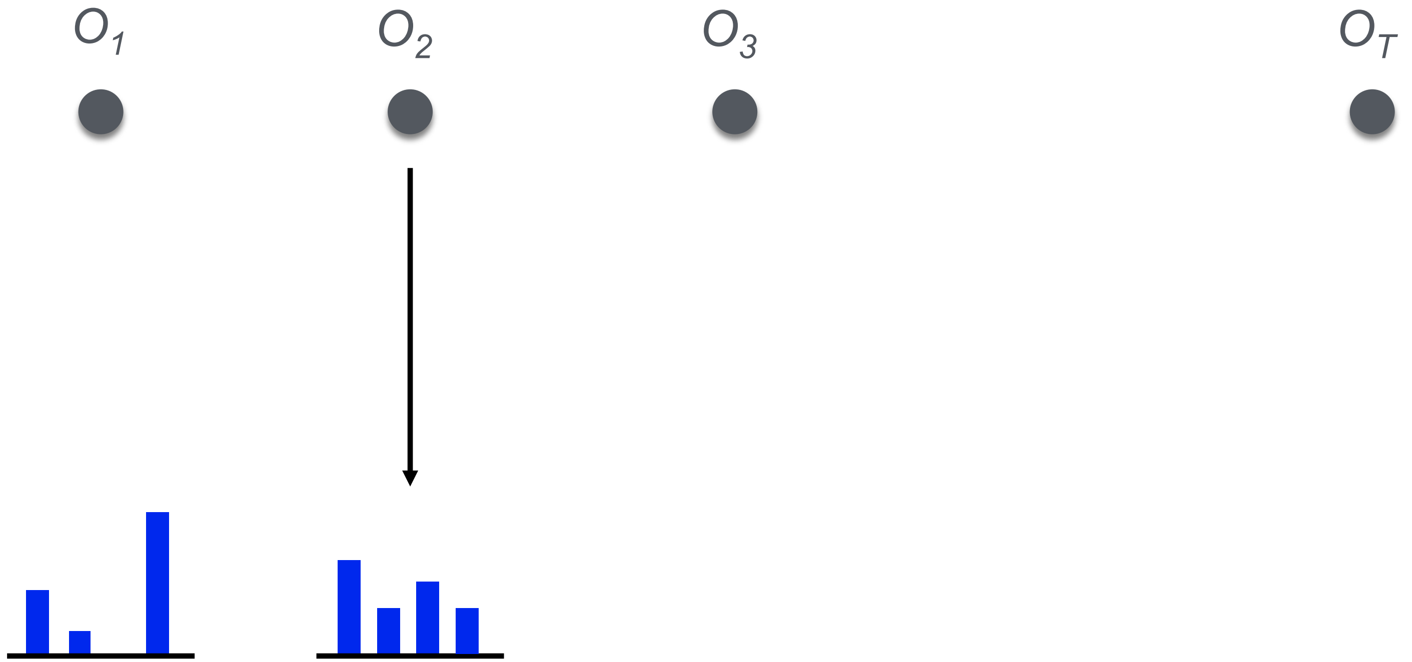




Predicted distribution

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

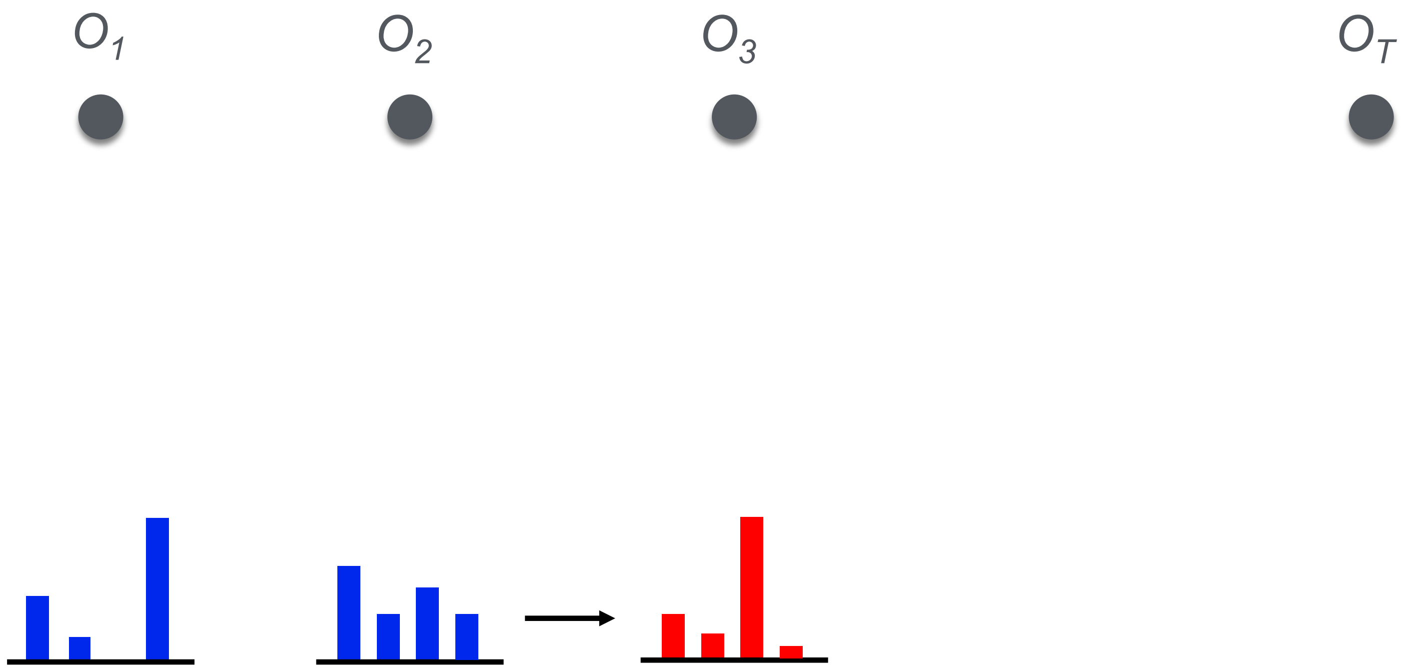
Forward algorithm intuition



"Improved" distribution

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

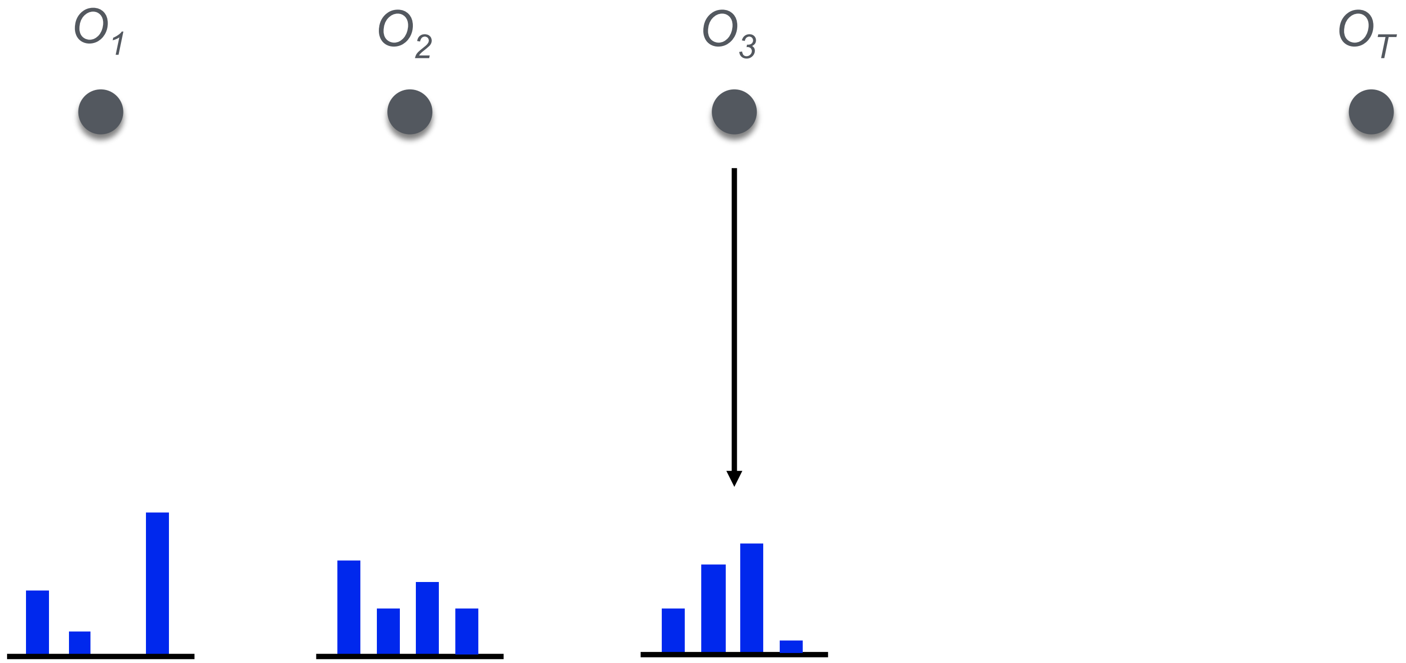
Forward algorithm intuition



Predicted distribution

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

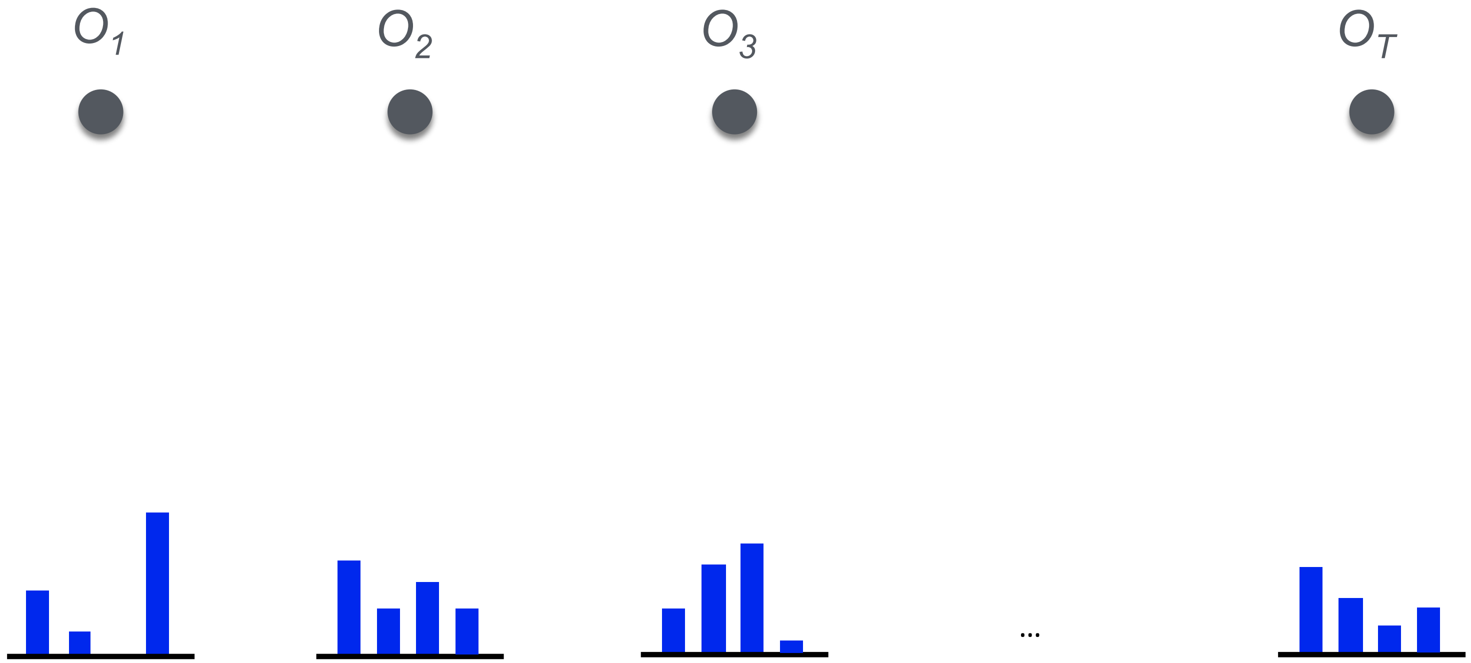
Forward algorithm intuition



"Improved" distribution

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

Forward algorithm intuition



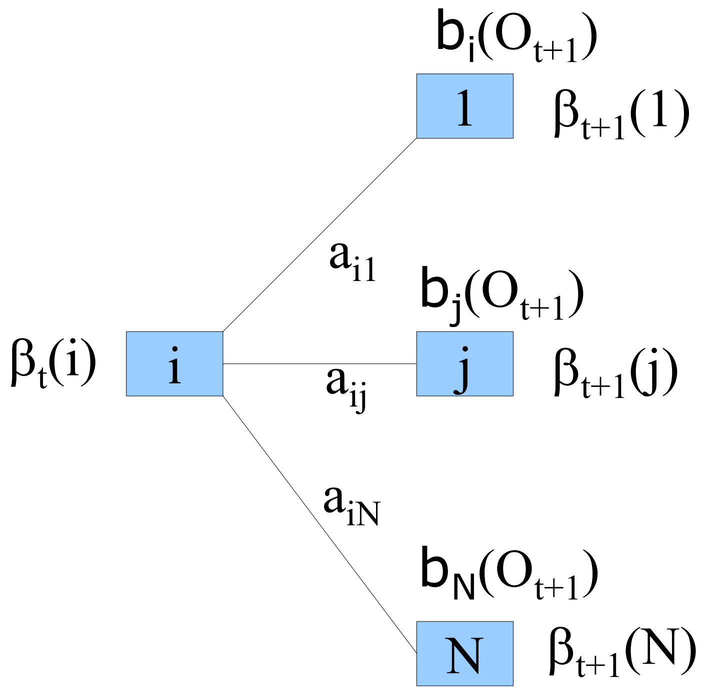
Backward algorithm (aka β-pass)

- βt(i) = Probability that the model is in the hidden state $X_t(i)$ $(i\ in\ [1,2,\ldots,N])$
- **will generate** the remainder of the emission sequence, from Ot+1 to OT, 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

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
   Backward algorithm

2. **Decoding**: 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

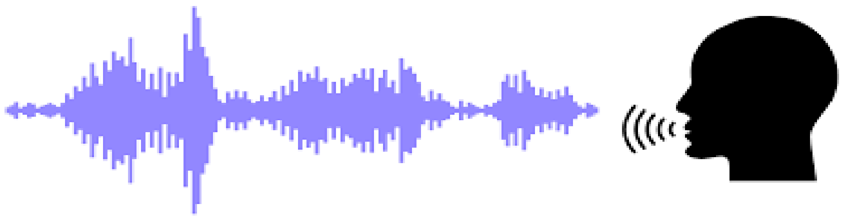
   Baum-Welch algorithm

2. Calculate 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 spoken

“Recognize speech”

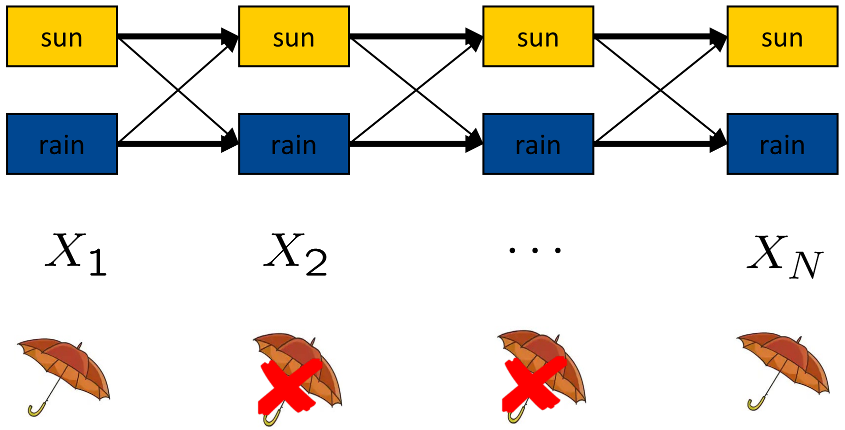
“Wreck a nice beach”



  - 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, π
- To Find:
  - Hidden state sequence $\boldsymbol{X}^* = \{X_1, X_2 \dots X_T\}$ that most likely produced **O**.
  - Probability of occurrence of $\boldsymbol{X}^*$



2. Calculate most likely state sequence

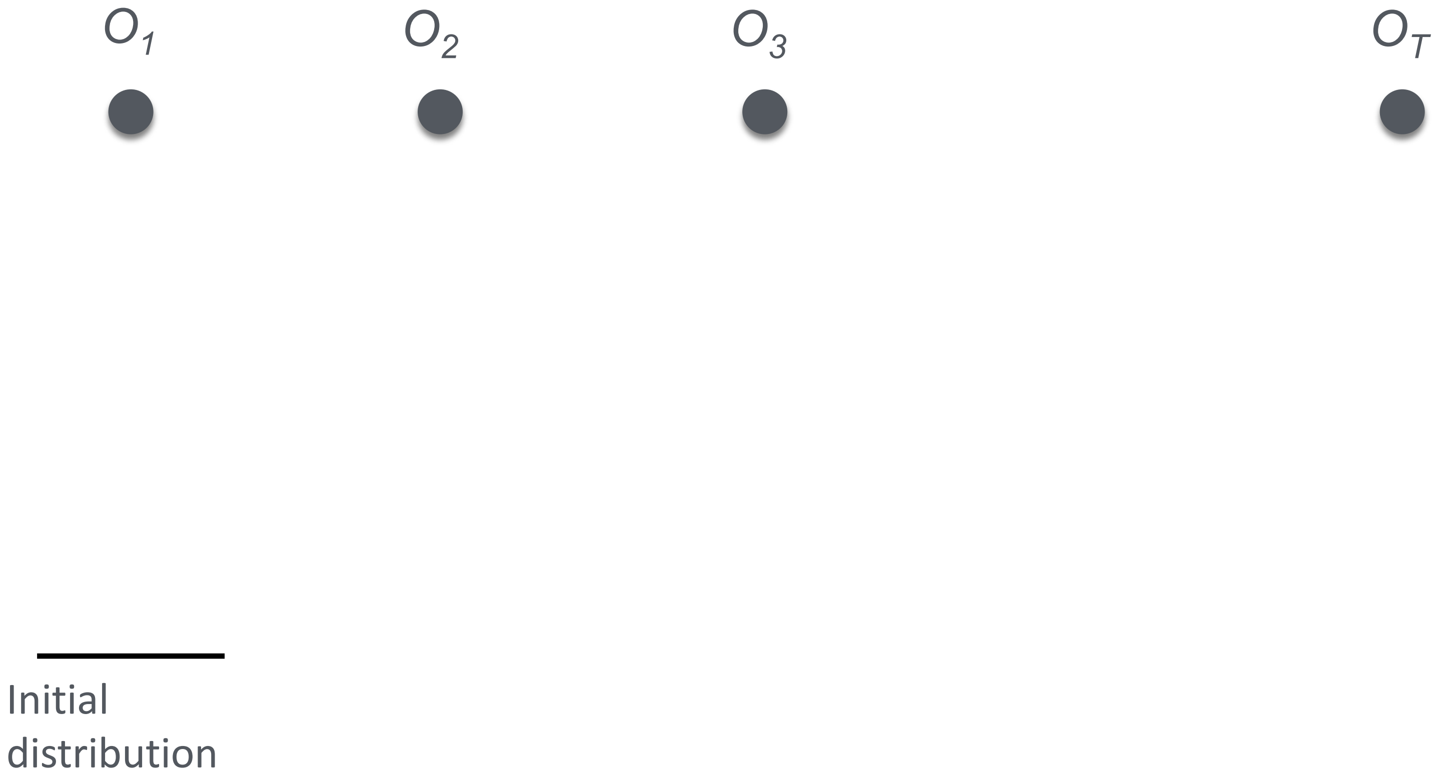
- We can find the most likely sequence by listing all possible sequences and finding the prob of the observed sequence for each of the combinations

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

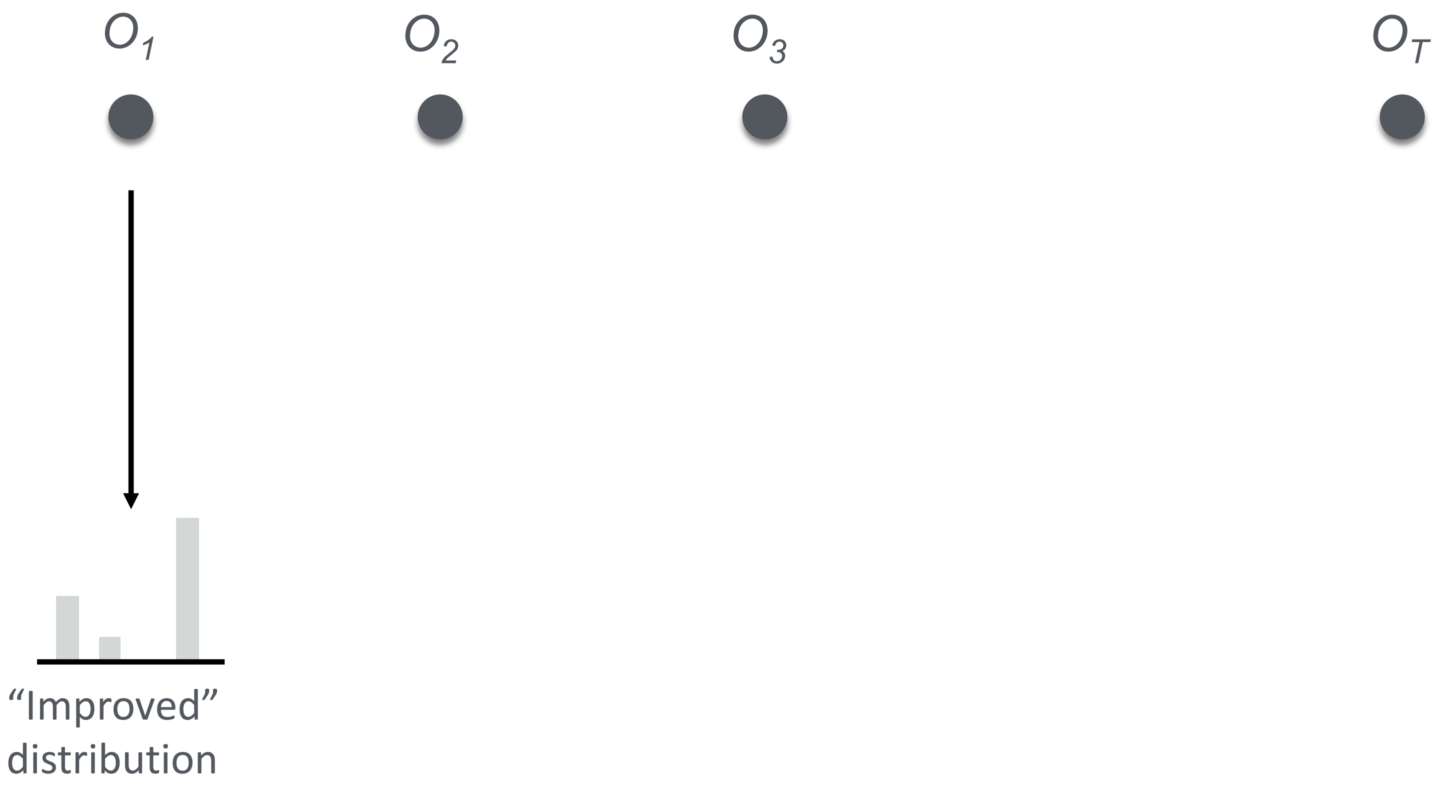
- Cannot solve individually for each time step

- Need to optimize the sequence not just individual states

2. Calc most likely state sequence

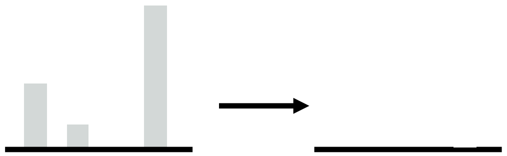


2. Calc most likely state sequence



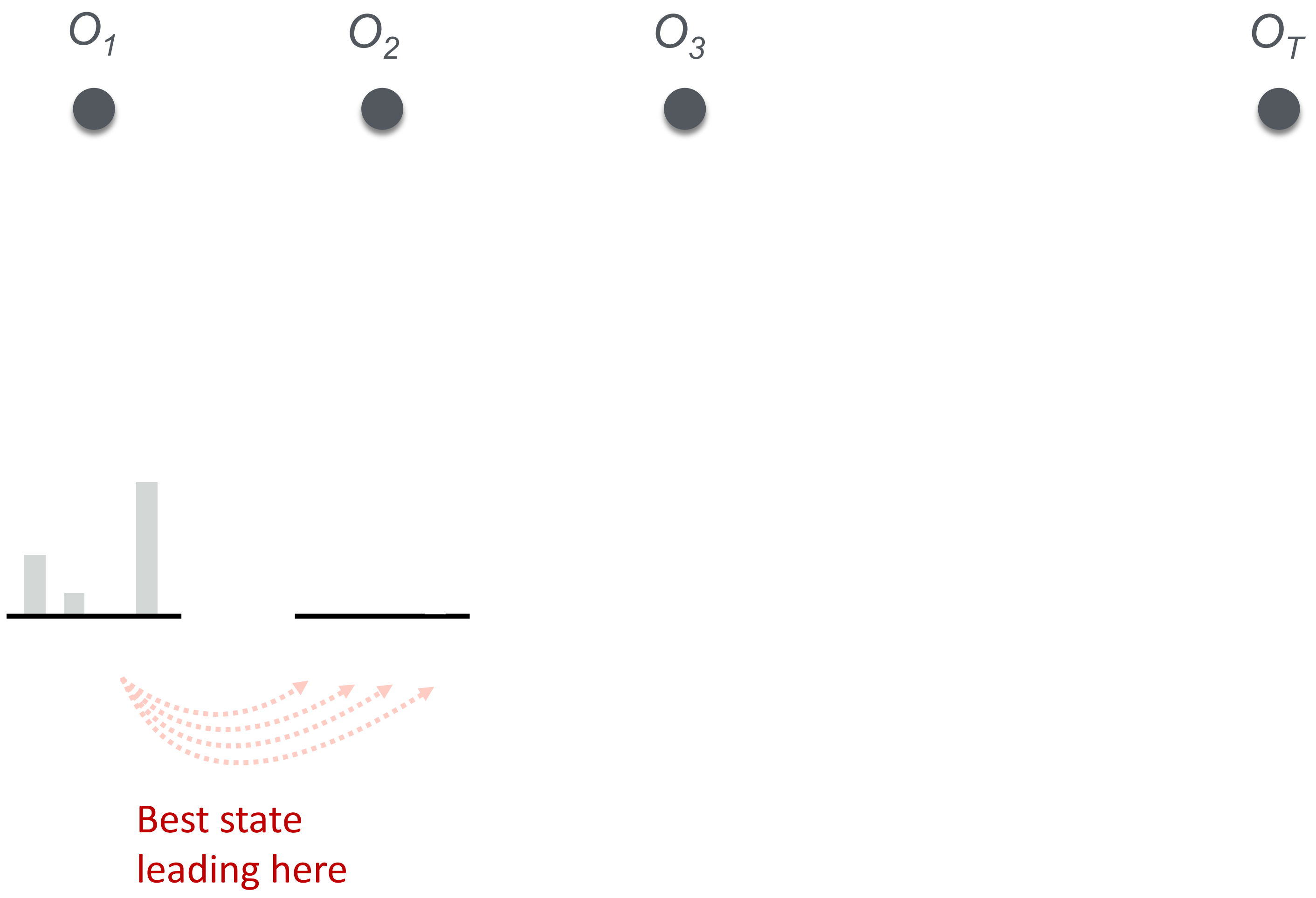
2. Calc most likely state sequence

O_1 O_2 O_3 O_T

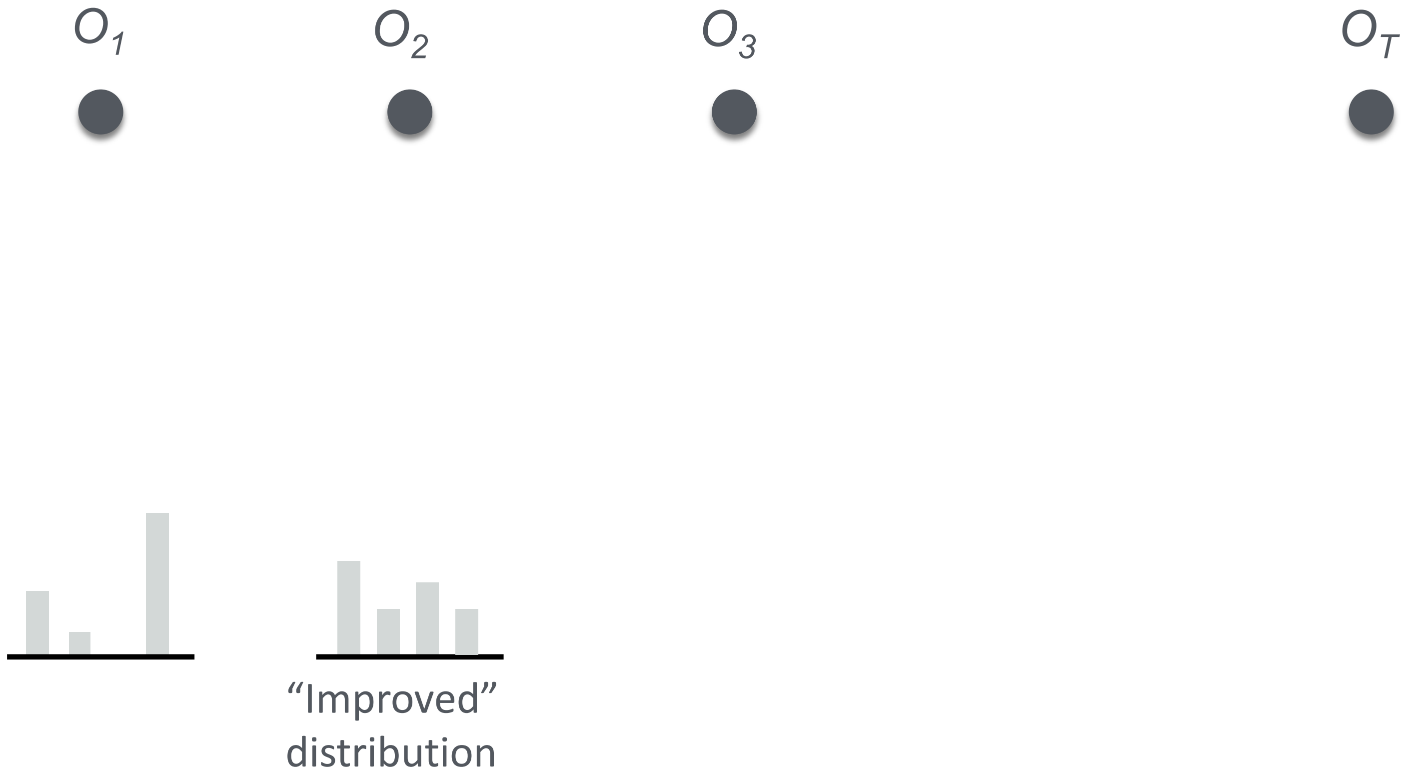


Predicted distribution

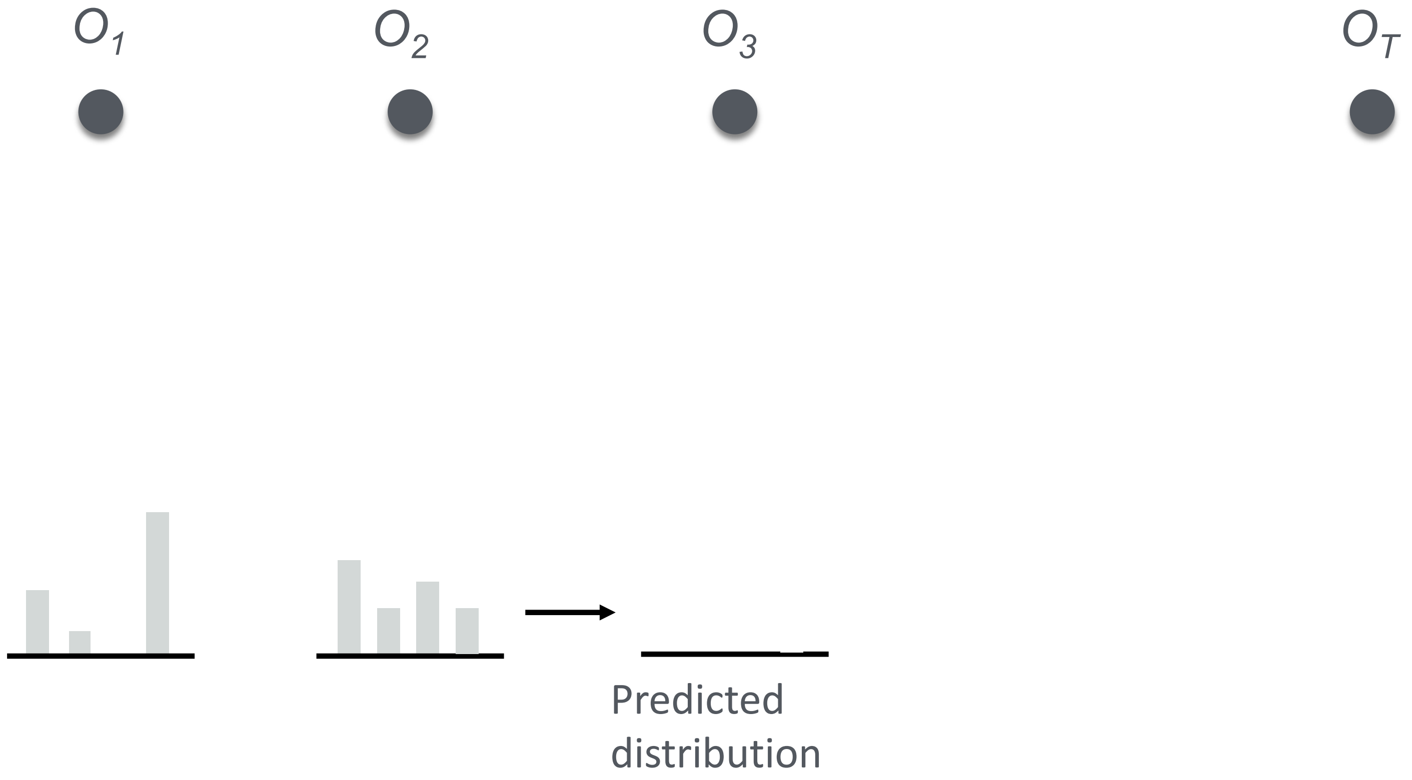
2. Calc most likely state sequence



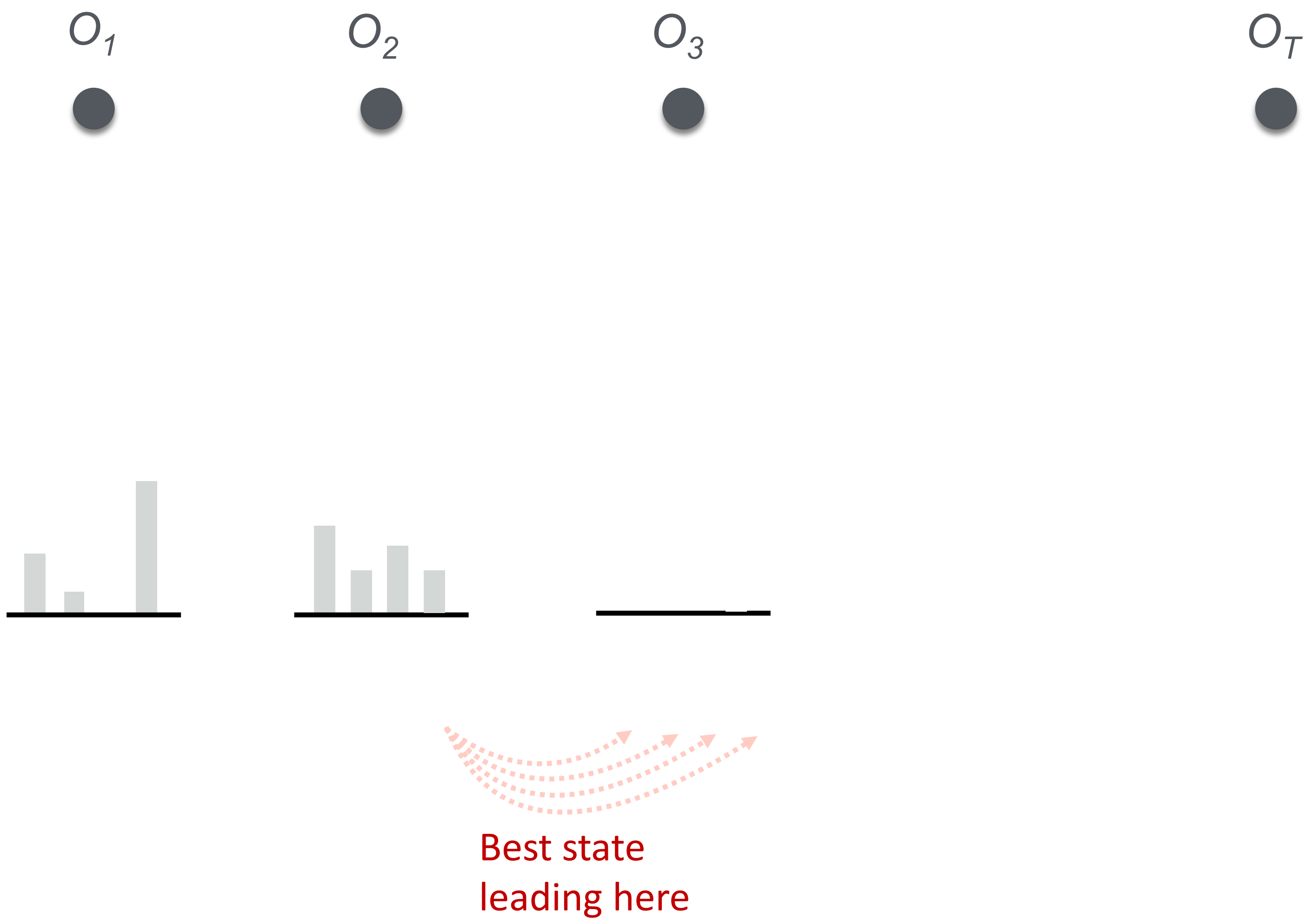
2. Calc most likely state sequence



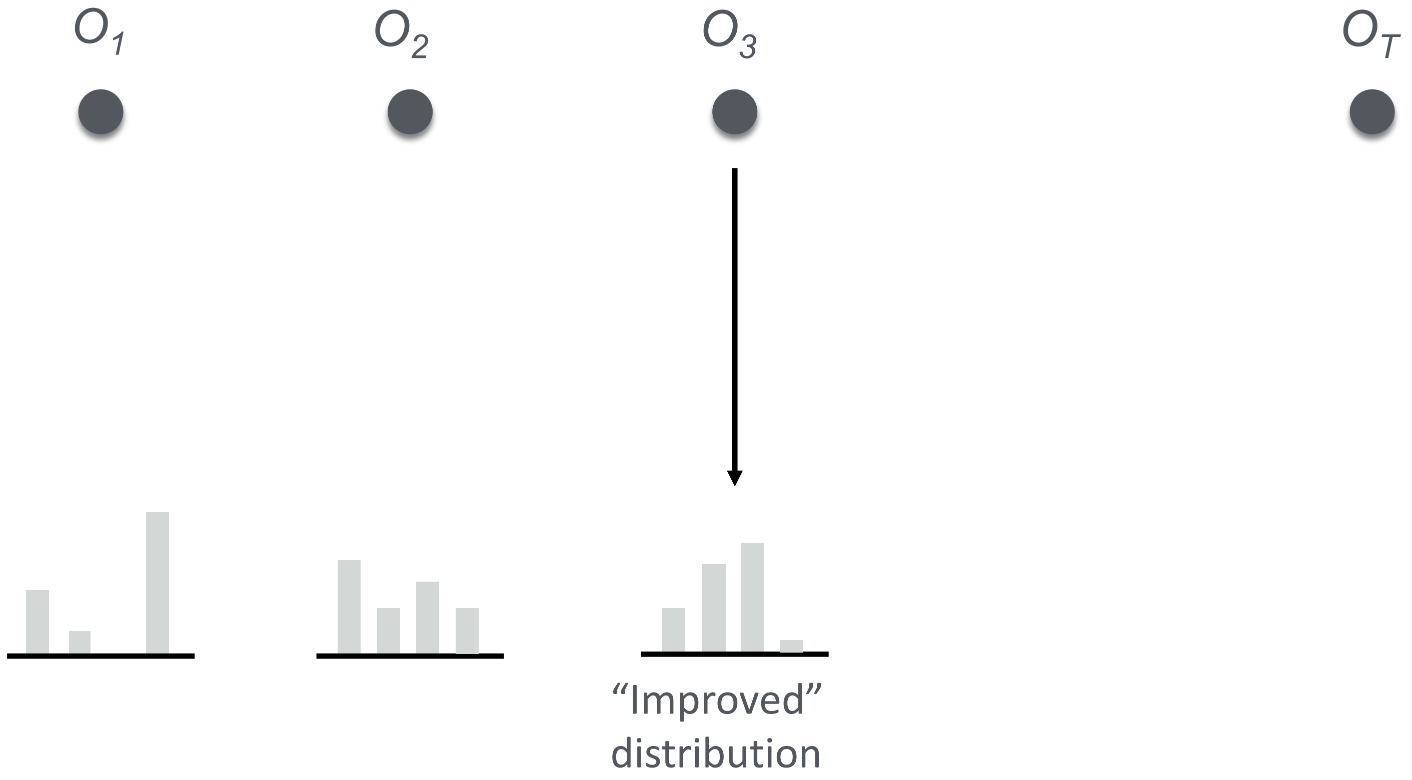
2. Calc most likely state sequence



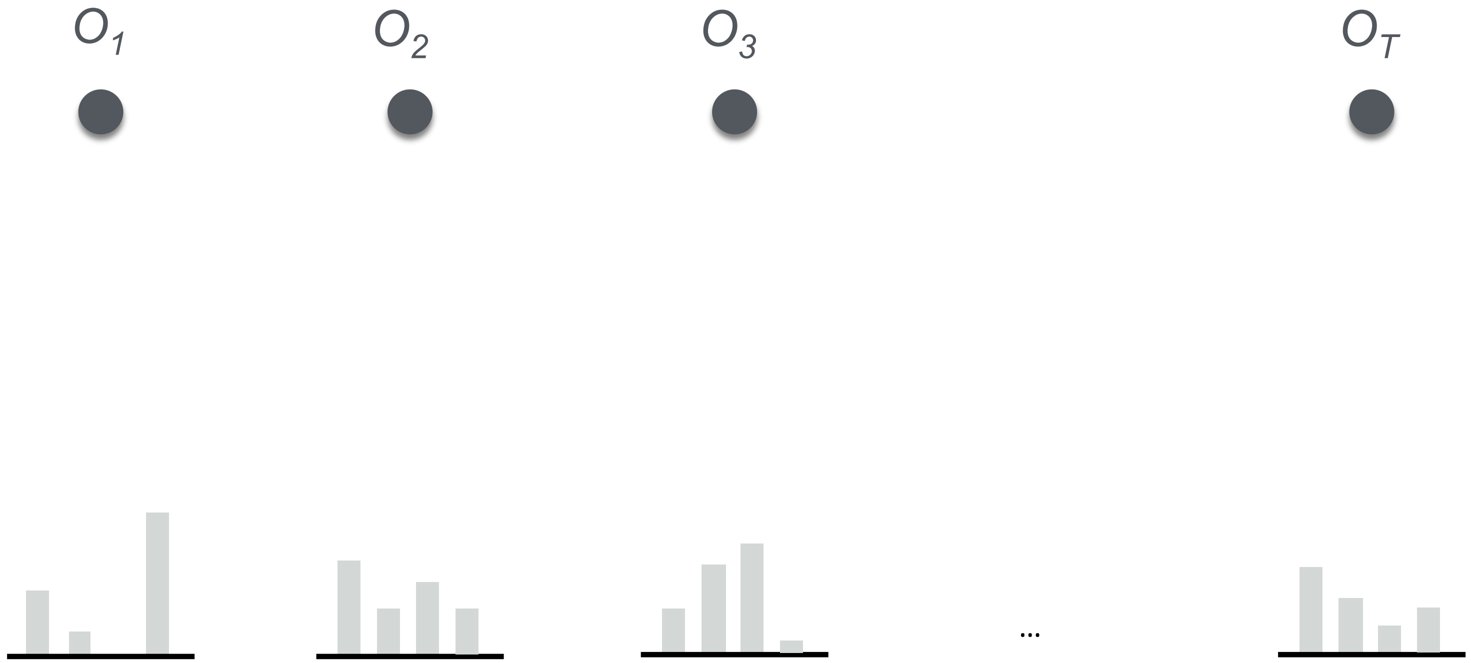
2. Calc most likely state sequence



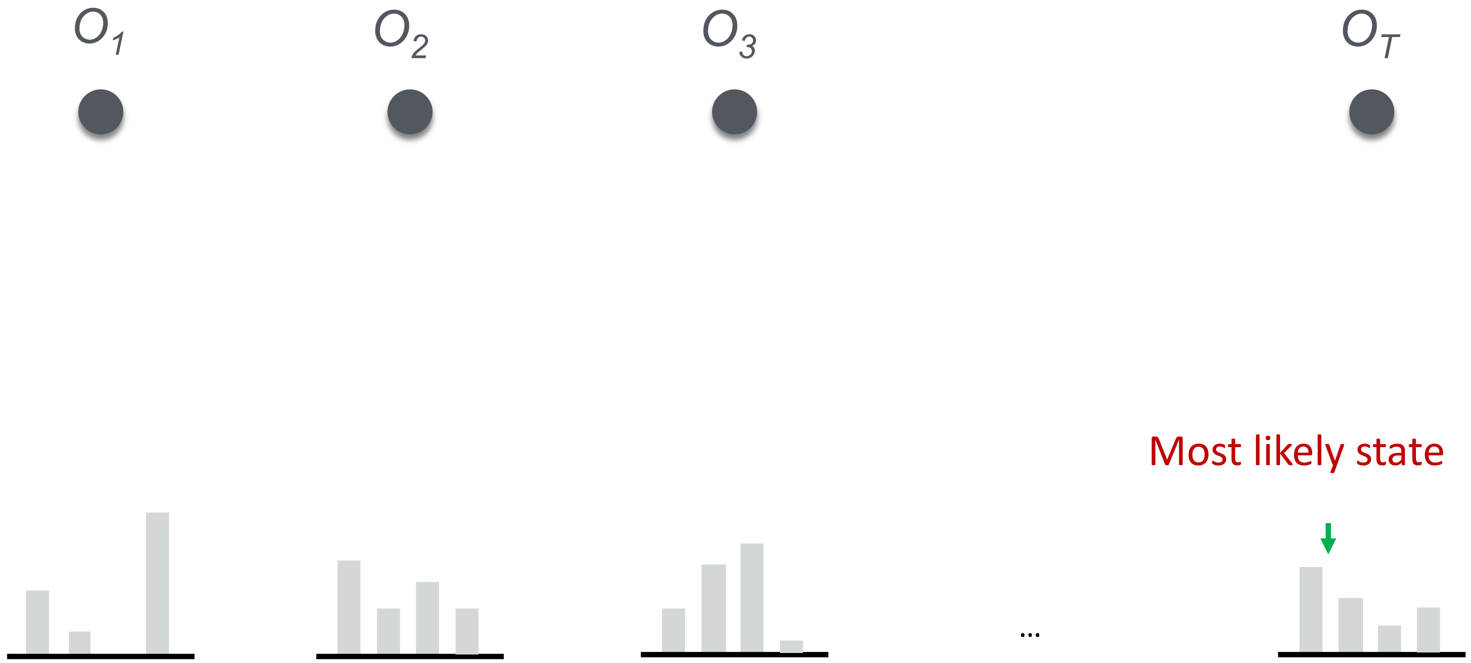
2. Calc most likely state sequence



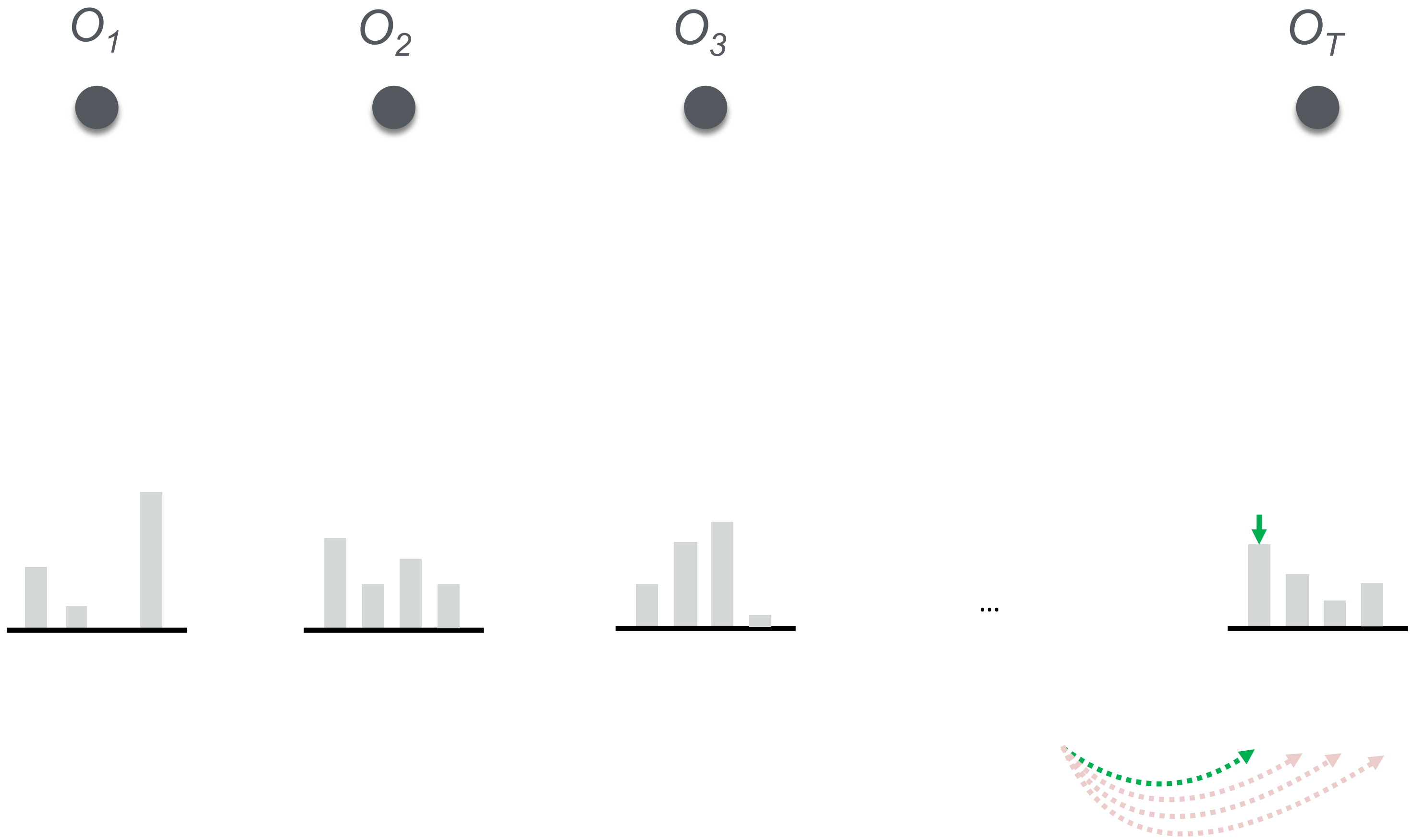
2. Calc most likely state sequence



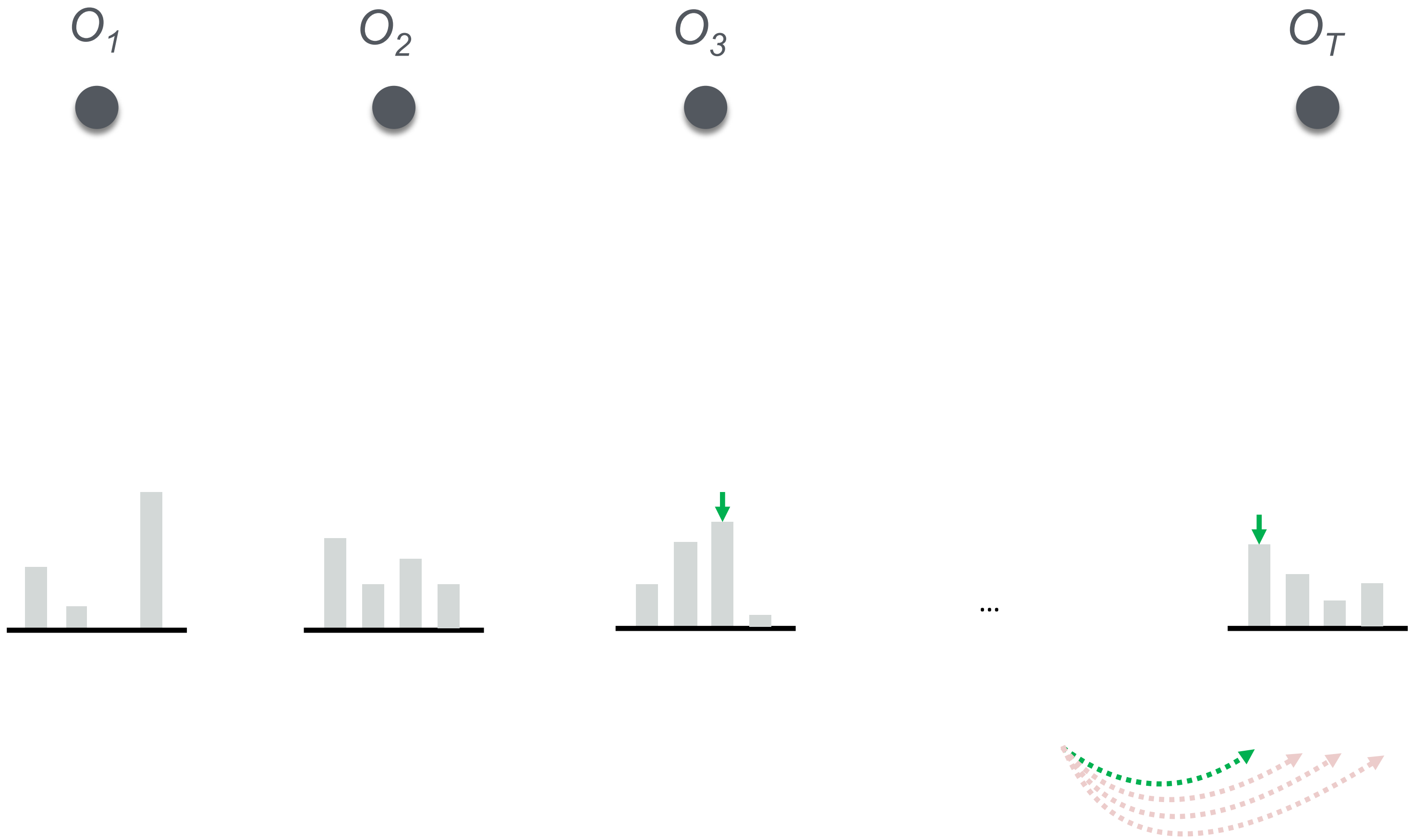
2. Calc most likely state sequence



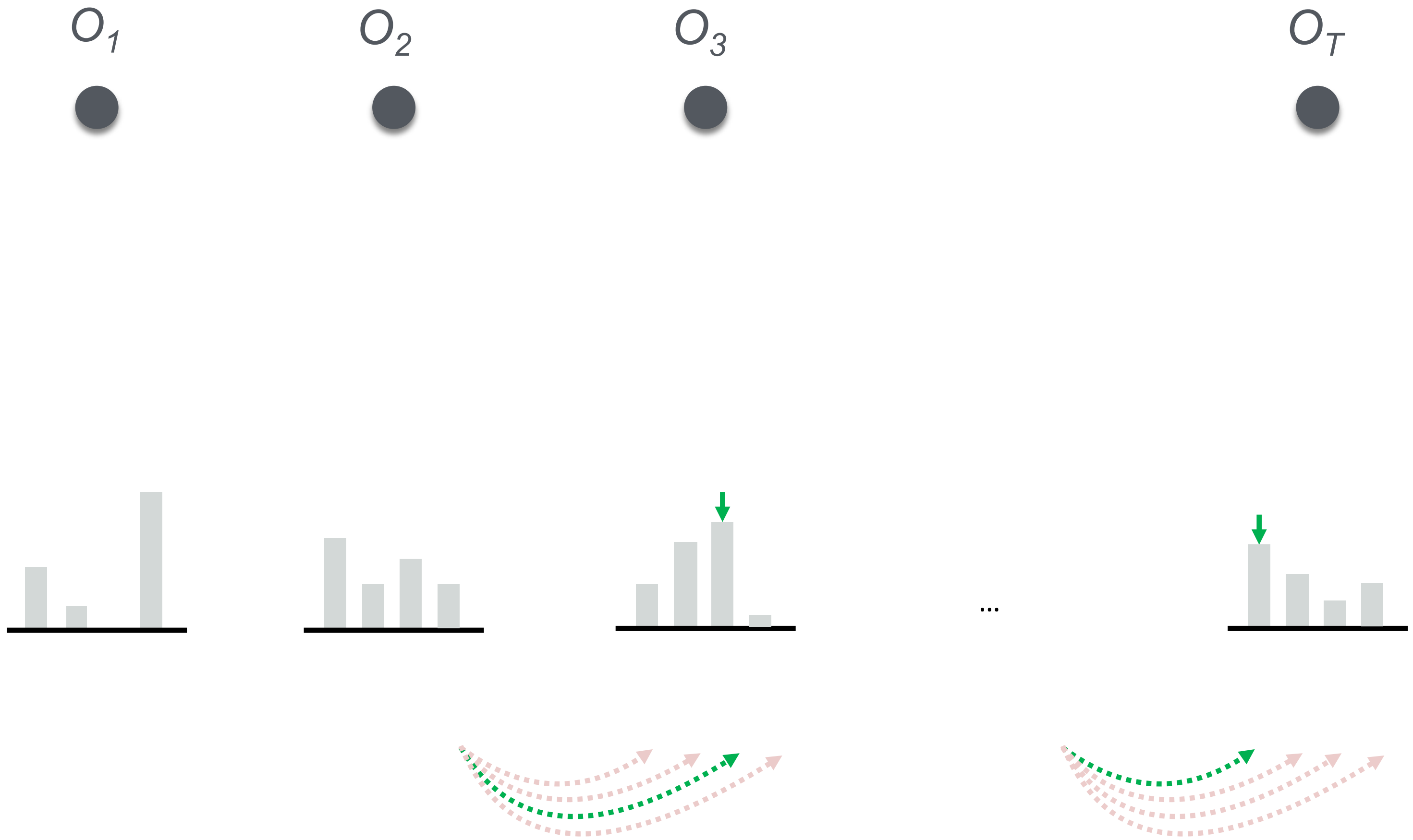
2. Calc most likely state sequence



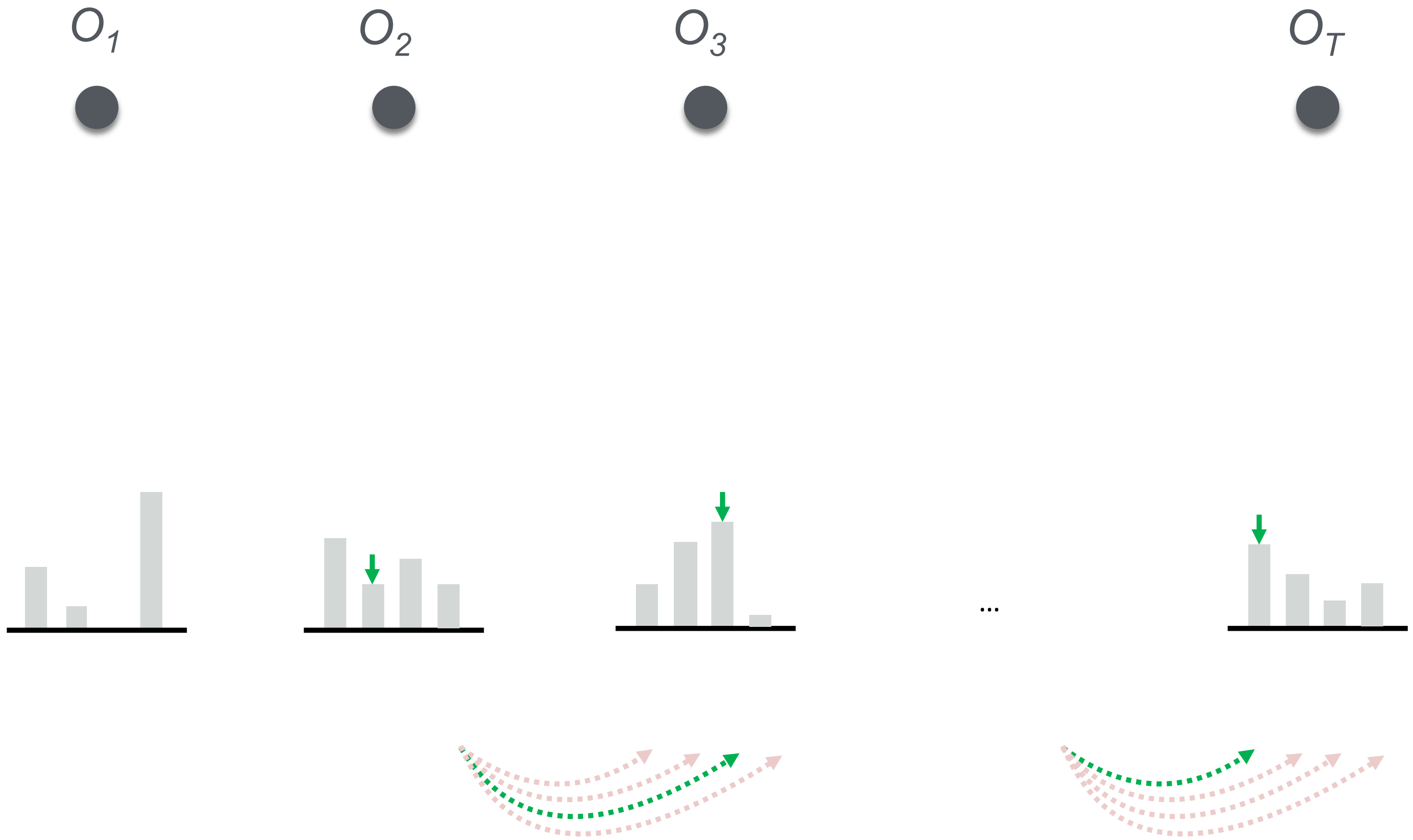
2. Calc most likely state sequence



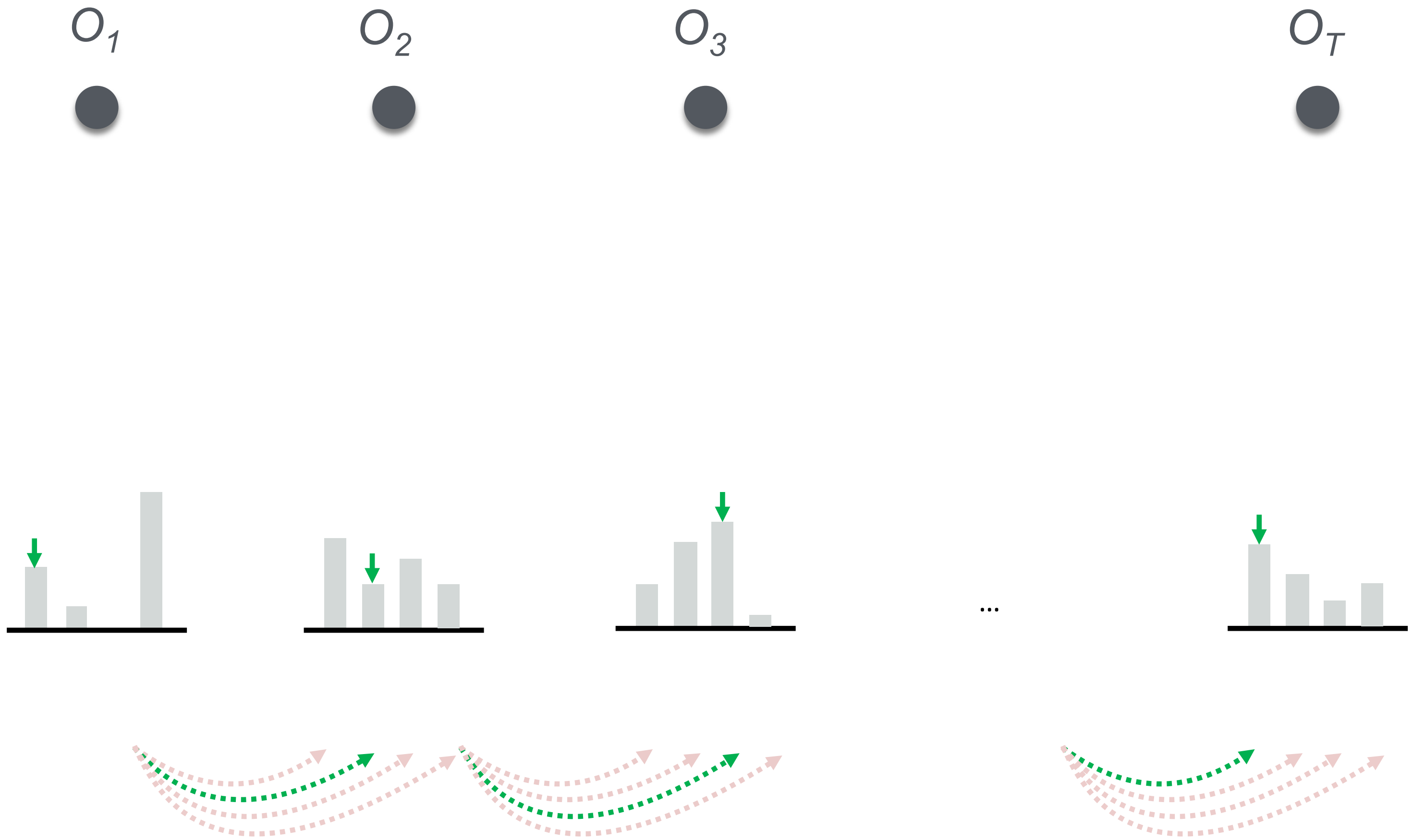
2. Calc most likely state sequence



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: $\Psi = \operatorname{argmax}_{j \in \{1,\ldots,N\}} [\delta_{t-1}(j) a_{ji}]$

Forward vs. Viterbi algorithm

- Both dynamic programming algorithms
  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)

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

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: $\max_{j \in \{1,\ldots,N\}} \left[\widehat{\delta_{T-1}}(j) \right]$

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
   Backward algorithm

2. **Decoding**: 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

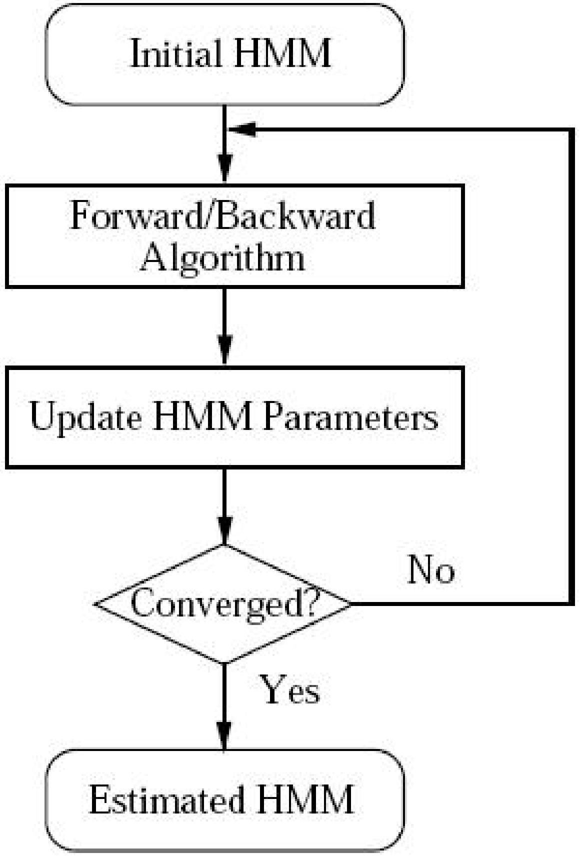
3. Estimate model parameters $\Lambda=\{\lambda\}$ given $O_{1:T}$

- Analogous to the training phase of Machine Learning
- Motivating examples:
  - Learn a model for a letter for recognizing handwritten text
  - Learn a model for the word "Bayesian" from audio data
  - 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.

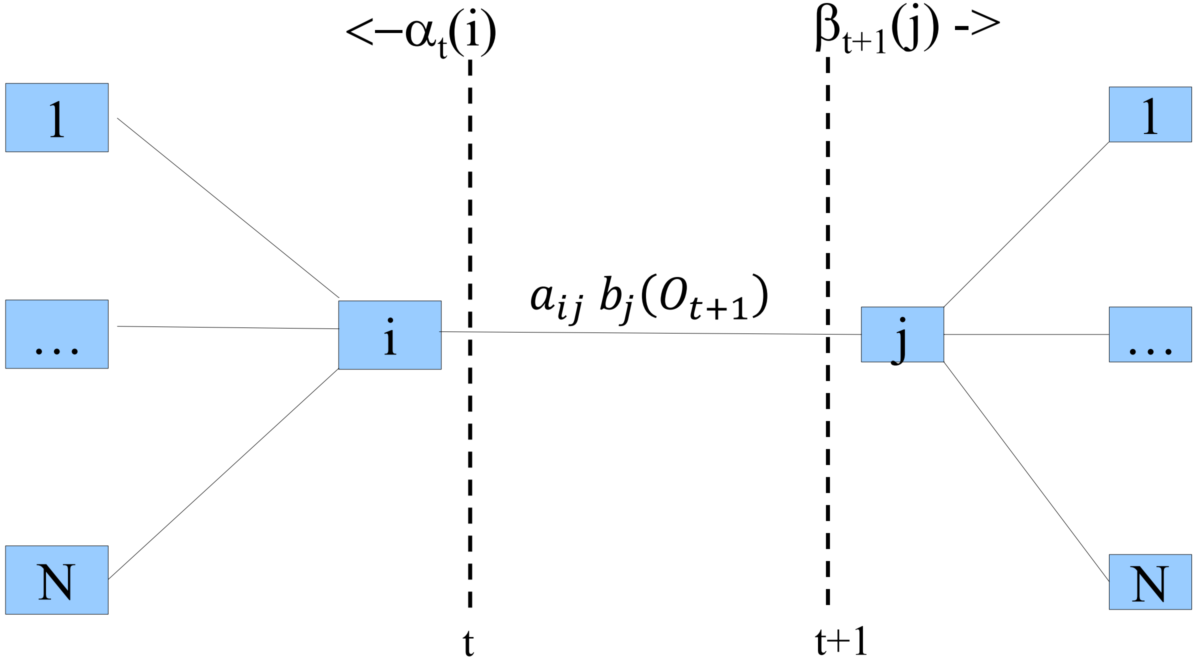
1. Initialize $\lambda=(A,B,\pi)$
2. Compute $\alpha_t(i)$, $\beta_t(k)$, $\gamma_t(i,j)$ and $\gamma_t(i)$
3. Re-estimate the model $\lambda=(A,B,\pi)$
4. Repeat from 2 until $p(O|\lambda)$ converges



GAMMA CALCULATIONS:

1) **Di – Gamma Function**

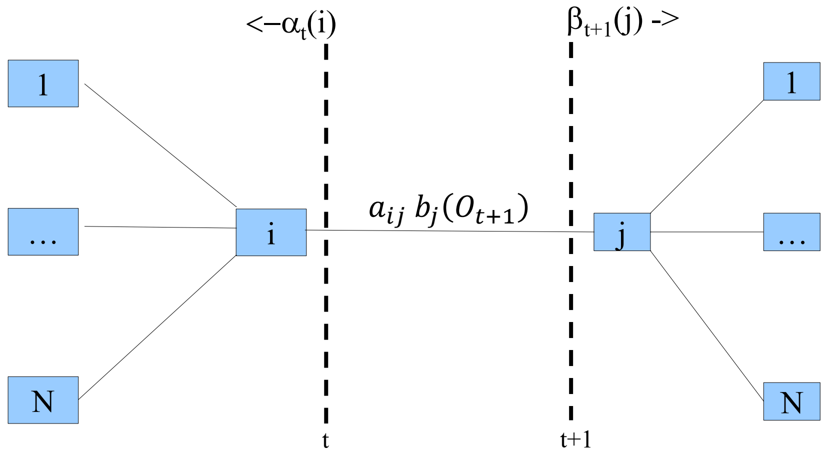
$$\gamma_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_T(i)} \qquad = p(X_t = i, X_{t+1} = j | O_{1:T}, \lambda)$$



GAMMA CALCULATIONS:

1) **Di – Gamma Function**

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

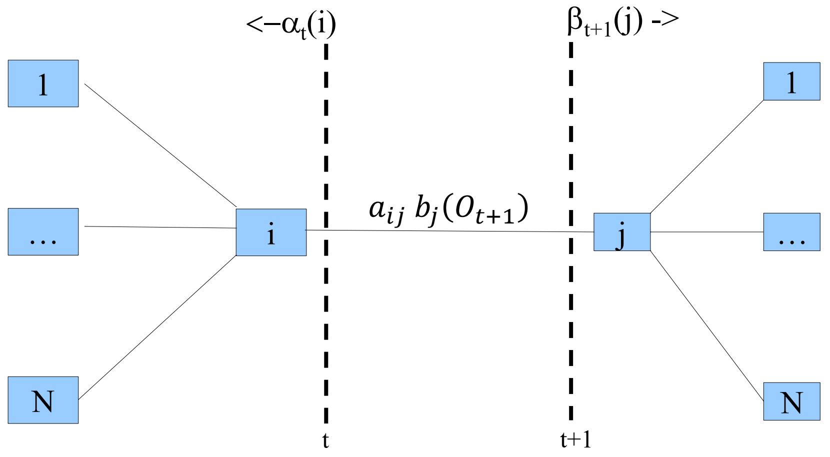


Interpretation: Given the entire observation sequence and current estimate of the HMM, what is the probability that at time (t) the hidden state is (X_t=i) & at time (t+1) the hidden state is (X_{t+1}=j)?

GAMMA CALCULATIONS:

1) Di – Gamma Function

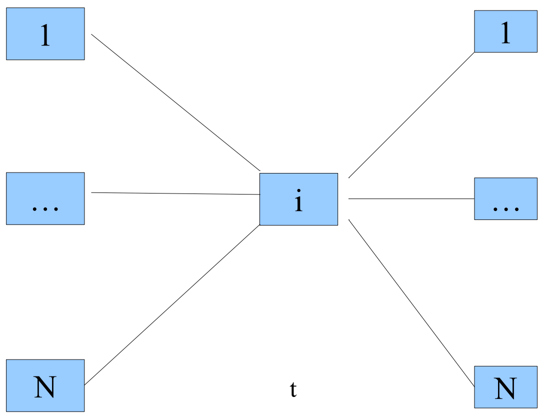
$$\gamma_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_T(i)} \qquad = p(X_t = i, X_{t+1} = j | O_{1:T}, \lambda)$$



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})

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



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)?

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

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

A) Transition estimates

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

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

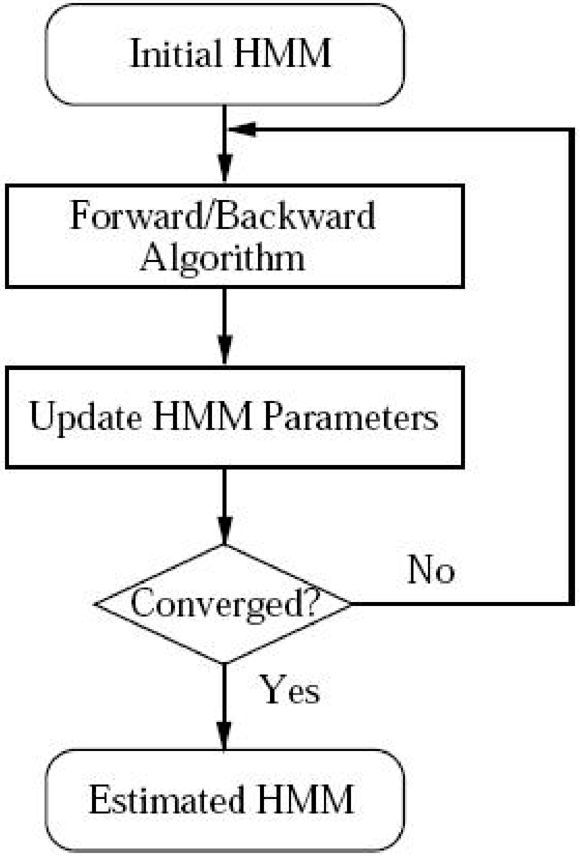
$$\pi_i = \gamma_1(i) \quad \forall \quad i = 1,\ldots,N$$

expected frequency of being in state i at t = 1

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



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 and considerations

- 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

  $a_{ij} \approx 1/N$

  $b_j(k) \approx 1/M$

  $\pi_i \approx 1/N$

  - Make sure that A,B and p 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

Three problems solved with HMMs

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

2. **Decoding**: 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
   → A and B matrices and π

   Baum-Welch algorithm

Additional (highly recommended) study material

- http://learn-ai.web.app/
  - Page developed by one of the previous course TAs with detailed (and visually appealing) explanations of today's lecture exercises + Forward algorithm
  - Visit the page before the HMM Tutorial sessions to consolidate

- Stamp Tutorial on Canvas containing the algorithmic implementations of these algorithms

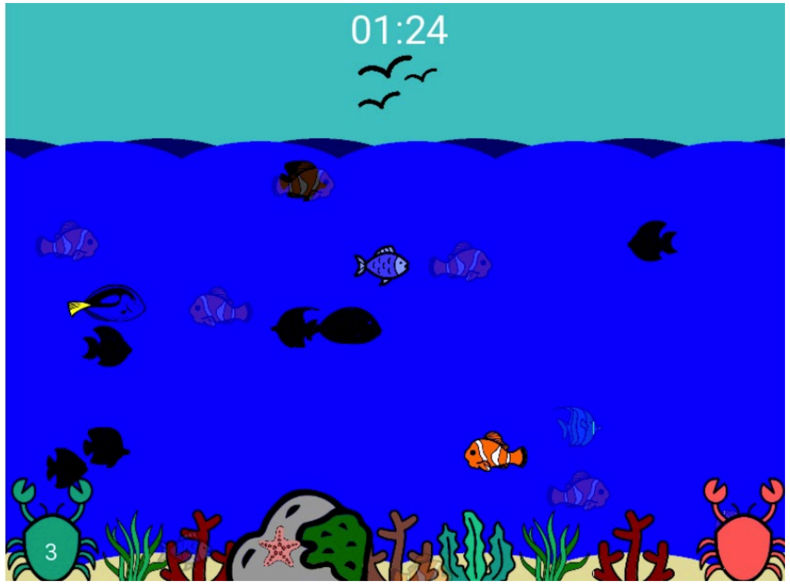
What's NEXT?

- HMM Tutorials
  - Check the schedule on Canvas for these and do not forget to attend next week (Monday)
  - Prepare tutorial exercise sheets (print or .doc)

- New quiz and last Lecture's quiz

- HMM assignment will be up after this lecture at 17:00pm!
  - Arm computers should now be able to run the assignment

- Lab sessions to start working on HMM assignment next week (Tuesday).



Motivation for HMM Assignment

- Solve an actual task, i.e., use the AI methods in a context
- Covers
  - Probabilistic reasoning
  - Machine learning
  - Decision making
  - Implementation
  - Testing and evaluation
- Work in pairs
  - Practice and implement HMMs
  - Using an existing implementation is not allowed

End of Probabilistic Reasoning Part 2/2 - HMMs