



香港科技大學  
THE HONG KONG  
UNIVERSITY OF SCIENCE  
AND TECHNOLOGY

COMP 5212  
Machine Learning  
Lecture 15

# Hidden Markov Models

Junxian He  
April 9, 2026

# Announcements

1. Mid-term exam grades are out, we will hold a paper-check session next week

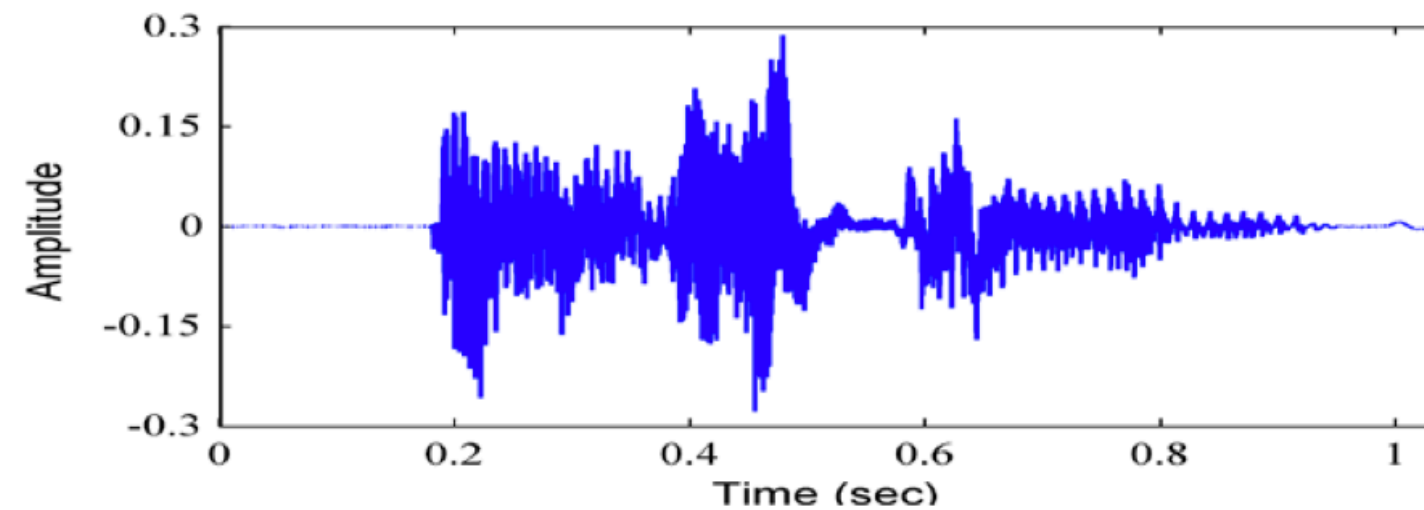
# i.i.d to sequential data

❑ So far we assumed independent, identically distributed data  $\{X_i\}_{i=1}^n \stackrel{iid}{\sim} p(\mathbf{X})$

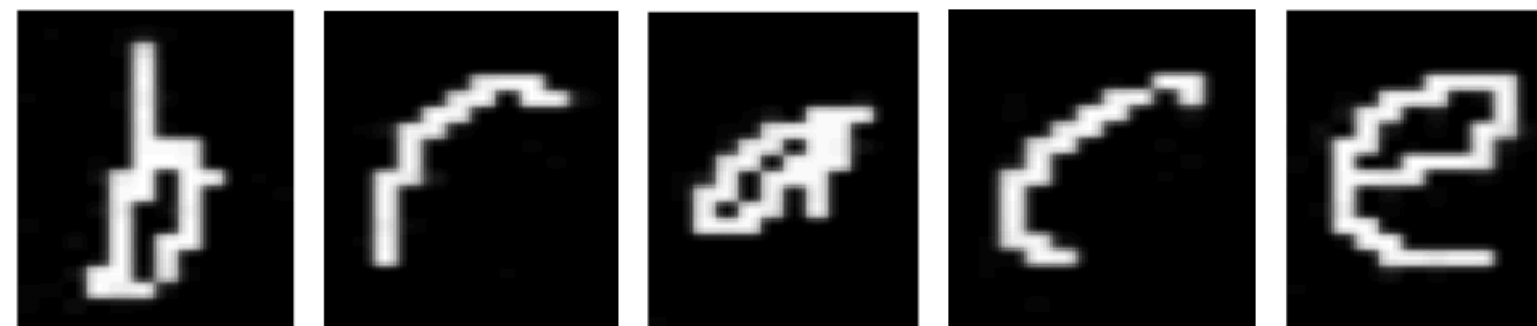
❑ Sequential (non i.i.d.) data

– Time-series data

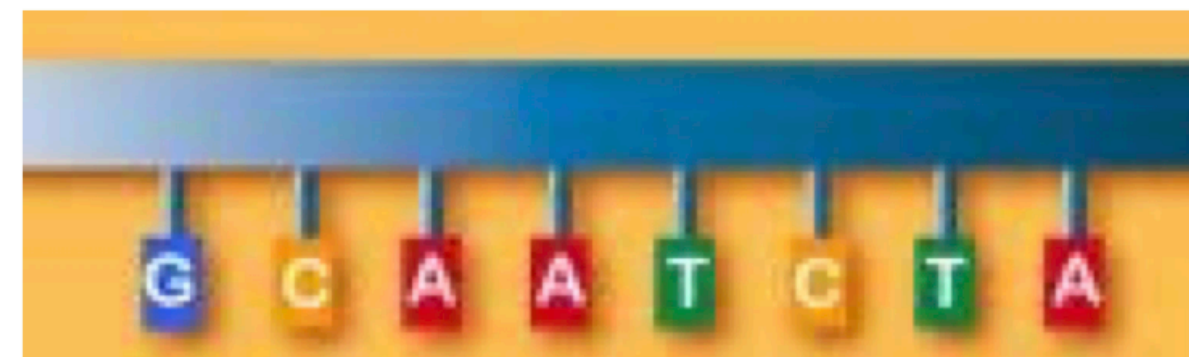
E.g. Speech



– Characters in a sentence



– Base pairs along a DNA strand



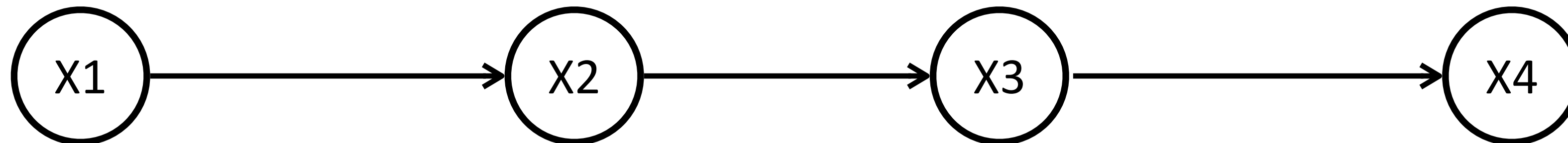
(Sequential data is still i.i.d on the sequence level)

# Review: Elimination Algorithm / Marginalization

$$P(h) = \sum_g \sum_f \sum_e \sum_d \sum_c \sum_b \sum_a P(a, b, c, d, e, f, g, h)$$



a naïve summation needs to enumerate over an exponential number of terms

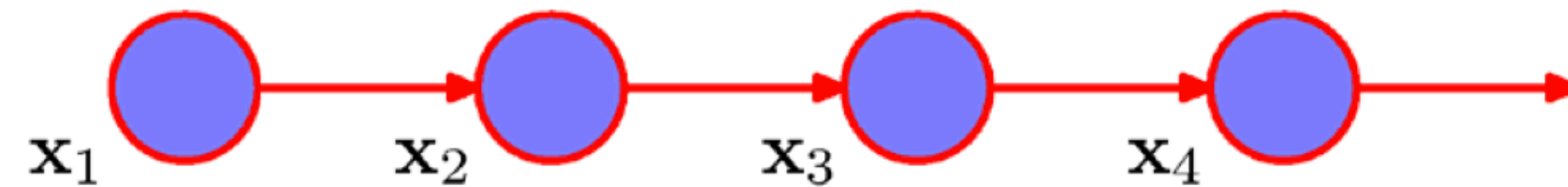


What if the random variables follow this chain structure?

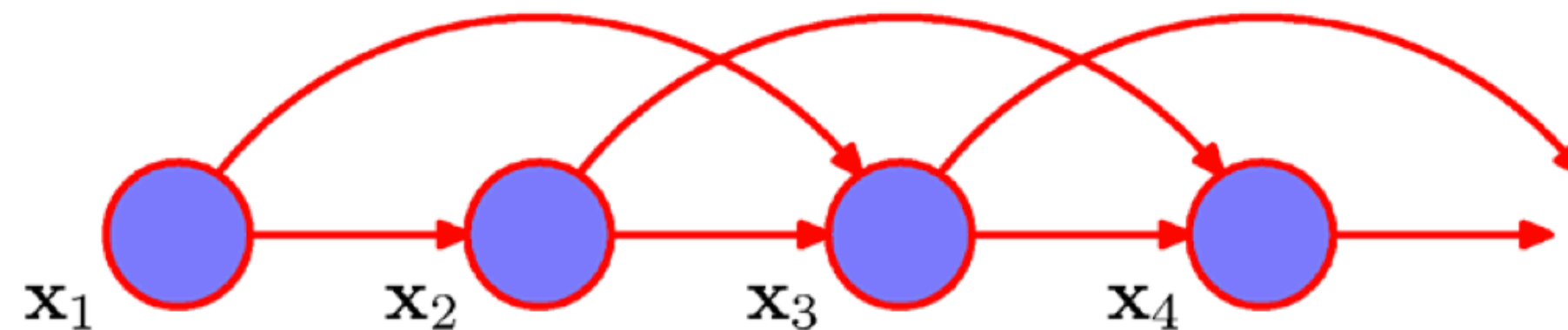
# Review: Markov Models

## □ Markov Assumption

1<sup>st</sup> order 
$$p(\mathbf{X}) = \prod_{i=1}^n p(X_i | X_{i-1})$$



2<sup>nd</sup> order 
$$p(\mathbf{X}) = \prod_{i=1}^n p(X_i | X_{i-1}, X_{i-2})$$



# Review: Markov Models

Homogeneous/stationary Markov model (probabilities don't depend on  $n$ )

## □ Markov Assumption

# parameters in  
stationary model  
K-ary variables

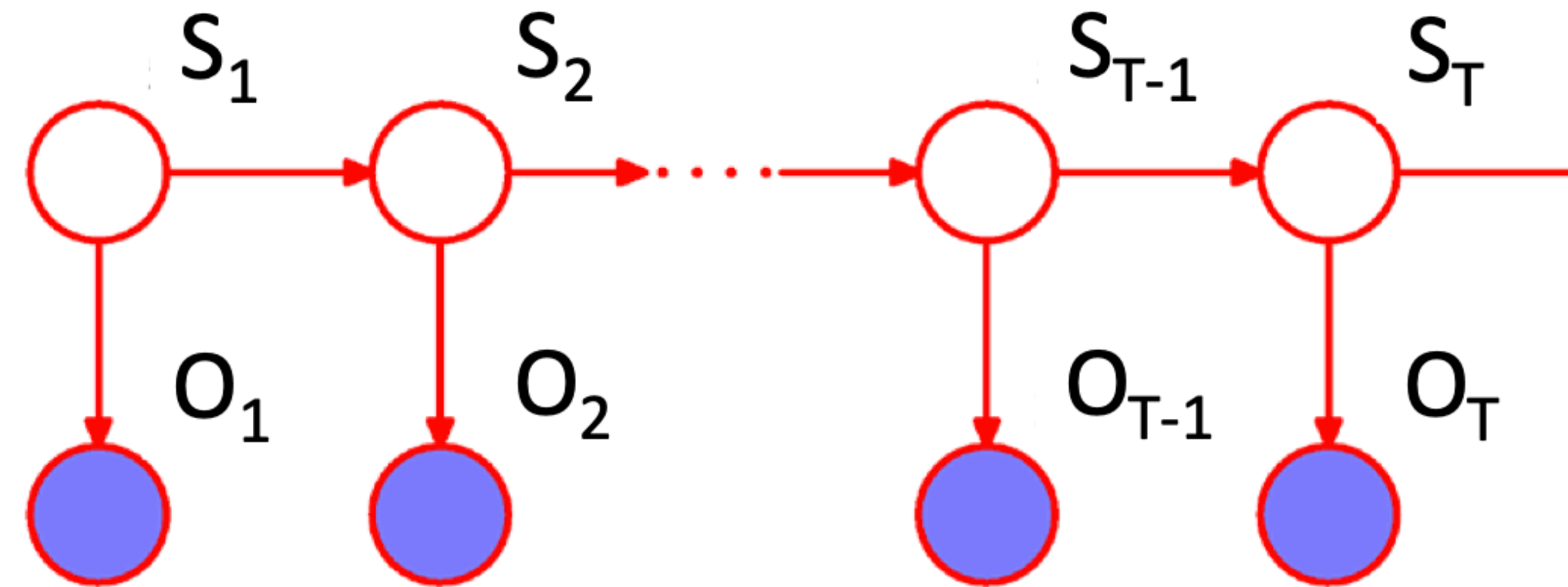
1<sup>st</sup> order  $p(\mathbf{X}) = \prod_{i=1}^n p(X_i | X_{i-1})$   $O(K^2)$

m<sup>th</sup> order  $p(\mathbf{X}) = \prod_{i=1}^n p(X_i | X_{i-1}, \dots, X_{i-m})$   $O(K^{m+1})$

n-1<sup>th</sup> order  $p(\mathbf{X}) = \prod_{i=1}^n p(X_i | X_{i-1}, \dots, X_1)$   $O(K^n)$

≡ no assumptions – complete (but directed) graph

# Review: Hidden Markov Models



Observation space

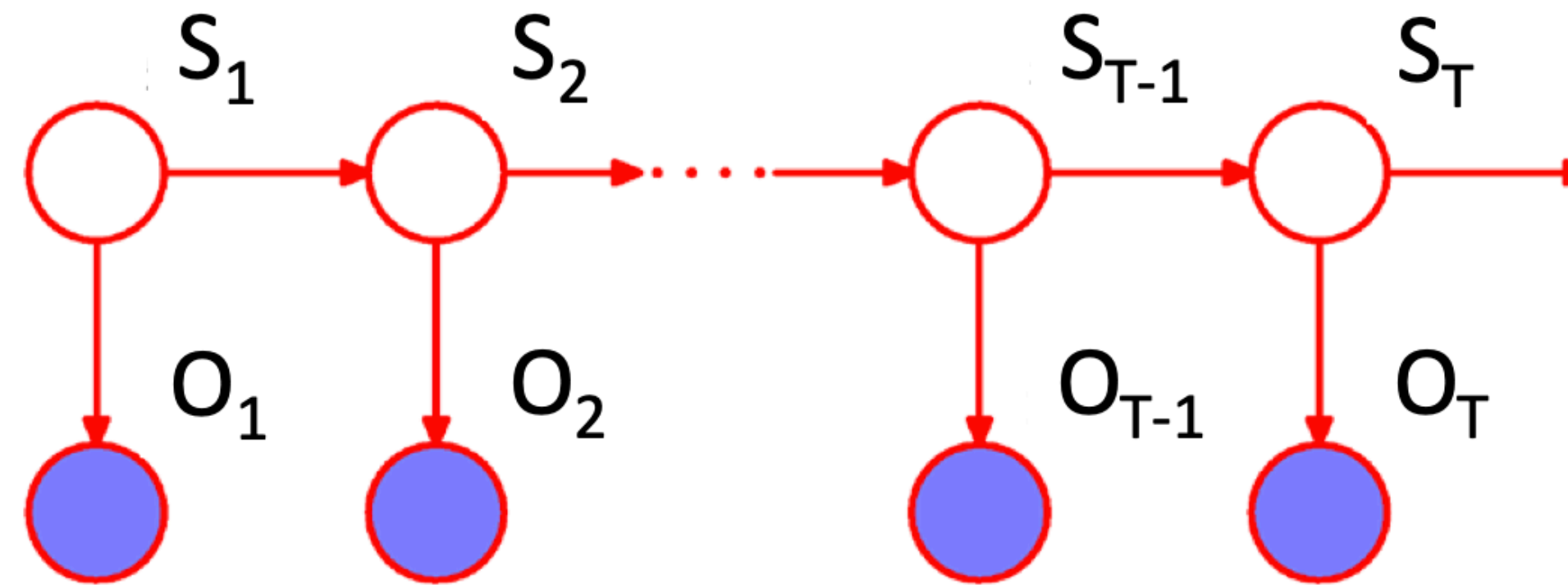
$$O_t \in \{y_1, y_2, \dots, y_K\}$$

Hidden states

$$S_t \in \{1, \dots, I\}$$



# Hidden Markov Models



$$p(S_1, \dots, S_T, O_1, \dots, O_T) = \prod_{t=1}^T p(O_t | S_t) \prod_{t=1}^T p(S_t | S_{t-1})$$

1<sup>st</sup> order Markov assumption on hidden states  $\{S_t\}$   $t = 1, \dots, T$   
(can be extended to higher order).

Is  $O_T$  and  $O_2$  independent?



# Hidden Markov Models

- Parameters – stationary/homogeneous markov model (independent of time  $t$ )

Initial probabilities

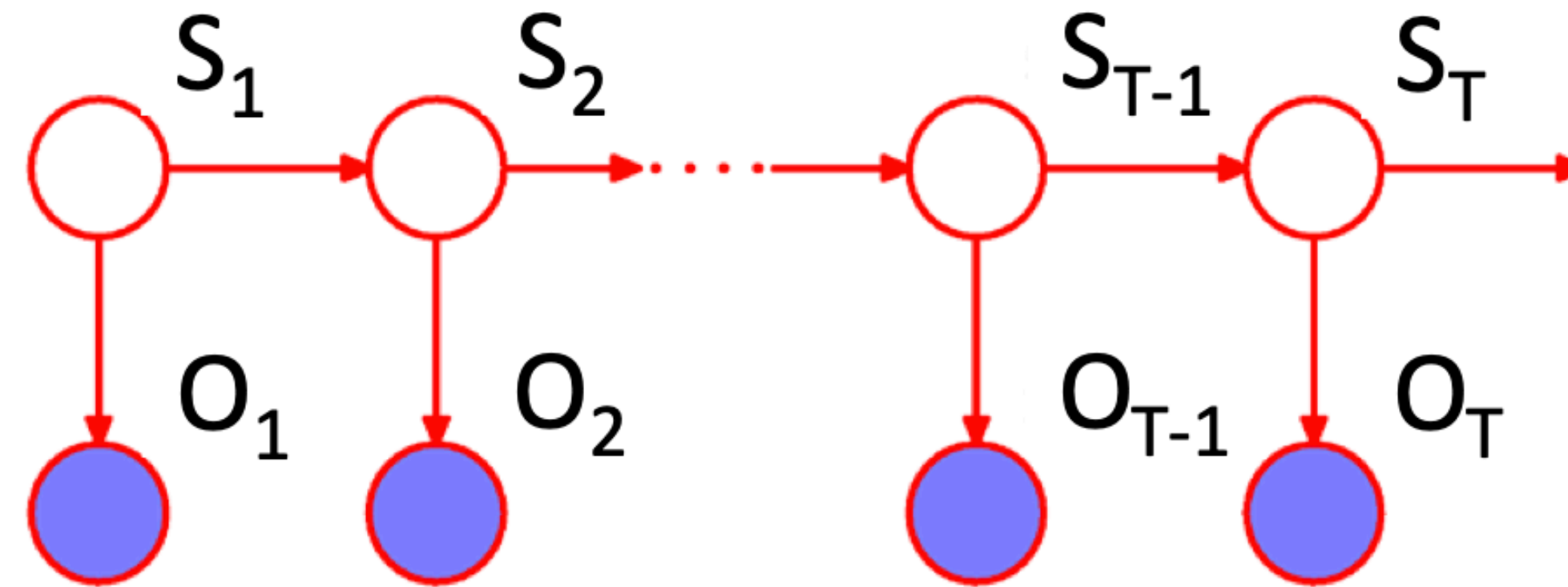
$$p(S_1 = i) = \pi_i$$

Transition probabilities

$$p(S_t = j | S_{t-1} = i) = p_{ij}$$

Emission probabilities

$$p(O_t = y | S_t = i) = q_i^y$$



$$p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) =$$

$$p(S_1) \prod_{t=2}^T p(S_t | S_{t-1}) \prod_{t=1}^T p(O_t | S_t)$$

# HMM Example

- The Dishonest Casino

A casino has two dices:

Fair dice

$$P(1) = P(2) = P(3) = P(5) = P(6) = 1/6$$

Loaded dice

$$P(1) = P(2) = P(3) = P(5) = 1/10$$

$$P(6) = 1/2$$

Casino player switches back-&-forth between fair and loaded die with 5% probability



# HMM Example

**GIVEN:** A sequence of rolls by the casino player

1245526462146146136136661664661636616366163616515615115146123562344

Question

1. How likely is the sequence given our model?

This is the evaluation problem in HMMs

2. What portion of the sequence was generated with the fair die, and what portion with the loaded die

This is the decoding question in HMMs

3. How “loaded” is the loaded die? How “fair” is the fair die? How often does the casino player change from fair to loaded, and back?

This is the learning question in HMMs



# Three Main Problems in HMMs

- **Evaluation** – Given HMM parameters & observation seqn  $\{O_t\}_{t=1}^T$   
find  $p(\{O_t\}_{t=1}^T | \theta)$  prob of observed sequence
- **Decoding** – Given HMM parameters & observation seqn  $\{O_t\}_{t=1}^T$   
find  $\arg \max_{s_1, \dots, s_T} p(\{S_t\}_{t=1}^T | \{O_t\}_{t=1}^T, \theta)$  most probable  
sequence of hidden states
- **Learning** – Given HMM with unknown parameters and  $\{O_t\}_{t=1}^T$   
observation sequence  
find  $\arg \max_{\theta} p(\{O_t\}_{t=1}^T | \theta)$  parameters that maximize  
likelihood of observed data

# HMM Algorithms

- **Evaluation** – What is the probability of the observed sequence? **Forward Algorithm**
- **Decoding** – What is the probability that the third roll was loaded given the observed sequence? **Forward-Backward Algorithm**
  - What is the most likely die sequence given the observed sequence? **Viterbi Algorithm**
- **Learning** – Under what parameterization is the observed sequence most probable? **Baum-Welch Algorithm (EM)**

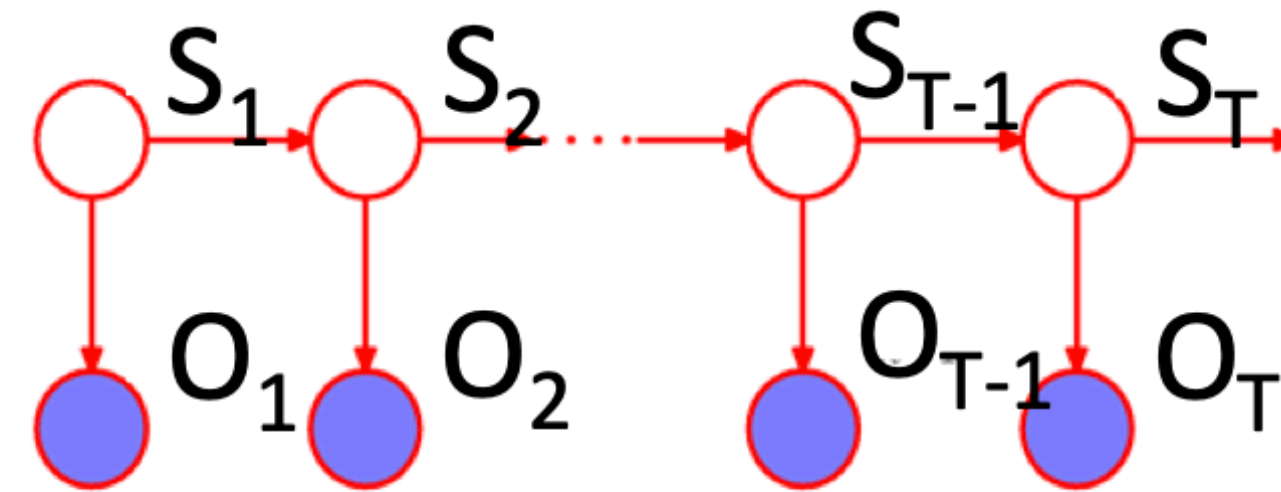
# Evaluation Problem

- Given HMM parameters  $p(S_1), p(S_t|S_{t-1}), p(O_t|S_t)$  & observation sequence  $\{O_t\}_{t=1}^T$

find probability of observed sequence

$$p(\{O_t\}_{t=1}^T) = \sum_{S_1, \dots, S_T} p(\{O_t\}_{t=1}^T, \{S_t\}_{t=1}^T)$$

$$= \sum_{S_1, \dots, S_T} p(S_1) \prod_{t=2}^T p(S_t|S_{t-1}) \prod_{t=1}^T p(O_t|S_t)$$



requires summing over all possible hidden state values at all times –  $K^T$  exponential # terms!

# Forward Probability

$$p(\{O_t\}_{t=1}^T) = \sum_k p(\{O_t\}_{t=1}^T, S_T = k) = \sum_k \alpha_T^k$$

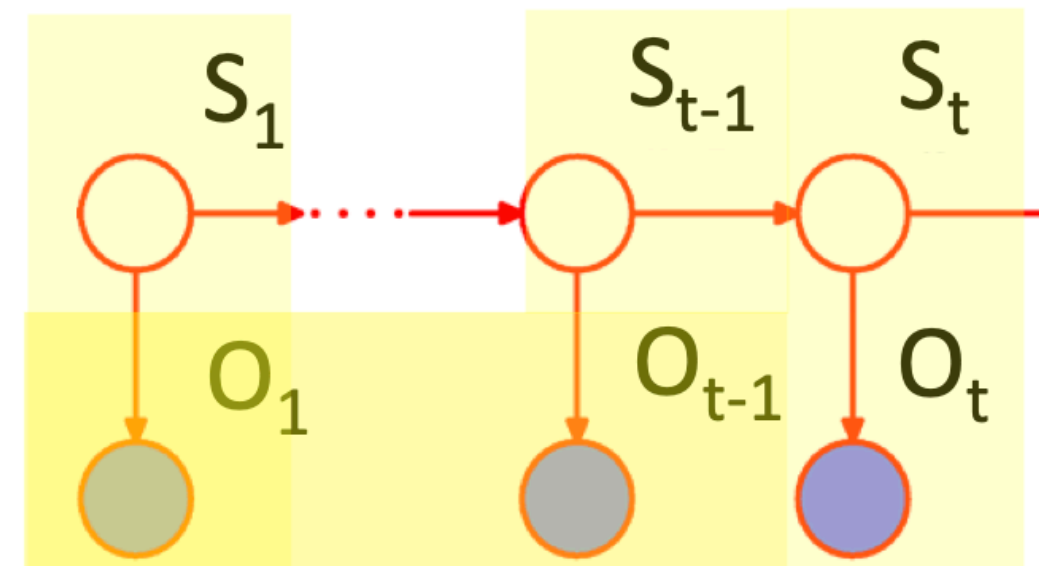
Compute forward probability  $\alpha_t^k$  recursively over  $t$

$$\alpha_t^k := p(O_1, \dots, O_t, S_t = k)$$

Introduce  $S_{t-1}$

Chain rule

Markov assumption



$$= p(O_t | S_t = k) \sum_i \alpha_{t-1}^i p(S_t = k | S_{t-1} = i)$$



# Forward Algorithm

Can compute  $\alpha_t^k$  for all  $k, t$  using dynamic programming:

- Initialize:  $\alpha_1^k = p(O_1 | S_1 = k) p(S_1 = k)$  for all  $k$

- Iterate: for  $t = 2, \dots, T$

$$\alpha_t^k = p(O_t | S_t = k) \sum_i \alpha_{t-1}^i p(S_t = k | S_{t-1} = i) \quad \text{for all } k$$

- Termination:  $p(\{O_t\}_{t=1}^T) = \sum_k \alpha_T^k$

You will try this in your HW

Can we do in the backward direction?

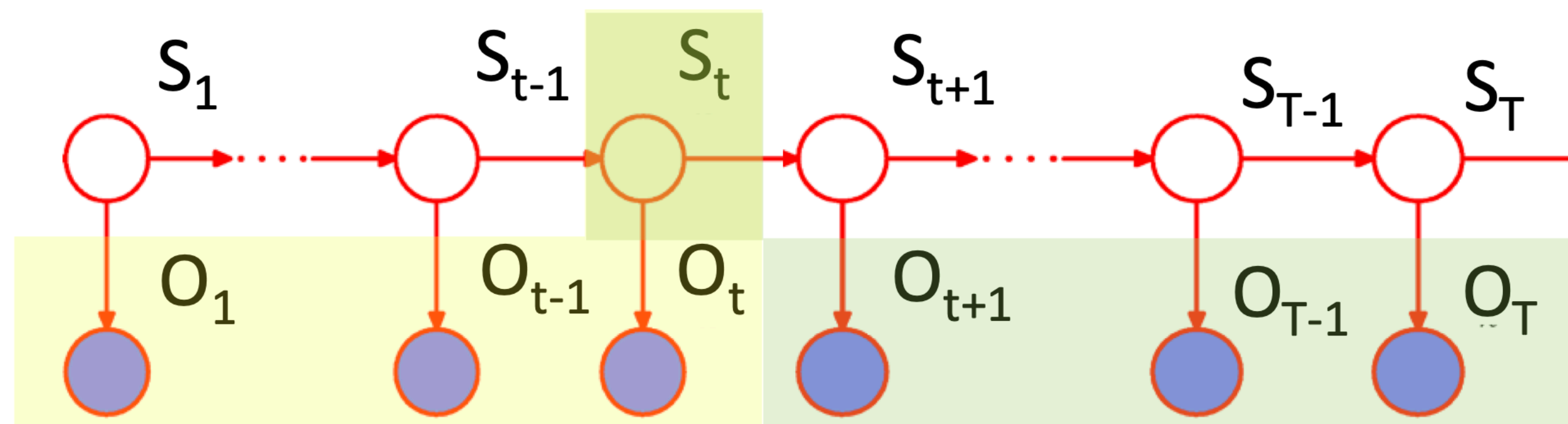
# Decoding Problem 1

- Given HMM parameters  $p(S_1), p(S_t|S_{t-1}), p(O_t|S_t)$  & observation sequence  $\{O_t\}_{t=1}^T$

find probability that hidden state at time t was k  $p(S_t = k | \{O_t\}_{t=1}^T)$

$$\begin{aligned}
 p(S_t = k, \{O_t\}_{t=1}^T) &= p(O_1, \dots, O_t, S_t = k, O_{t+1}, \dots, O_T) \\
 &= \underbrace{p(O_1, \dots, O_t, S_t = k)}_{\alpha_t^k} \underbrace{p(O_{t+1}, \dots, O_T | S_t = k)}_{\beta_t^k}
 \end{aligned}$$

Compute recursively



# Backward Algorithm

Can compute  $\beta_t^k$  for all  $k, t$  using dynamic programming:

- Initialize:  $\beta_T^k = 1$  for all  $k$  **Why this initialization?**

- Iterate: for  $t = T-1, \dots, 1$

$$\beta_t^k = \sum_i p(S_{t+1} = i | S_t = k) p(O_{t+1} | S_{t+1} = i) \beta_{t+1}^i \quad \text{for all } k$$

- Termination:  $p(S_t = k, \{O_t\}_{t=1}^T) = \alpha_t^k \beta_t^k$

**Can we compute  $\beta$  in a forward manner?**

$$p(S_t = k | \{O_t\}_{t=1}^T) = \frac{p(S_t = k, \{O_t\}_{t=1}^T)}{p(\{O_t\}_{t=1}^T)} = \frac{\alpha_t^k \beta_t^k}{\sum_i \alpha_t^i \beta_t^i}$$

# Most Likely State vs. Most Likely Sequence

- Most likely state assignment at time t

$$\arg \max_k p(S_t = k | \{O_t\}_{t=1}^T) = \arg \max_k \alpha_t^k \beta_t^k$$

E.g. Which die was most likely used by the casino in the third roll given the observed sequence?

- Most likely assignment of state sequence

$$\arg \max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T | \{O_t\}_{t=1}^T)$$

Are the solutions the same?

# Decoding Problem 2

- Given HMM parameters  $p(S_1), p(S_t|S_{t-1}), p(O_t|S_t)$  & observation sequence  $\{O_t\}_{t=1}^T$

find most likely assignment of state sequence

$$\begin{aligned} \arg \max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T | \{O_t\}_{t=1}^T) &= \arg \max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) \\ &= \arg \max_k \max_{\{S_t\}_{t=1}^{T-1}} p(S_T = k, \underbrace{\{S_t\}_{t=1}^{T-1}, \{O_t\}_{t=1}^T}_{V_T^k}) \end{aligned}$$

Compute recursively

$V_T^k$  - probability of most likely sequence of states ending at state  $S_T = k$

# Viterbi Decoding

$$\max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) = \max_k V_T^k$$

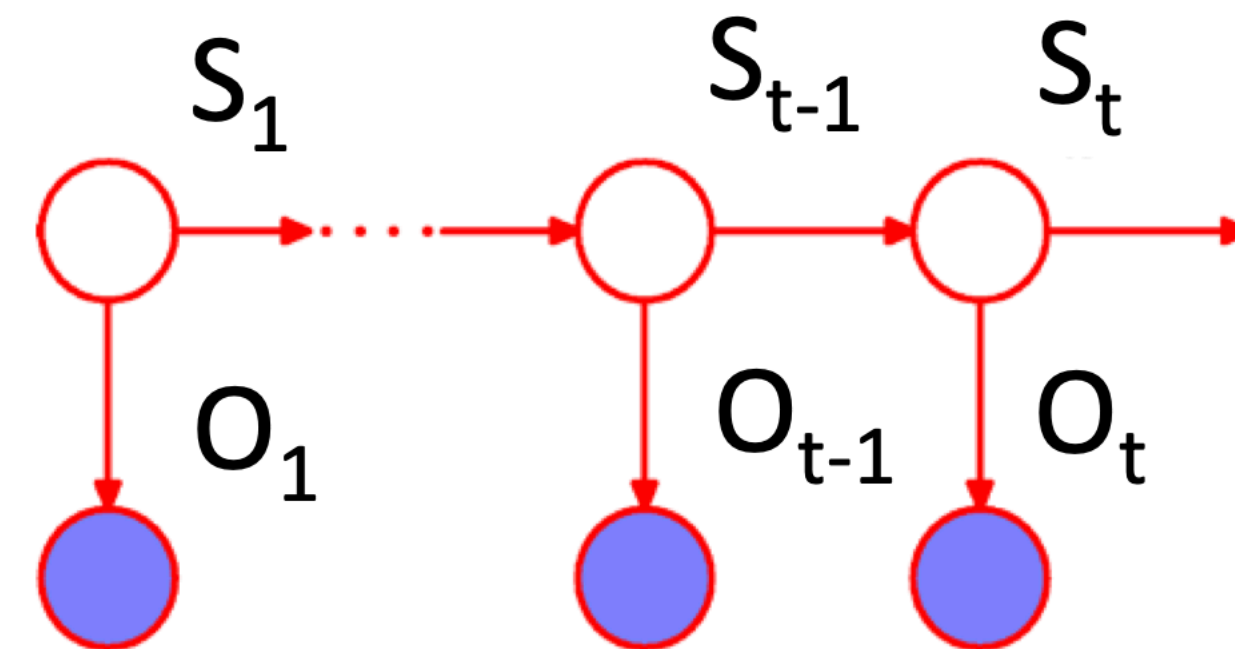
Compute probability  $V_t^k$  recursively over t

$$V_t^k := \max_{S_1, \dots, S_{t-1}} p(S_t = k, S_1, \dots, S_{t-1}, O_1, \dots, O_t)$$

·  
·  
·

Bayes rule

Markov assumption



$$= p(O_t | S_t = k) \max_i p(S_t = k | S_{t-1} = i) V_{t-1}^i$$



# Viterbi Algorithm

Can compute  $V_t^k$  for all  $k, t$  using dynamic programming:

- Initialize:  $V_1^k = p(O_1|S_1=k)p(S_1 = k)$  for all  $k$

- Iterate: for  $t = 2, \dots, T$

$$V_t^k = p(O_t|S_t = k) \max_i p(S_t = k|S_{t-1} = i) V_{t-1}^i \quad \text{for all } k$$

- Termination:  $\max_{\{S_t\}_{t=1}^T} p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) = \max_k V_T^k$

Traceback:

$$S_T^* = \arg \max_k V_T^k$$

$$S_{t-1}^* = \arg \max_i p(S_t^*|S_{t-1} = i) V_{t-1}^i$$

Can we do in the  
backward direction?



# Computational Complexity

- What is the running time for Forward, Backward, Viterbi?

$$\alpha_t^k = q_k^{O_t} \sum_i \alpha_{t-1}^i p_{i,k}$$

$$\beta_t^k = \sum_i p_{k,i} q_i^{O_{t+1}} \beta_{t+1}^i$$

$$V_t^k = q_k^{O_t} \max_i p_{i,k} V_{t-1}^i$$

$O(K^2T)$  linear in  $T$  instead of  $O(K^T)$  exponential in  $T$ !

# Learning with EM

- Start with random initialization of parameters
- **E-step** – Fix parameters, find expected state assignments

$$\gamma_i(t) = p(S_t = i | \mathbf{O}, \theta) = \frac{\alpha_t^i \beta_t^i}{\sum_j \alpha_t^j \beta_t^j} \quad \mathbf{O} = \{O_t\}_{t=1}^T$$

## Forward-Backward algorithm

$$\begin{aligned} \xi_{ij}(t) &= p(S_{t-1} = i, S_t = j | \mathbf{O}, \theta) \\ &= \frac{p(S_{t-1} = i | \mathbf{O}, \theta) p(S_t = j, O_t, \dots, O_T | S_{t-1} = i, \theta)}{p(O_t, \dots, O_T | S_{t-1} = i, \theta)} \\ &= \frac{\gamma_i(t-1) p_{ij} q_j^{O_t} \beta_t^j}{\beta_{t-1}^i} \end{aligned}$$

You will derive the EM  
in your HW

# If you still remember why we do EM in the first place...

$$\begin{aligned}\ell(\phi, \mu, \Sigma) &= \sum_{i=1}^n \log p(x^{(i)}; \phi, \mu, \Sigma) \\ &= \sum_{i=1}^n \log \sum_{z^{(i)}=1}^k p(x^{(i)}|z^{(i)}; \mu, \Sigma)p(z^{(i)}; \phi).\end{aligned}$$

Wait, HMM has closed-form likelihood?

1. Intractable (no closed-form for the solution)
2. Large variance in gradient descent

Expectation Maximization is to address the MLE optimization problem

Can we do MLE directly for HMM using gradient descent, without EM?

**Thank You!**