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

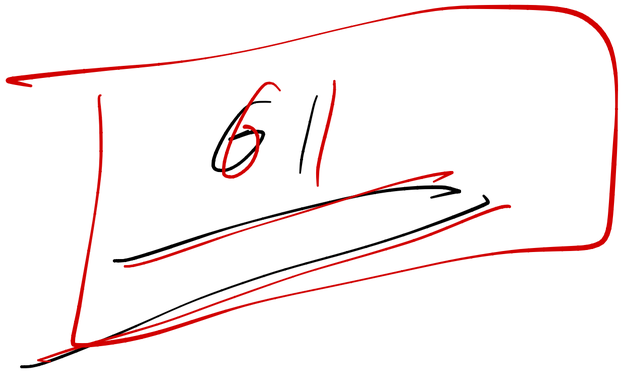
COMP 5212
Machine Learning
Lecture 15

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

i.i.d to sequential data

- So far we assumed independent, identically distributed data

$$\{X_i\}_{i=1}^{n} \overset{iid}{\sim} p(\mathbf{X})$$

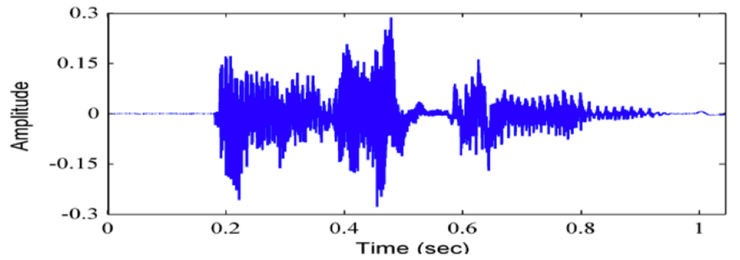
i.i.d to sequential data

- So far we assumed independent, identically distributed data

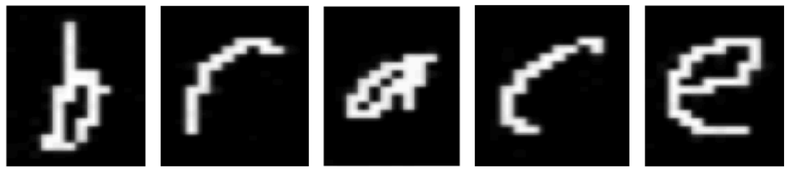
$$\{X_i\}_{i=1}^{n} \overset{iid}{\sim} p(\mathbf{X})$$

- Sequential (non i.i.d.) data
  - Time-series data

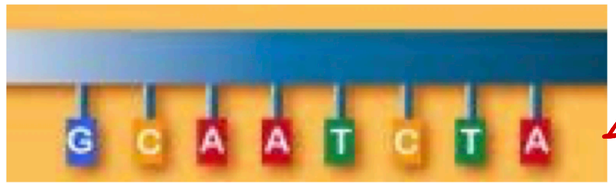
    E.g. Speech



  - Characters in a sentence



  - Base pairs along a DNA strand



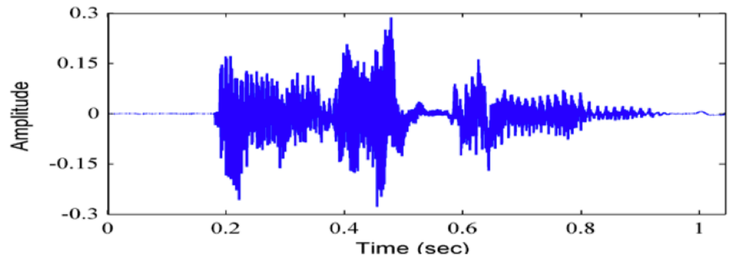
i.i.d to sequential data

- So far we assumed independent, identically distributed data

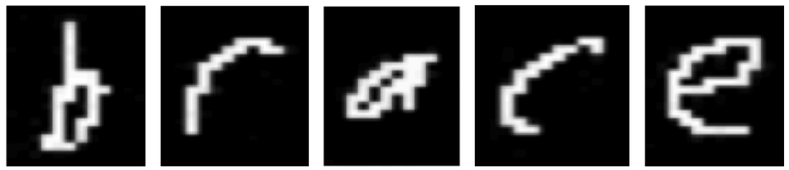
$$\{X_i\}_{i=1}^{n} \overset{iid}{\sim} p(\mathbf{X})$$

- Sequential (non i.i.d.) data
  - Time-series data

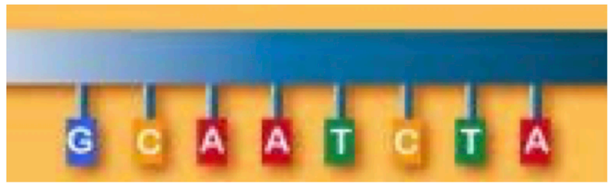
    E.g. Speech



  - 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

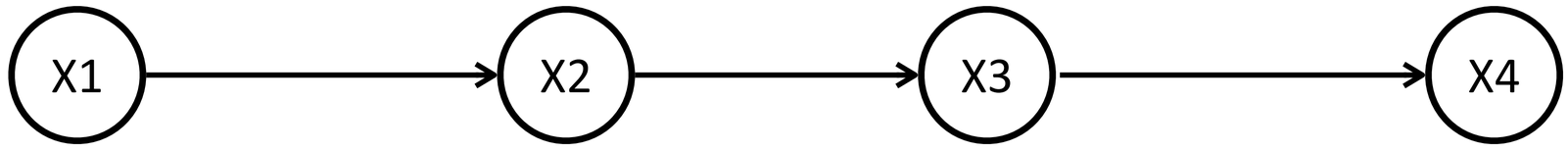
$$P(h) = \sum_{g} \sum_{f} \sum_{e} \sum_{d} \sum_{c} \sum_{b} \sum_{a} P(a,b,c,d,e,f,g,h)$$

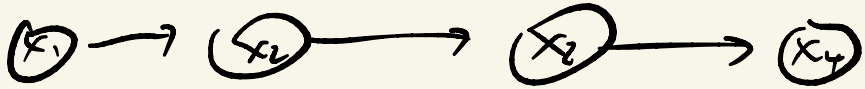
a naive summation needs to enumerate over an exponential number of terms

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





$$P(x_4)$$

$$\sum_{x_1} \sum_{x_2} \sum_{x_3} P(x_2|x_1)\, P(x_1)\, P(x_3|x_2)\, P(x_4|x_3)$$

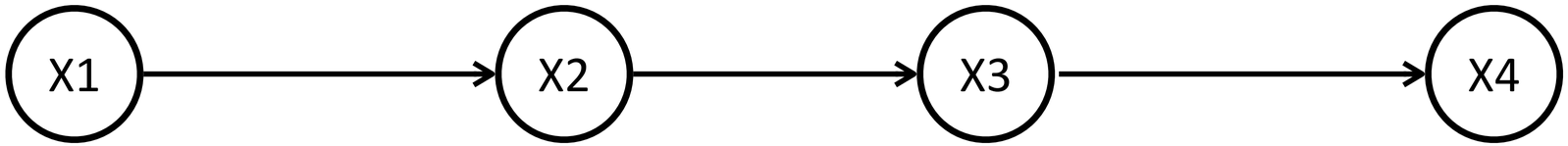
$$\sum_{x_2} \sum_{x_3} P(x_3|x_2) P(x_4|x_3) \left[\sum_{x_1} P(x_2|x_1) P(x_1) \right] = P(x_2)$$

$$f(x_2)$$

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?

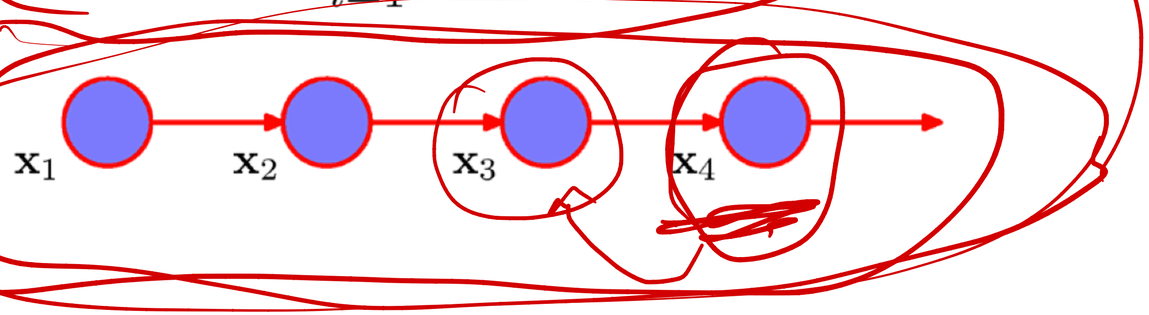
Review: Markov Models

Review: Markov Models

- Markov Assumption

  1st order

$$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1})$$

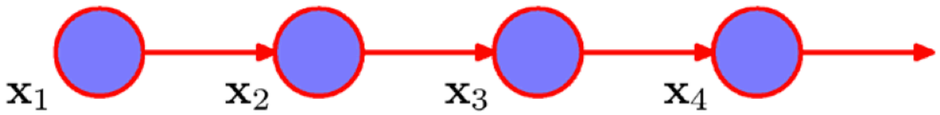


Review: Markov Models

- Markov Assumption

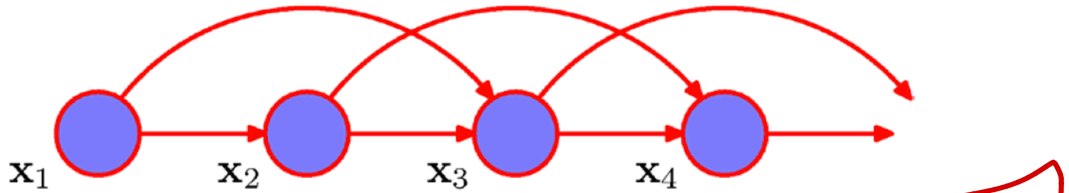
1st order

$$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1})$$



2nd order

$$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1}, X_{n-2})$$



Review: Markov Models

Review: Markov Models

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

Review: Markov Models

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

- Markov Assumption

  1st order $$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1})$$

parameters in stationary model
K-ary variables
O(K²)

K^2

$P(X_n|X_{n-1})$

$P(X_2=3|X_1=0)$

$P(X_{10}=3|X_9=0)$

Review: Markov Models

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

- Markov Assumption

parameters in stationary model
K-ary variables

1st order $p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1})$ $O(K^2)$

mth order $p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1},\ldots,X_{n-m})$ $O(K^{m+1})$

$O(K^{m+1})$

Review: Markov Models

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

- Markov Assumption

1st order $$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1})$$

mth order $$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1},\ldots,X_{n-m})$$

n-1th order $$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1},\ldots,X_1)$$

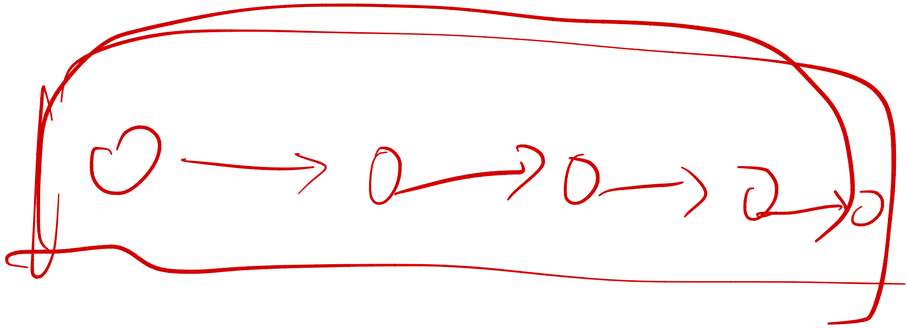
≡ no assumptions – complete (but directed) graph

parameters in stationary model
K-ary variables

$O(K^2)$

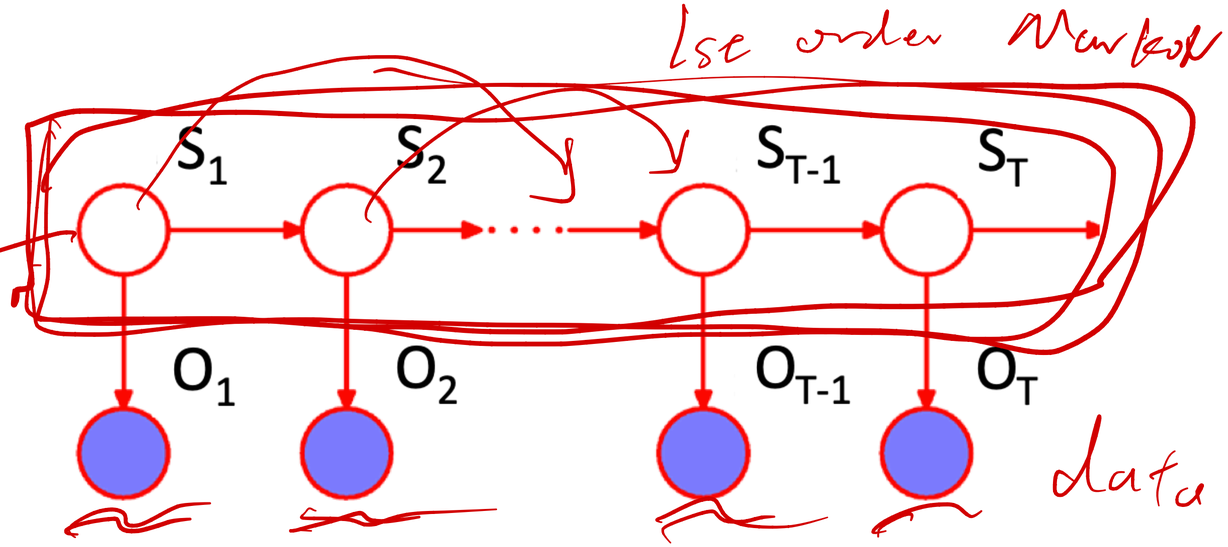
$O(K^{m+1})$

$O(K^n)$



exponential growth

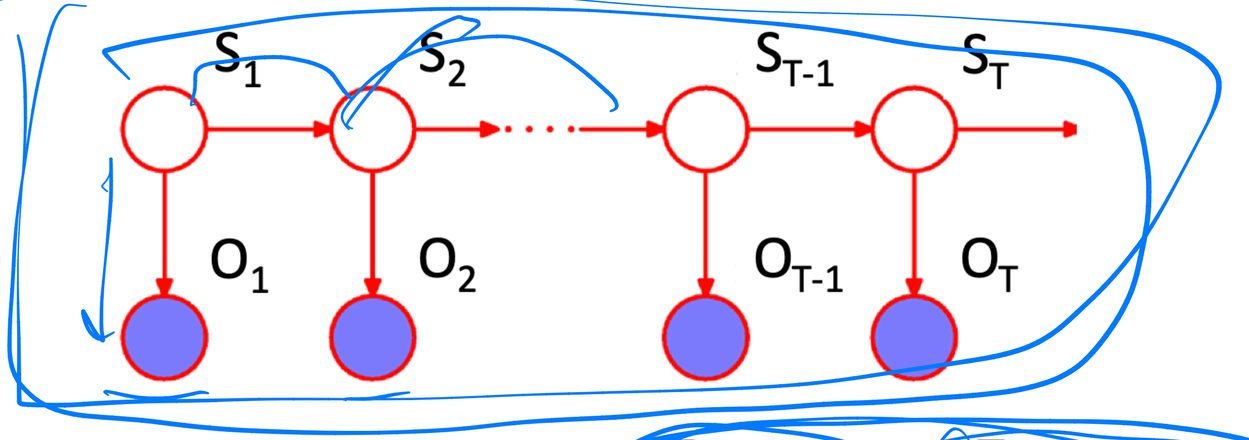
Review: Hidden Markov Models



Observation space	$O_t \in \{y_1, y_2, \ldots, y_K\}$
Hidden states	$S_t \in \{1, \ldots, I\}$

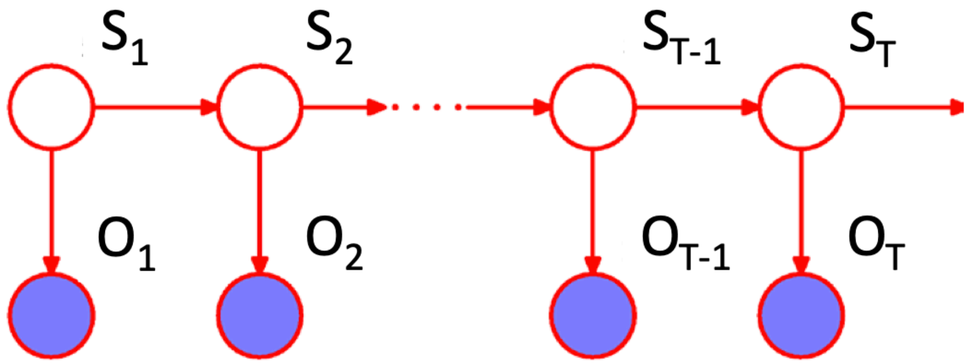
Hidden Markov Models

Hidden Markov Models



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

Hidden Markov Models

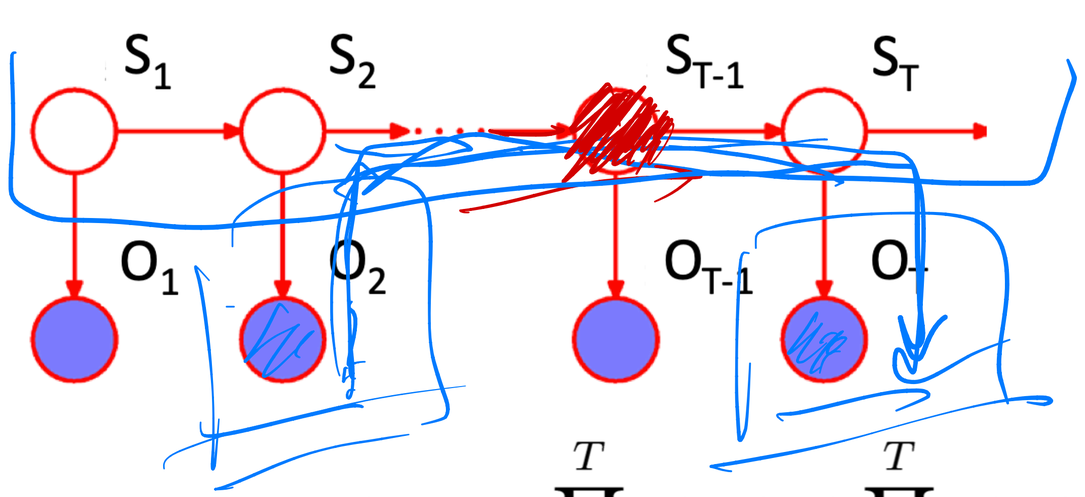


$$p(S_1, \ldots, S_T, O_1, \ldots, O_T) = \prod_{t=1}^{T} p(O_t|S_t) \prod_{t=1}^{T} p(S_t|S_{t-1})$$

1st order Markov assumption on hidden states $\{S_t\}$ t = 1, …, T

(can be extended to higher order).

Hidden Markov Models



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

1st order Markov assumption on hidden states $\{S_t\}$ t = 1, …, T

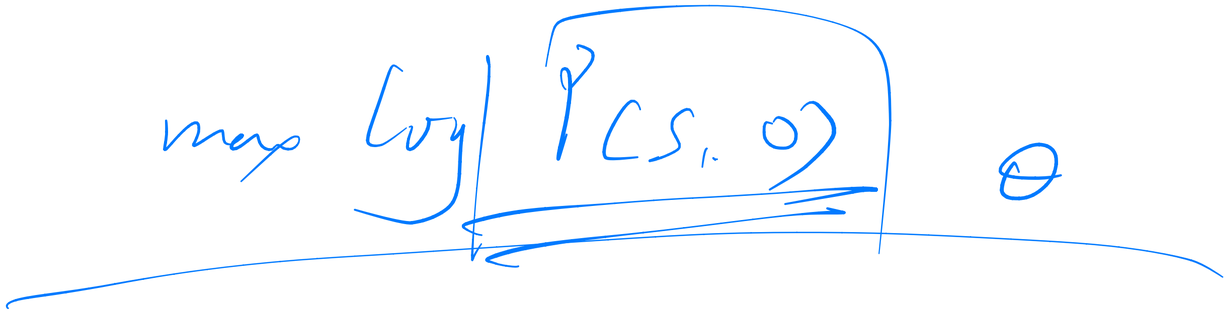
(can be extended to higher order).

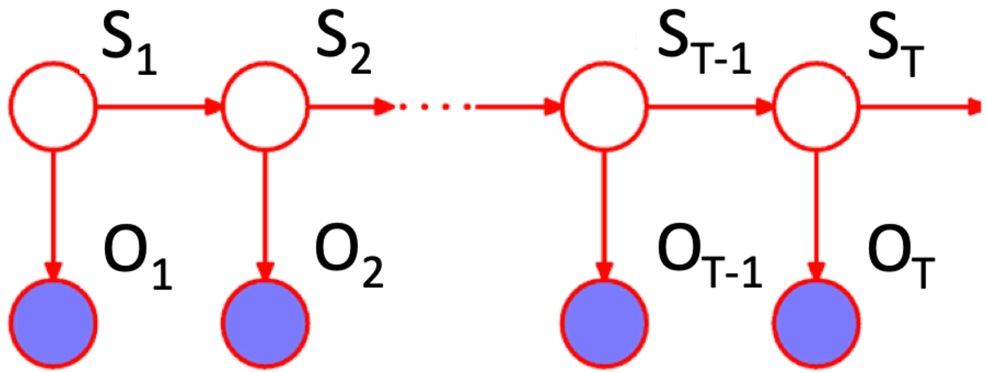
Is O_T and O_2 independent?

buyes bull

O_T and O_2 independent conditioned on S_{T-1}

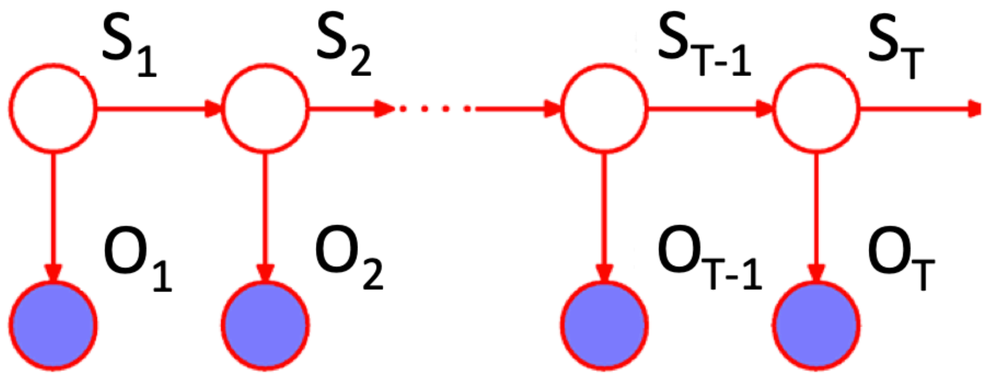
Hidden Markov Models





Hidden Markov Models

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

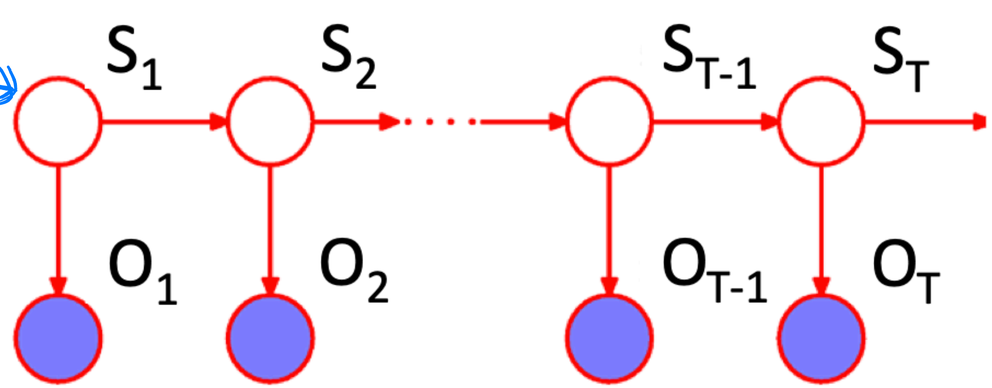


Hidden Markov Models

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

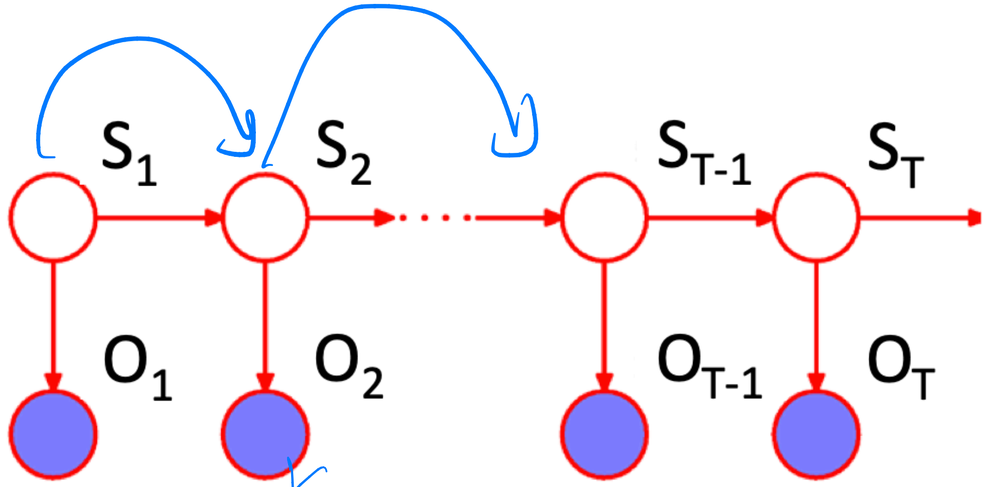
Initial probabilities

$p(S_1 = i) = \pi_i$



Hidden Markov Models

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



Initial probabilities

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

Transition probabilities

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

Hidden Markov Models

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

Initial probabilities

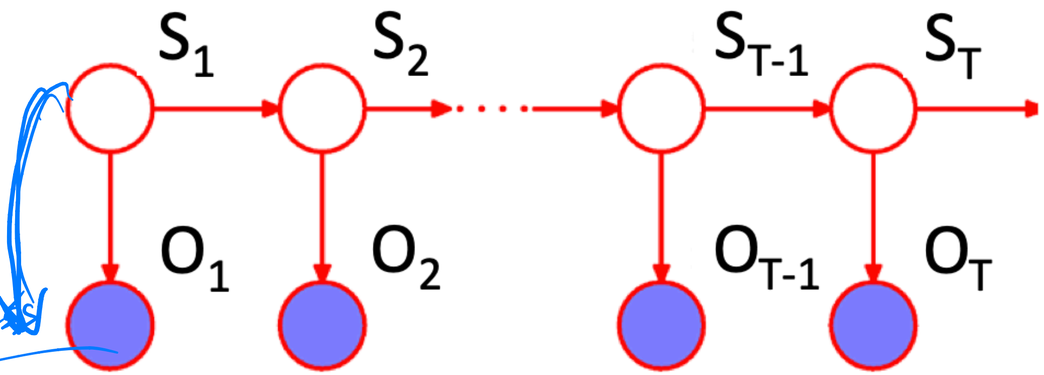
$$p(S_1 = i) = \pi_i$$

Transition probabilities

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

Emission probabilities

$$p(O_t = y \mid S_t = i) = q_i^y$$



Hidden Markov Models

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

Initial probabilities

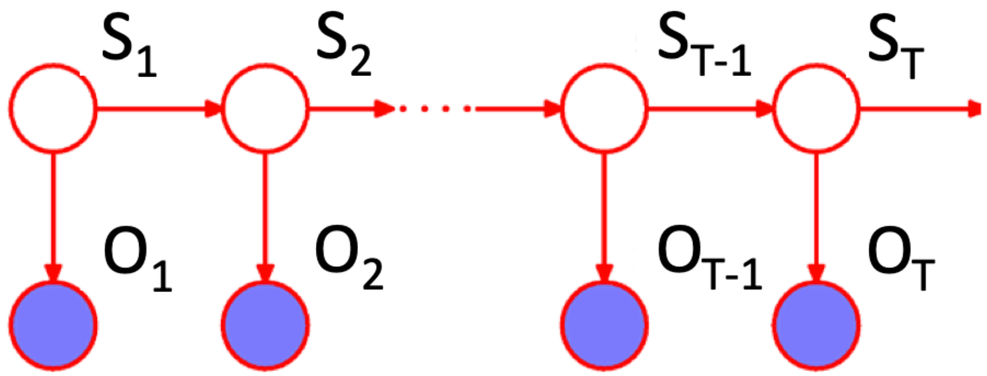
$p(S_1 = i) = \pi_i$

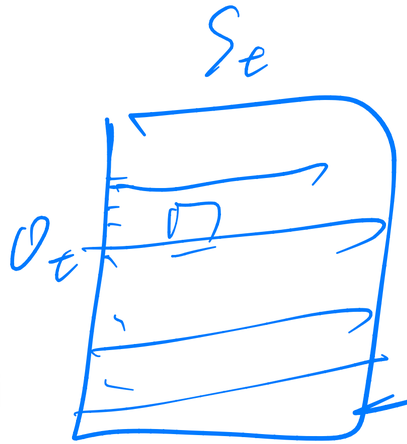
Transition probabilities

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

Emission probabilities

$p(O_t = y \mid S_t = i) = q_i^y$





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

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

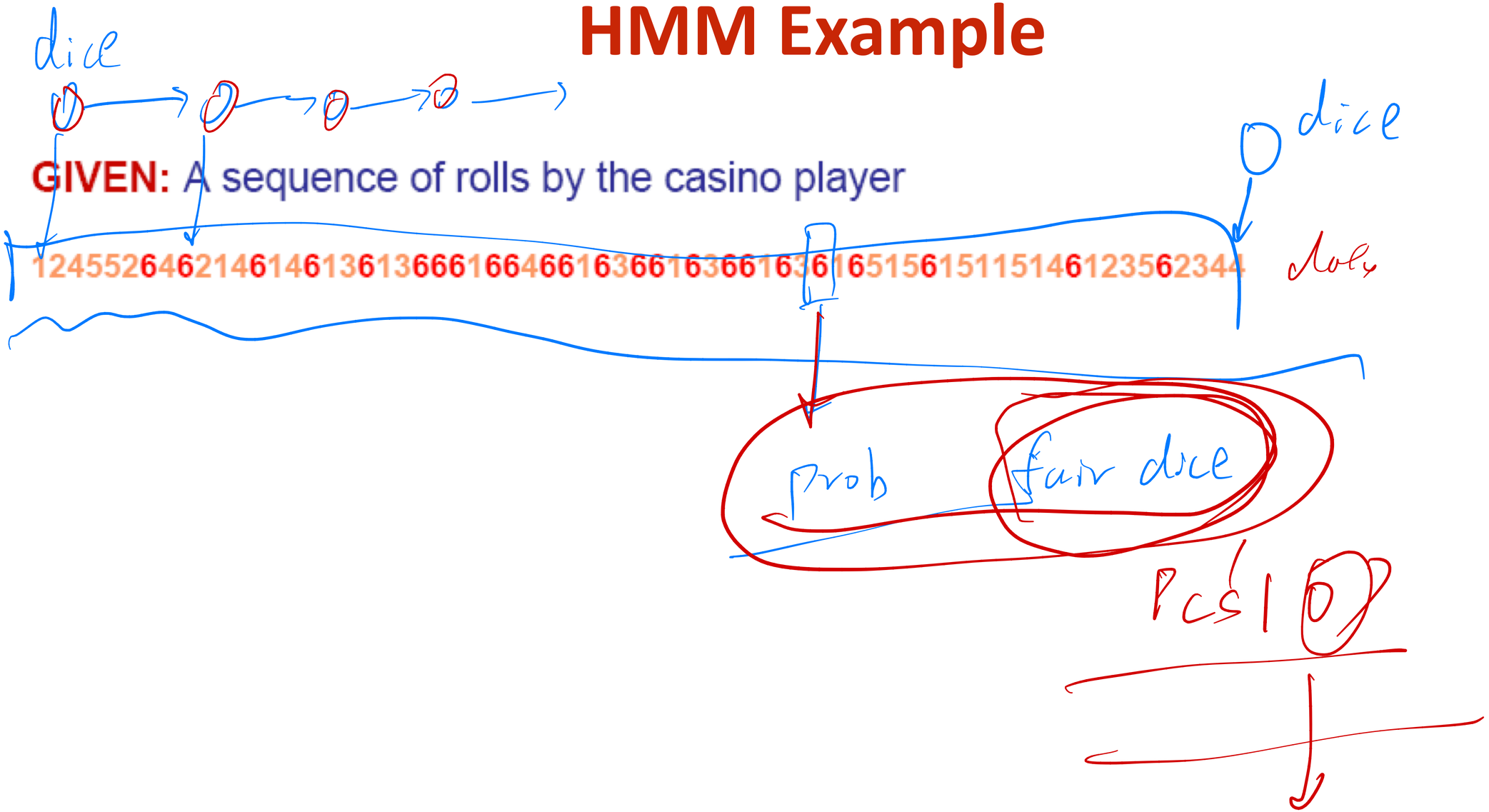
P(1) = P(2) = P(3) = P(5) = 1/10

P(6) = ½

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

HMM Example

HMM Example


dice
GIVEN: A sequence of rolls by the casino player
12455264621461461361366616646616366163661636165156151151461235623 44
dice
prob
fair dice
P(S|O)

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

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

HMM Example

GIVEN: A sequence of rolls by the casino player

12455264621461461361366616646616366163661636165156151151461235623 44

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

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

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 $\arg\max_{s_1,\ldots,s_T} p(\{S_t\}_{t=1}^{T}|\{O_t\}_{t=1}^{T},\theta)$ most probable

  sequence of hidden states

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 $\arg\max_{s_1,\ldots,s_T} p(\{S_t\}_{t=1}^T|\{O_t\}_{t=1}^T,\theta)$ 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

HMM Algorithms

- Evaluation – What is the probability of the observed sequence? Forward Algorithm

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

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

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

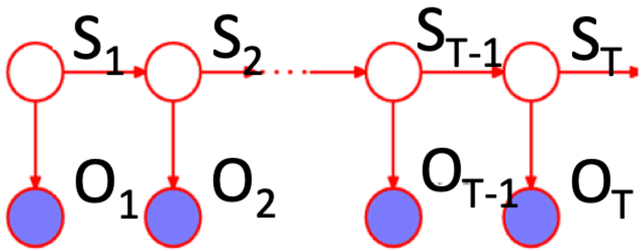
Evaluation Problem

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

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

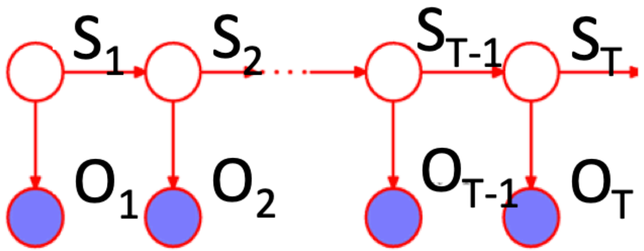


$$p(\{O_t\}_{t=1}^T) = \sum_{S_1,\ldots,S_T} p(\{O_t\}_{t=1}^T, \{S_t\}_{t=1}^T)$$

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

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

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

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

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

Forward Probability

Forward Probability

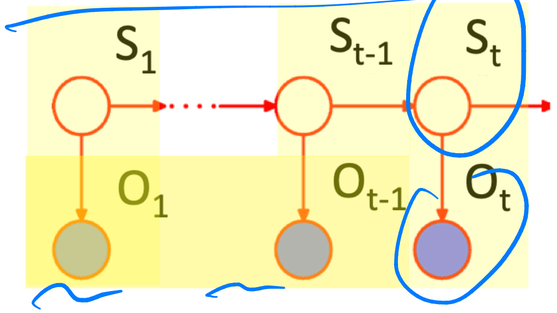
$$p(\{O_t\}_{t=1}^T) = \sum_k p(\{O_t\}_{t=1}^T, S_T = k) = \sum_k \alpha_T^k$$

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 α_t^k recursively over t

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



Introduce S_{t-1}

.
.
.

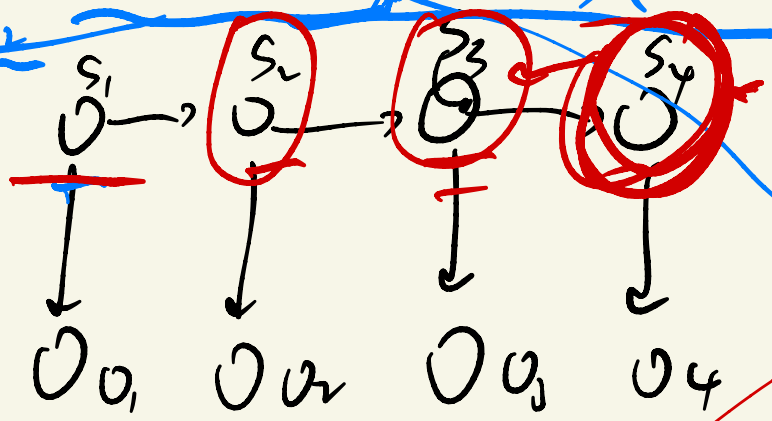
Chain rule

Markov assumption

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

$\sum_{S_2} P(S_2, O_1, O_2) \; P(S_3|S_2) \; P(O_3|S_3)$

$P(S_3|S_2) \; P(O_3|S_3)$



$\sum_{S_1} P(S_1, S_2, O_1, O_2)$

$= P(S_2, O_1, O_2)$

$P(O_1, \cdots O_4) = \sum_{S_1} \sum_{S_2} \sum_{S_3} \sum_{S_4} P(S_1) \; P(O_1|S_1) \; P(S_2|S_1)$

$P(S_3|S_2) \; P(O_3|S_2) \cdots$

$= \sum_{S_2} \sum_{S_3} \sum_{S_4} \left(\sum_{S_1} P(S_1) \; P(O_1|S_1) \; P(S_2|S_1) \right)$

$P(O_2|S_2)$

$\sum_{S_2} P(S_2, S_3, O_1, O_2, O_3)$

$= P(S_3, O_1, O_2, O_3)$

$$\sum_{S_1} P(S_1) P(O_1 | S_1) P(S_2 | S_1)$$

$$= \sum_{S_1} P(S_1, S_2, O_1)$$

$P(O_1 \cdots O_t, S_{t+1} = S_j)$

$P(S_2, O_1, O_2)$

~~$P(S_2, O_1, O_2)$~~

$$= P(S_2, O_1) \times P(O_2 | S_2)$$

α

$$= P(S_2, O_1, O_2)$$

Forward Algorithm

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

Forward Algorithm

Can compute α_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

Forward Algorithm

Can compute α_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, …, T

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

Forward Algorithm

Can compute α_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, …, 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: $p(\{O_t\}_{t=1}^T) = \sum_k \alpha_T^k$

Forward Algorithm

Can compute α_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, …, 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: $p(\{O_t\}_{t=1}^T) = \sum_k \alpha_T^k$

Can we do in the backward direction?

Forward Algorithm

Can compute α_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, …, 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

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

Posterior

Decoding Problem 1

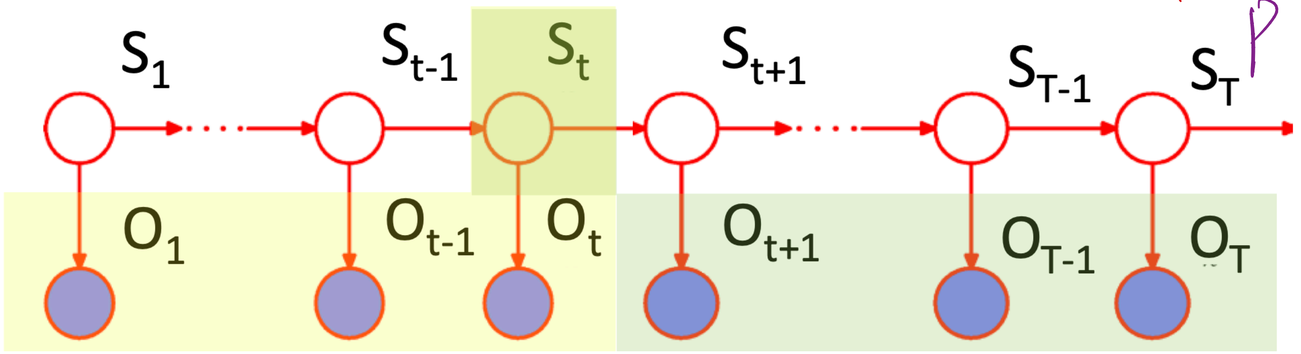
- Given HMM parameters $p(S_1), p(S_t|S_{t-1}), p(O_t|S_t)$ & observation sequence $\{O_t\}_{t=1}^T$

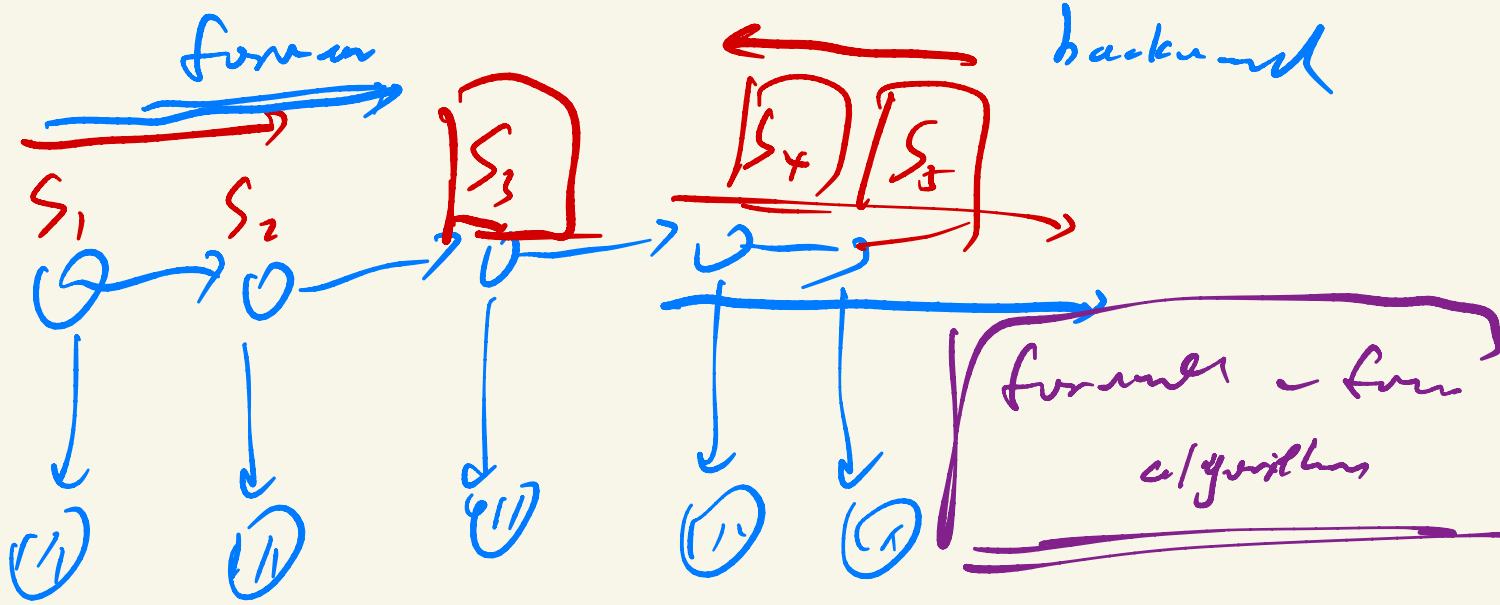
  find probability that hidden state at time t was k $p(S_t = k|\{O_t\}_{t=1}^T)$

$$p(S_t = k, \{O_t\}_{t=1}^T) = p(O_1, \ldots, O_t, S_t = k, O_{t+1}, \ldots, O_T)$$
$$= p(O_1, \ldots, O_t, S_t = k) p(O_{t+1}, \ldots, O_T | S_t = k)$$

Compute recursively

α_t^k β_t^k





$$P(S_t \mid O_1 \cdots O_5) \propto P(S_t, O_1 \cdots O_5)$$

$$\sum_{S_1} \sum_{S_2} \sum_{S_4} \sum_{S_5} P(S_1 \cdots S_5, O_1 \cdots O_5)$$

Backward Algorithm

Backward Algorithm

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

- Initialize: $\beta_T^k = 1$ for all k

Backward Algorithm

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

- Initialize: $\beta_T^k = 1$ for all k

$P(O_{T+1} | S_T = k)$

Why this initialization?

Backward Algorithm

Can compute β_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, …, 1

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

Backward Algorithm

Can compute β_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, …, 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: $p(S_t = k, \{O_t\}_{t=1}^T) = \alpha_t^k \beta_t^k$

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

Backward Algorithm

Can compute β_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, …, 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: $p(S_t = k, \{O_t\}_{t=1}^T) = \alpha_t^k \beta_t^k$

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

Can we compute β in a forward manner?

Most Likely State vs. Most Likely Sequence

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 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

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

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?

same only when

$P(S_1, S_2, S_3 | O_1, O_2, O_3) = P(S_1 | O_1, O_2, O_3) P(S_2 | O_1, O_2, O_3) \ldots$

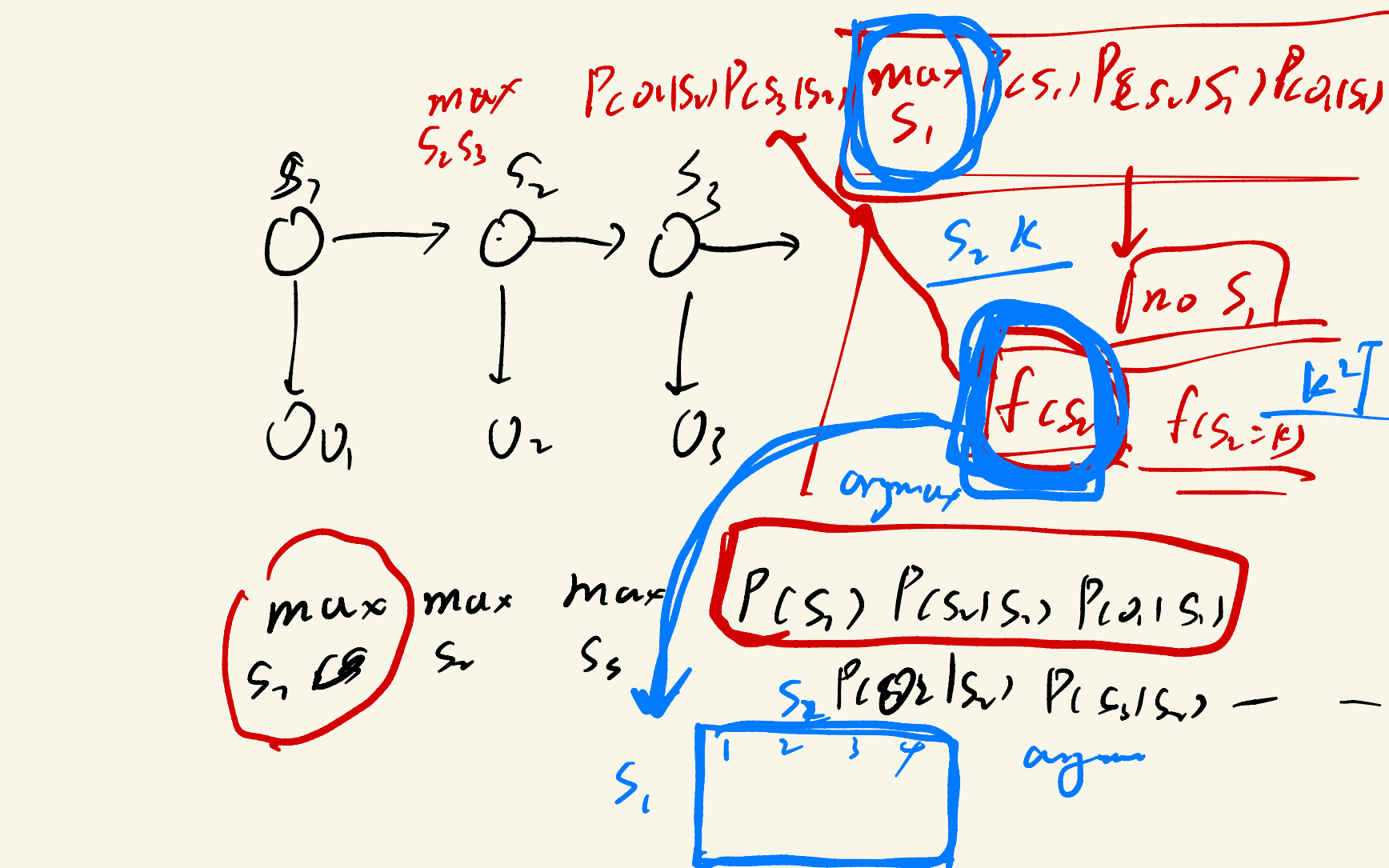
Decoding Problem 2

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

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


max
$s_2 s_3$
s_1
s_2
s_3
o_1
o_2
o_3
$P(o_1|s_1)P(s_2|s_1)$
max
s_1
$P(s_1)P(s_2|s_1)P(o_1|s_1)$
s_2 k
no s_1
$f(s_2)$
k^2
$f(s_2=k)$
argmax
max
s_1
max
s_2
max
s_3
$P(s_1)P(s_2|s_1)P(o_1|s_1)$
s_2 $P(o_2|s_2)P(s_3|s_2)$ — —
1 2 3 4
s_1
argm

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

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

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

Compute recursively

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

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

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

Compute recursively

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

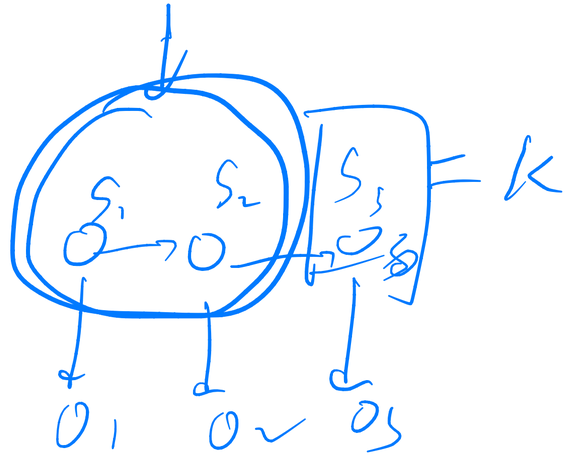
Viterbi Decoding

Viterbi Decoding

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

Viterbi Decoding

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

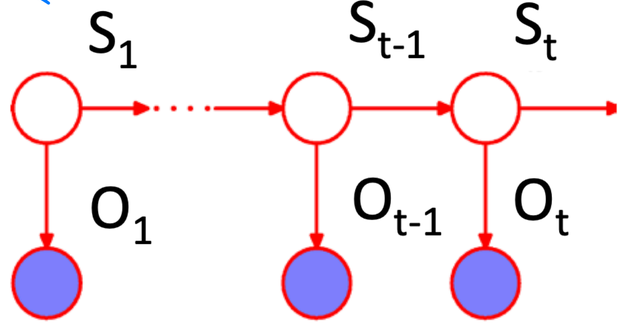


Compute probability V_t^k recursively over t

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

. 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

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

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, …, 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:

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

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

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, …, 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

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

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

Forward-Backward algorithm

Learning with EM

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

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

Forward-Backward algorithm

$$\xi_{ij}(t) = p(S_{t-1}= i, S_t = j|O,\theta)$$

$$= \frac{p(S_{t-1} = i|O,\theta)p(S_t = j, O_t, \ldots, O_T|S_{t-1} = i,\theta)}{p(O_t, \ldots, O_T|S_{t-1} = i,\theta)}$$

$$= \frac{\gamma_i(t-1)\ p_{ij}\ q_j^{O_t}\ \beta_t^j}{\beta_{t-1}^i}$$

Learning with EM

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

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

Forward-Backward algorithm

$$\xi_{ij}(t) = p(S_{t-1}= i, S_t = j|O,\theta)$$

$$= \frac{p(S_{t-1} = i|O,\theta)p(S_t = j, O_t, \ldots, O_T|S_{t-1} = i,\theta)}{p(O_t, \ldots, O_T|S_{t-1} = i,\theta)}$$

$$= \frac{\gamma_i(t-1)\ p_{ij}\ q_j^{O_t}\ \beta_t^j}{\beta_{t-1}^i}$$

You will derive the EM in your HW

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

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

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

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{n} \log p(x^{(i)}; \phi, \mu, \Sigma)$$

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

1. Intractable (no closed-form for the solution)

$\sum_z p(z, x)$

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

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

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

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

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

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

Expectation Maximization is to address the MLE optimization problem

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

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{n} \log p(x^{(i)}; \phi, \mu, \Sigma)$$

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

log P(O)

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

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

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{n} \log p(x^{(i)}; \phi, \mu, \Sigma)$$

forward

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

Wait, HMM has closed-form likelihood?

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

non-convex

Expectation Maximization is to address the MLE optimization problem

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

Thank You!