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

COMP 5212
Machine Learning
Lecture 17

Hidden Markov Models

Junxian He
Nov 7, 2024

Announcements

1. Mid-term exam grades are out, we will hold a paper-check session next week
2. Programming Assignment and HW3 will be out this week
3. We have a makeup lecture today, 7pm-820pm, at Room 2303. Attendance is not required, zoom recording will be released

Review: Elimination Algorithm / Marginalization

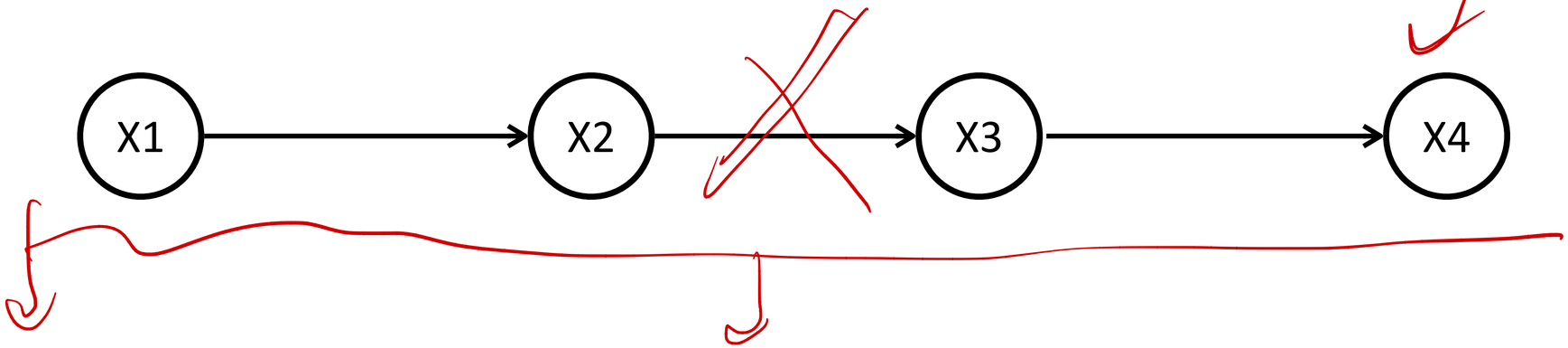
$$P(h) = \sum_{g} \sum_{f} \sum_{e} \sum_{d} \sum_{c} \sum_{b} \sum_{a} P(a,b,c,d,e,f,g,h)$$

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

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



Variable elimination

$$P(x_1, \dots, x_7) = P(x_1)\, P(x_2 \mid x_1)\, P(x_3 \mid x_2) \cdots$$

$$\cdots P(x_7 \mid x_6)$$

$x_1 \to x_2 \to x_3 \to \cdots \to x_7$

$\max_{x_1} \max_{x_2} \max_{x_3} \cdots \max_{x_7} P(x_1)\, P(x_2 \mid x_1) \cdots P(x_7 \mid x_6)$

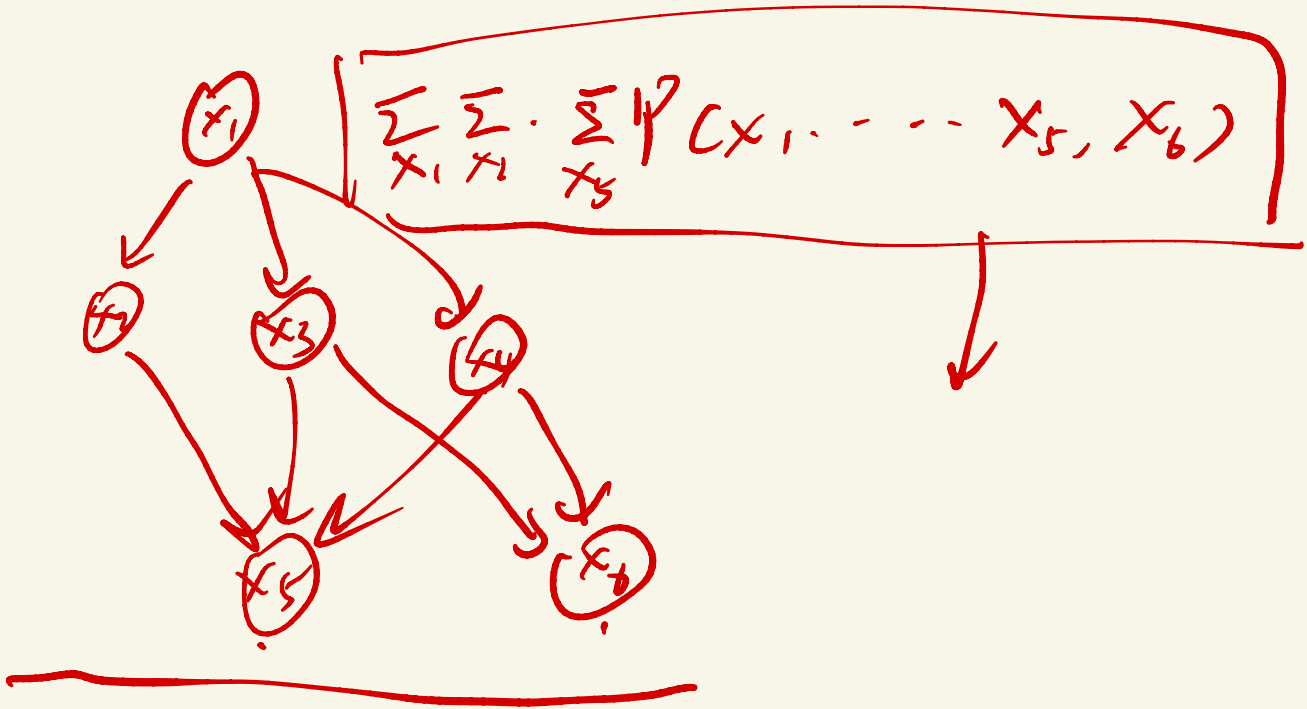
$P(x_3 \mid x_2)$

$f(x_2)$

$\sum_{x_2} \sum_{x_3} \cdots \sum_{x_7} P(x_3 \mid x_2) \cdots P(x_7 \mid x_6) \left(\sum_{x_1} P(x_1)\, P(x_2 \mid x_1) \right)$

$O(K^2)$

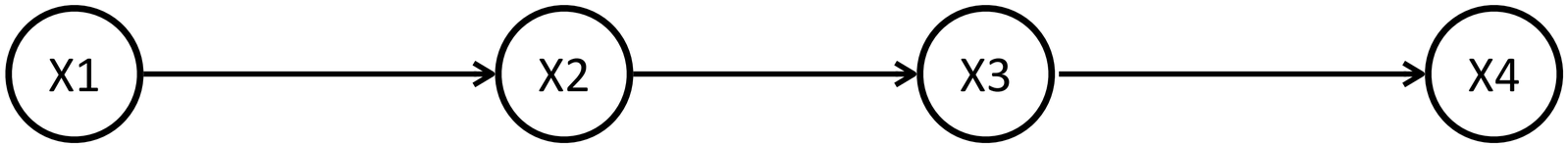
$O(K^2 \times N)$


$\sum_{x_1} \sum_{x_2} \cdot \sum_{x_5} \psi(x_1, \cdots, x_5, x_6)$
x_1
x_2
x_3
x_4
x_5
x_6

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?

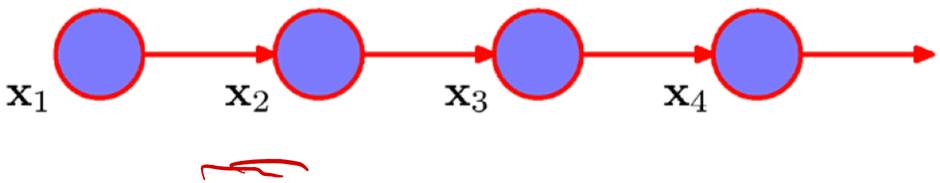
Review: Markov Models

Review: Markov Models

- Markov Assumption

1st order

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

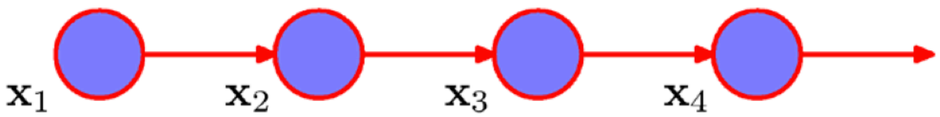


Review: Markov Models

- Markov Assumption

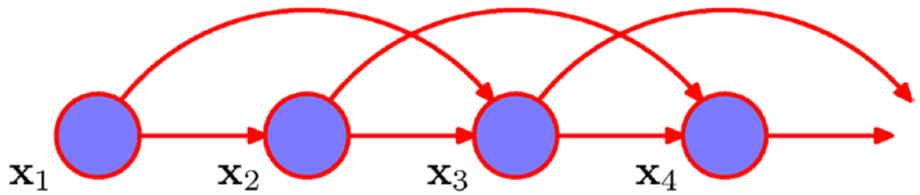
1st order

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



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

$O(K^2)$

$P(X_{n-1} | X_{n-2})$

Review: Markov Models

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

parameters in stationary model
K-ary variables

- Markov Assumption

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

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

Review: Markov Models

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

parameters in stationary model K-ary variables

- Markov Assumption

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

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

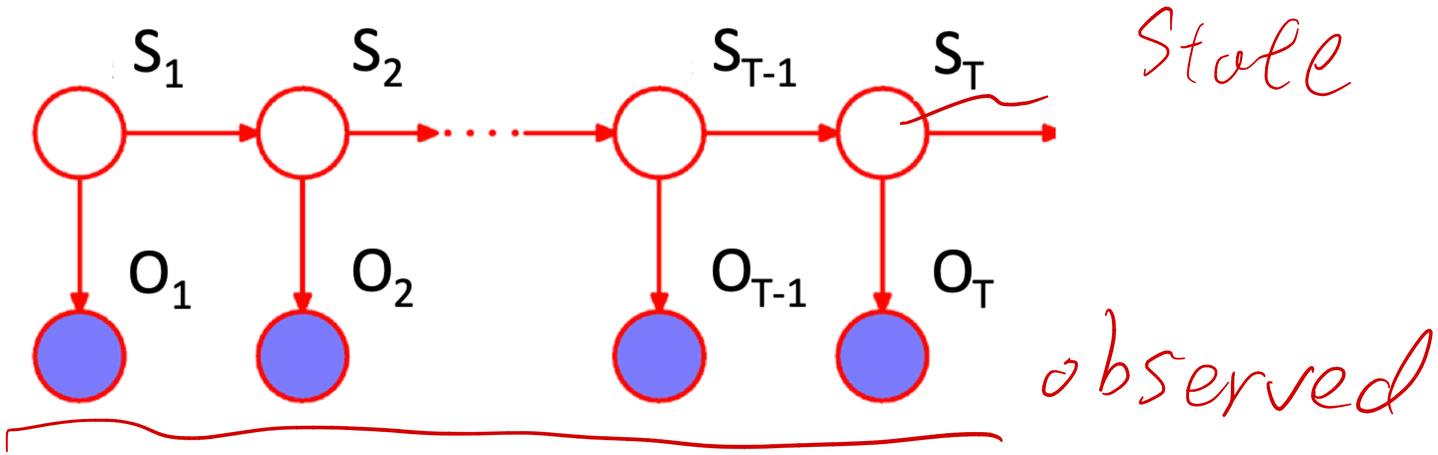
n-1th order $p(\mathbf{X}) = \prod_{i=1}^{n} p(X_n|X_{n-1},\ldots,X_1)$ $O(K^n)$

≡ no assumptions – complete (but directed) graph

naive bayes

$O(K)$

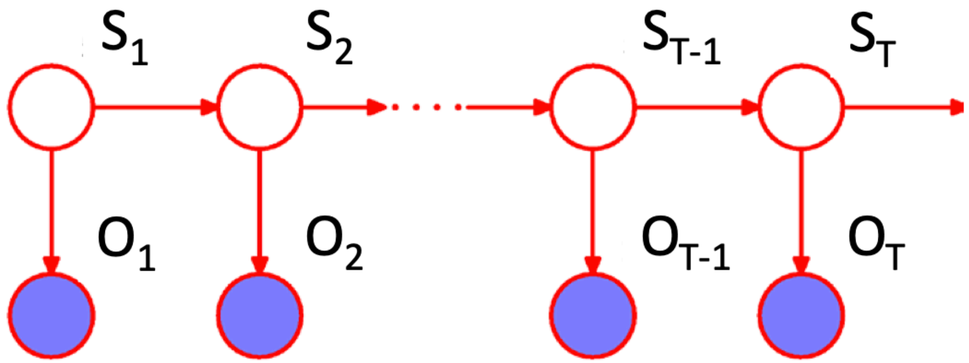
Review: Hidden Markov Models



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

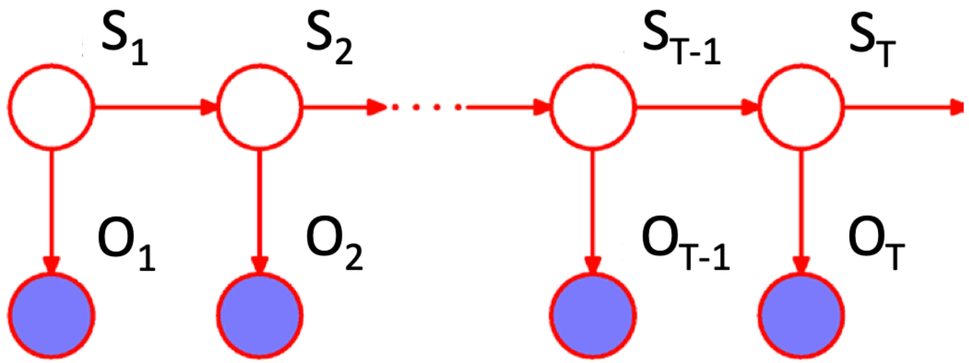
Hidden Markov Models

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

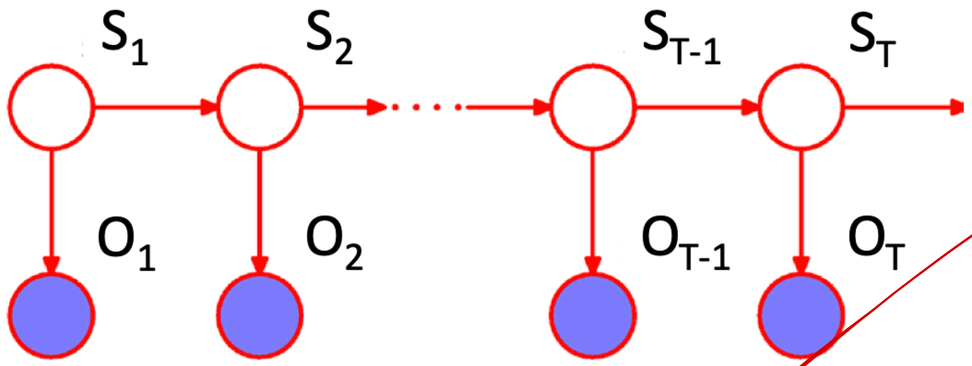


$$p(S_1, \ldots, S_T, O_1, \ldots, O_T) = \prod_{t=1}^{T} p(O_t|S_t) \prod_{t=1}^{T} p(S_t|S_{t-1})$$

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

(can be extended to higher order).

Hidden Markov Models



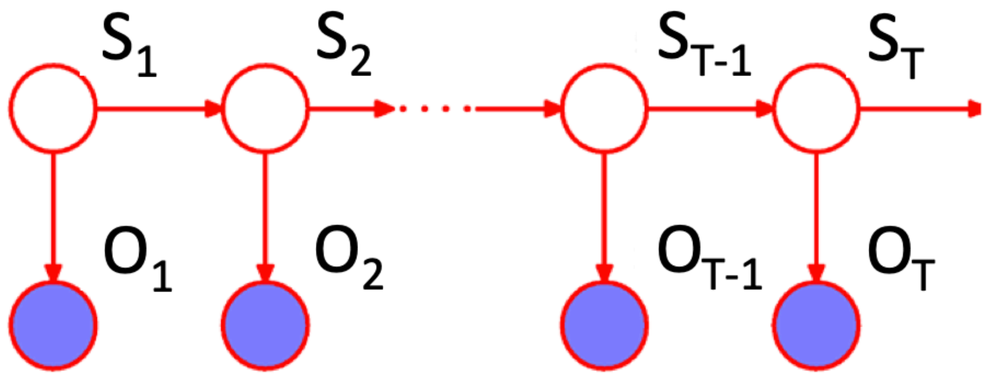
$$p(S_1, \ldots, S_T, O_1, \ldots, O_T) = \prod_{t=1}^{T} p(O_t|S_t) \prod_{t=1}^{T} p(S_t|S_{t-1})$$

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

(can be extended to higher order).

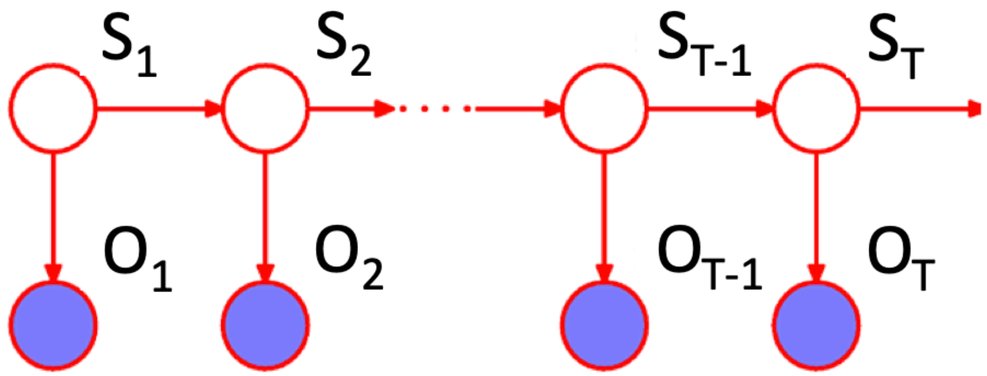
Is O_T and O_2 independent?

Hidden Markov Models



Hidden Markov Models

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

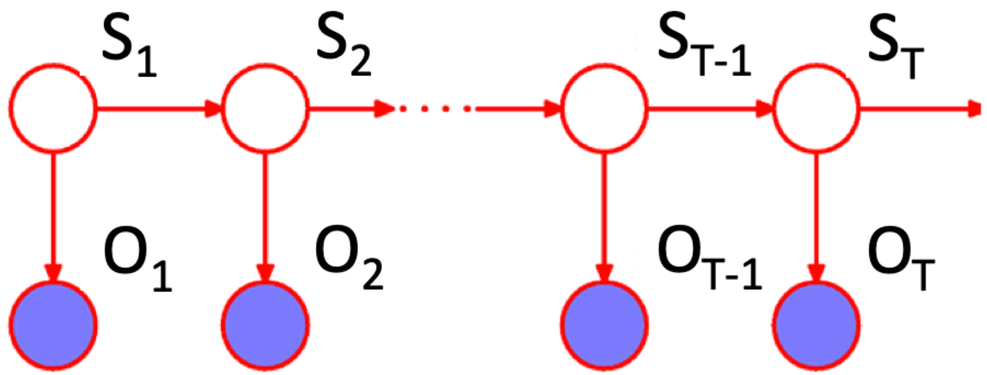


Hidden Markov Models

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

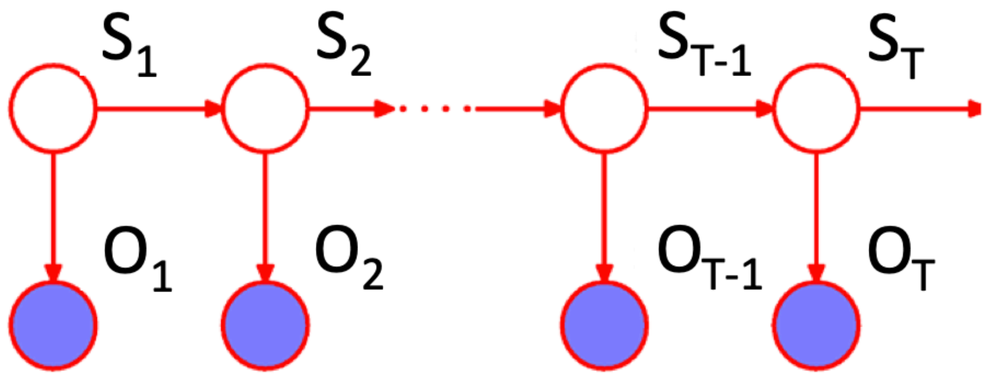
Initial probabilities

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



Hidden Markov Models

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



Initial probabilities

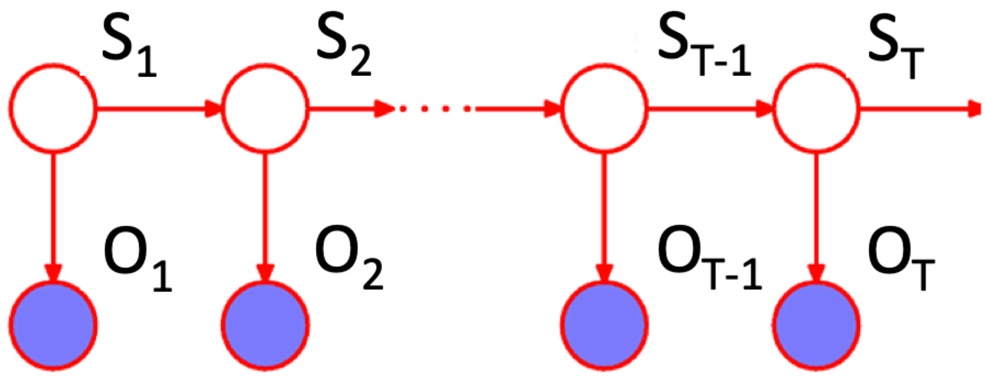
$$p(S_1 = i) = \pi_i$$

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

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

$O_{1:T}$

learning

Hidden Markov Models

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

Initial probabilities

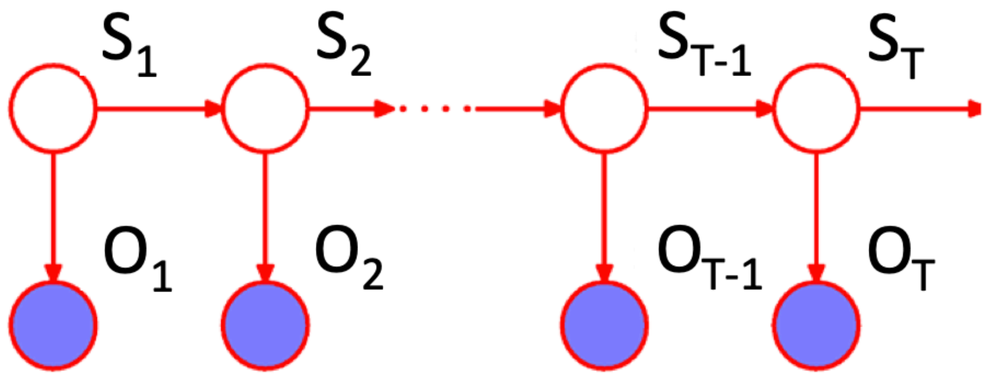
$p(S_1 = i) = \pi_i$

Transition probabilities

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

Emission probabilities

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



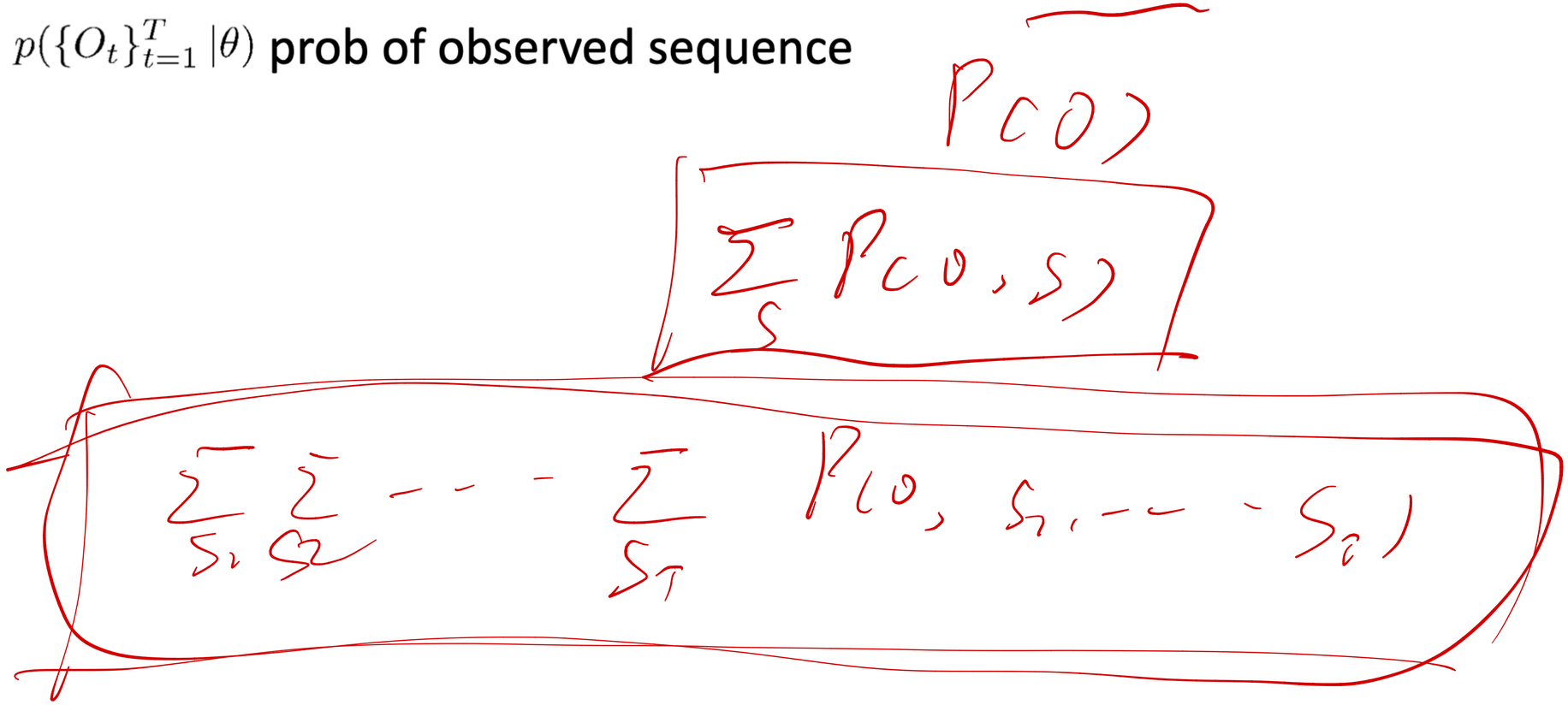
$$p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) = p(S_1) \prod_{t=2}^{T} p(S_t|S_{t-1}) \prod_{t=1}^{T} p(O_t|S_t)$$

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

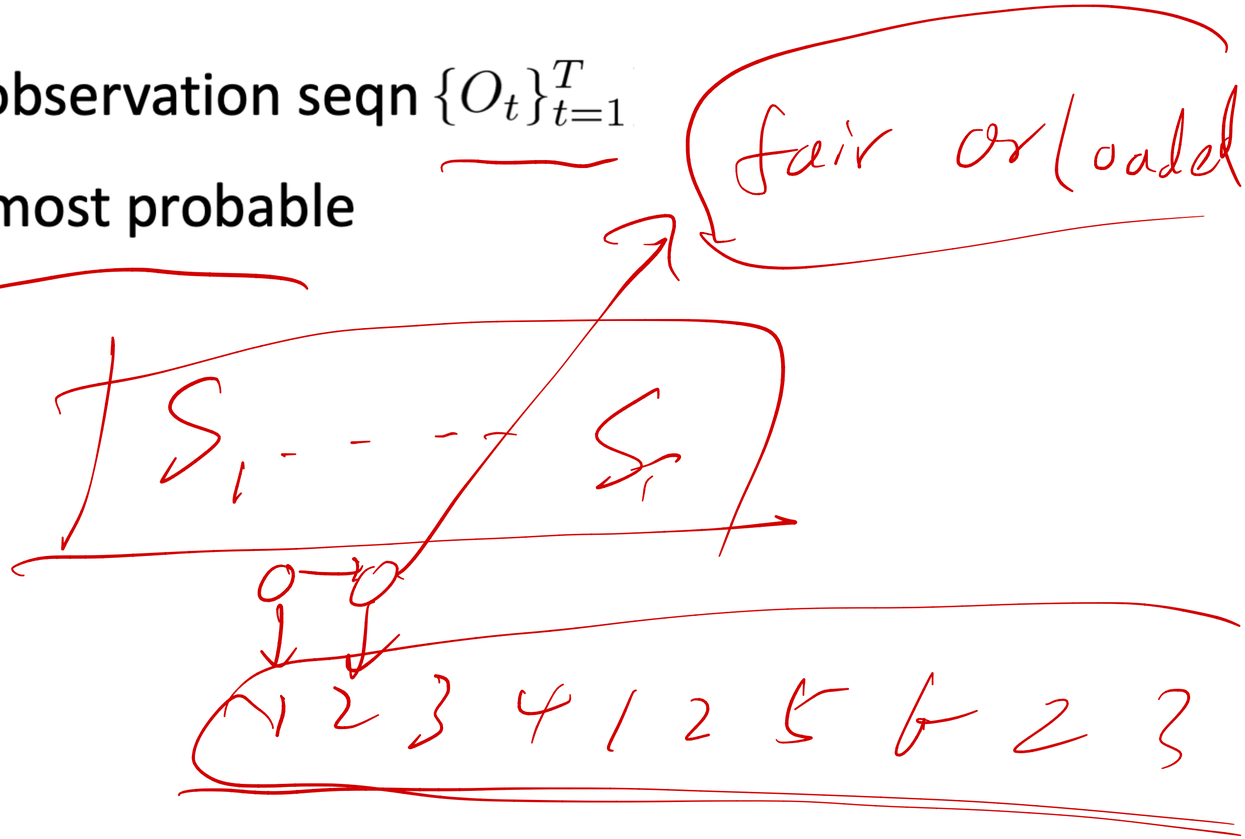
- **Evaluation** – Given HMM parameters & observation seqn $\{O_t\}_{t=1}^T$

  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 MLE

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)

forward-backward

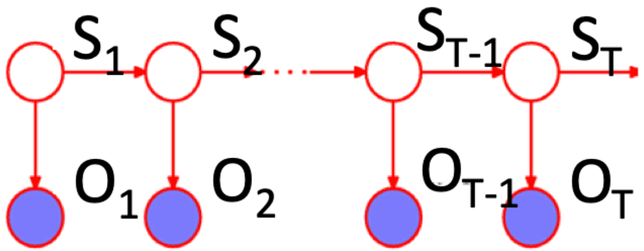
Evaluation Problem

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

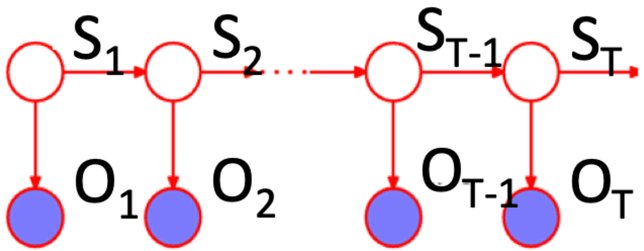
$$p(\{O_t\}_{t=1}^T) = \sum_{S_1,\ldots,S_T} p(\{O_t\}_{t=1}^T, \{S_t\}_{t=1}^T)$$

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

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

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

$S_T = k$

α_T^k

$\sum_k \alpha_T^k$

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

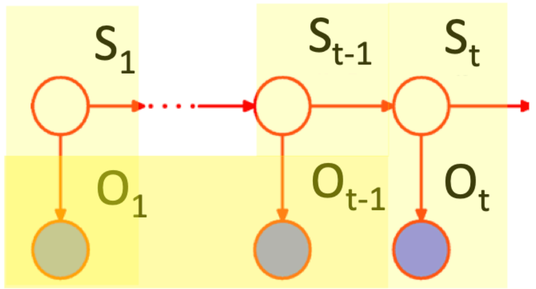
$$\alpha_T^k = \sum_{S_1} \sum_{S_2} \cdots \sum_{S_{T-1}} P(O, S_1, S_2 \cdots S_T = k)$$

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

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

$P(a,b,c,d,e,f) = P(a)\,P(b|a)\,P(c|a,b)\cdots$ $P(d|a,b,c)$

$$P(O_1, \cdots O_t, S_t = k) = \alpha_t^k$$

$$\alpha_t^k = f(\alpha_{t-1}^k) \qquad \alpha_{t-1}^k = P(O_1 \cdots O_{t-1}, S_{t-1} = k)$$

$$P(O_1, \cdots O_t, S_t = k) = \sum_i P(O_1 \cdots O_{t-1}, O_t, S_{t-1} = i, S_t = k)$$

chain rule; i: not included; α_{t-1}^i

$$= \sum_i P(O_1, \cdots O_{t-1}, S_{t-1} = i)\, P(S_t = k \mid O_1, \cdots O_{t-1}, S_{t-1} = i)$$

?

$P(S_t = k \mid S_{t-1} = i)$

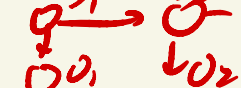
$$P(O_t \mid S_t = k, O_1, \cdots O_{t-1}, S_{t-1} = i)$$

?

$S_t \perp O_1 \cdots O_{t-1}$?

given $S_{t-1} = i$

$P(O_t \mid S_t = k)$



elimination

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

$$= \sum_{S_1} \sum_{S_2} \cdots \sum_{S_T} P(O_1|S_1)\, P(S_2)\, P(O_2|S_2)\, P(S_2|S_1) \cdots P(O_T|S_T)\, P(S_T|S_{T-1})$$

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

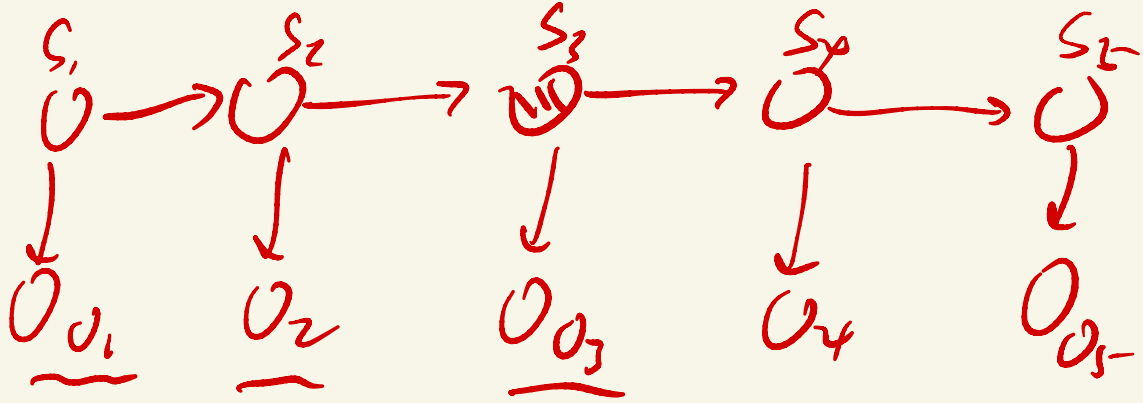
$$= \sum_{S_2} P(O_2|S_2)\, P(S_3|S_2)\, f(S_2|O_1) \quad S_2 = k$$

$f(S_2, O_1) \in \mathbb{R}$

chain rule

$P(a, b, c, \cdots e, f)$

$= P(a, b, c)\, P(d \mid a, b, c)\, P(e, f \mid a, b, c, d)$



$O_1, O_2, O_3 \perp S_4$ given S_3

$$\sum_{S_1} \sum_{S_2} \cdots \sum_{S_T} P(S_1)\, P(O_1|S_1)\, P(S_2|S_1) \cdots P(O_T|S_T)$$

$$\sum_{S_1} \sum_{S_2} \cdots \sum_{S_{T-1}} P(S_1)\, P(O_1|S_1) \cdots \sum_{S_T} P(S_T|S_{T-1})\, P(O_T|S_T)$$

backward

S_{T-1}

$f(S_{T-1})$

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

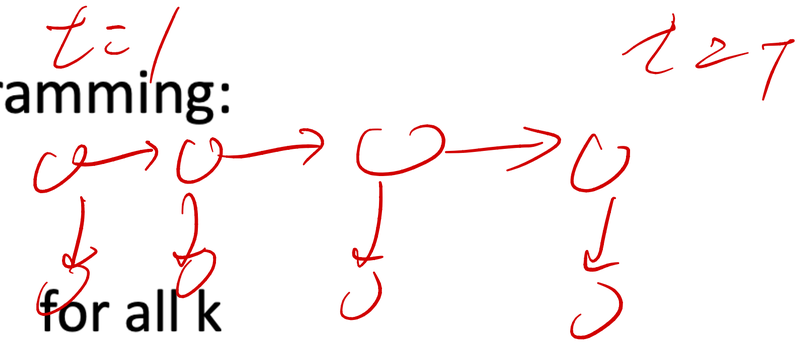
Can compute α_t^k for all k, t using dynamic programming:

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

$P(S_t = k \mid O_1, O_2 \cdots O_T)$

Decoding Problem 1

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

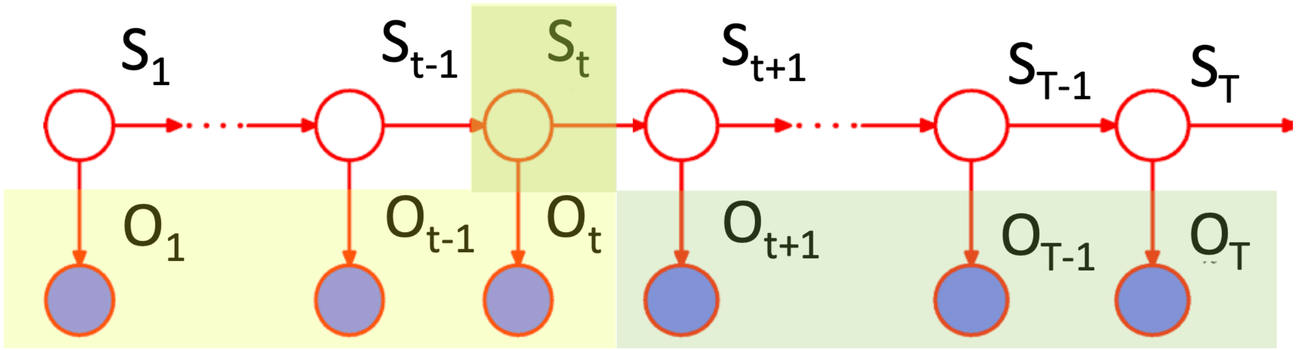
  find probability that hidden state at time t was k $p(S_t = k|\{O_t\}_{t=1}^T)$

Chain rule

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

Compute recursively

α_t^k $\beta_t^k = f(\beta_{t+1}^k)$

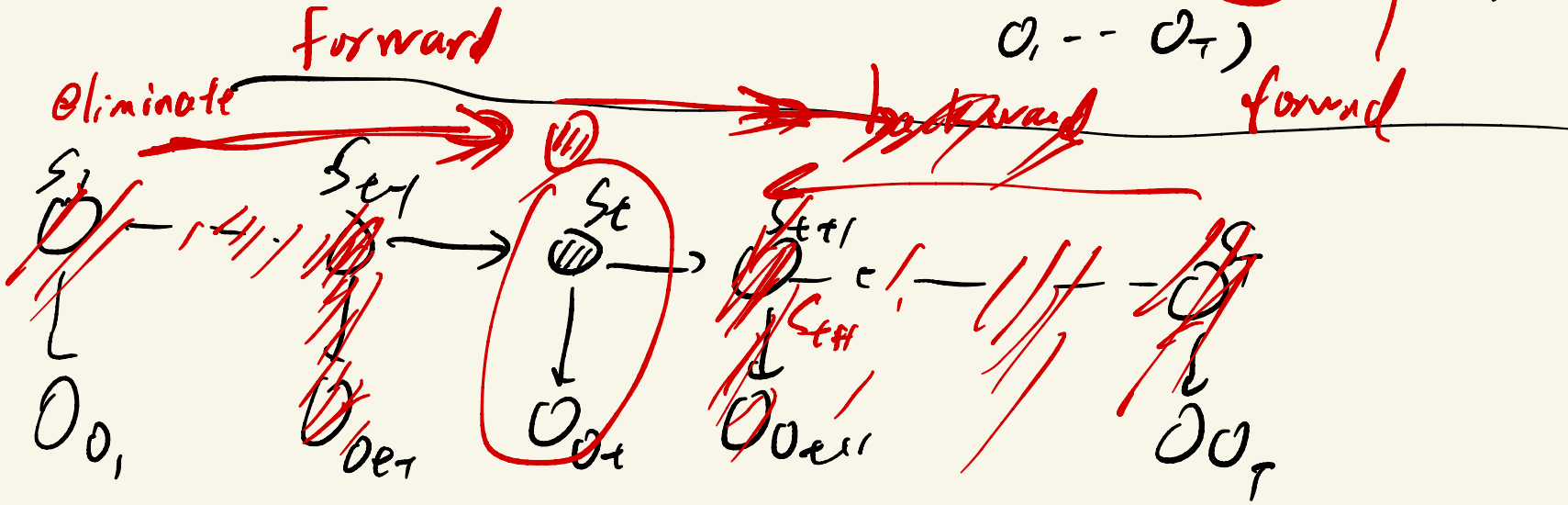


forward – forward

backward – backward

$$P(S_t = k \mid O_1, \dots O_T) \propto P(S_t = k, O_1 \dots O_T)$$

$$= \sum_{S_1} \sum_{S_2} \cdots \sum_{S_{t-1}} \sum_{S_{t+1}} \cdots \sum_{S_T} P(S_1, S_2 \cdots S_t = k, S_{t+1} \cdots S_T, O_1 \cdots O_T)$$



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

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

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

$P(S_t = k, O_1, O_2 \cdots O_T) = \alpha_t^k \beta_t^k$

$P(S_t = k | O) \propto$

Normalization: $P(S_t = k, O_1 \cdots)$

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?

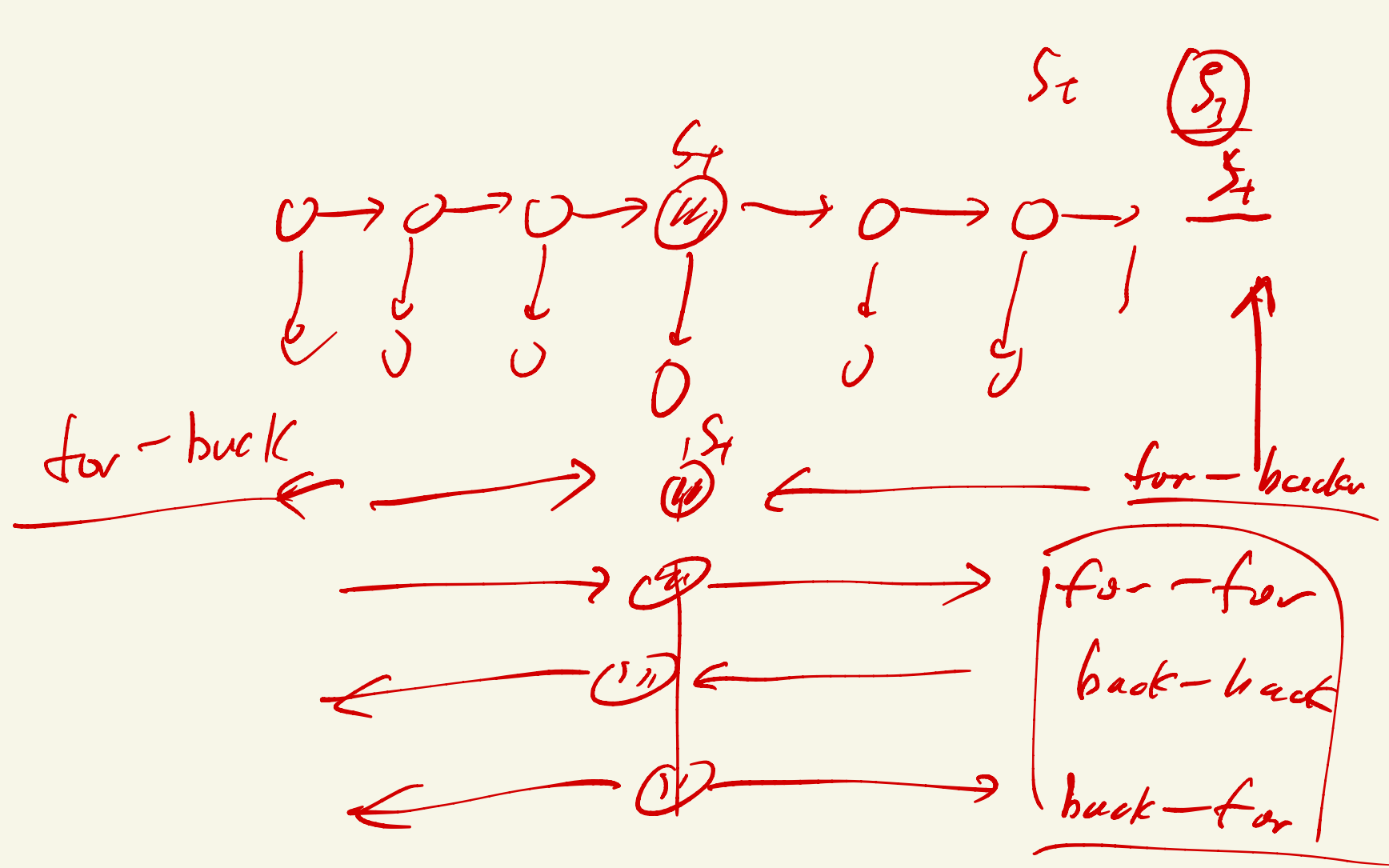
- Iterate: for t = T-1, …, 1

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

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

Can we compute β in a forward manner?

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


S_t
S_1
S_t
S_t
S_t
for - back
for - header
for - for
back - back
back - for

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? not

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

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

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

viterbi algorithm

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

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

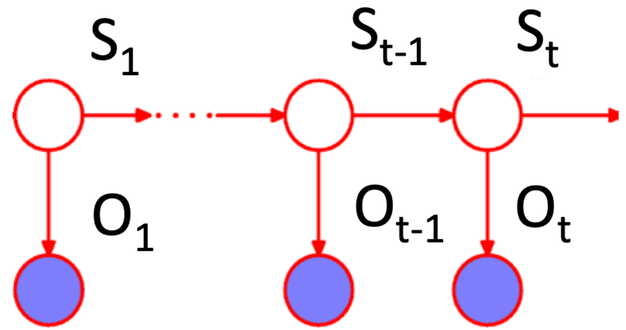
Compute probability V_t^k recursively over t

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

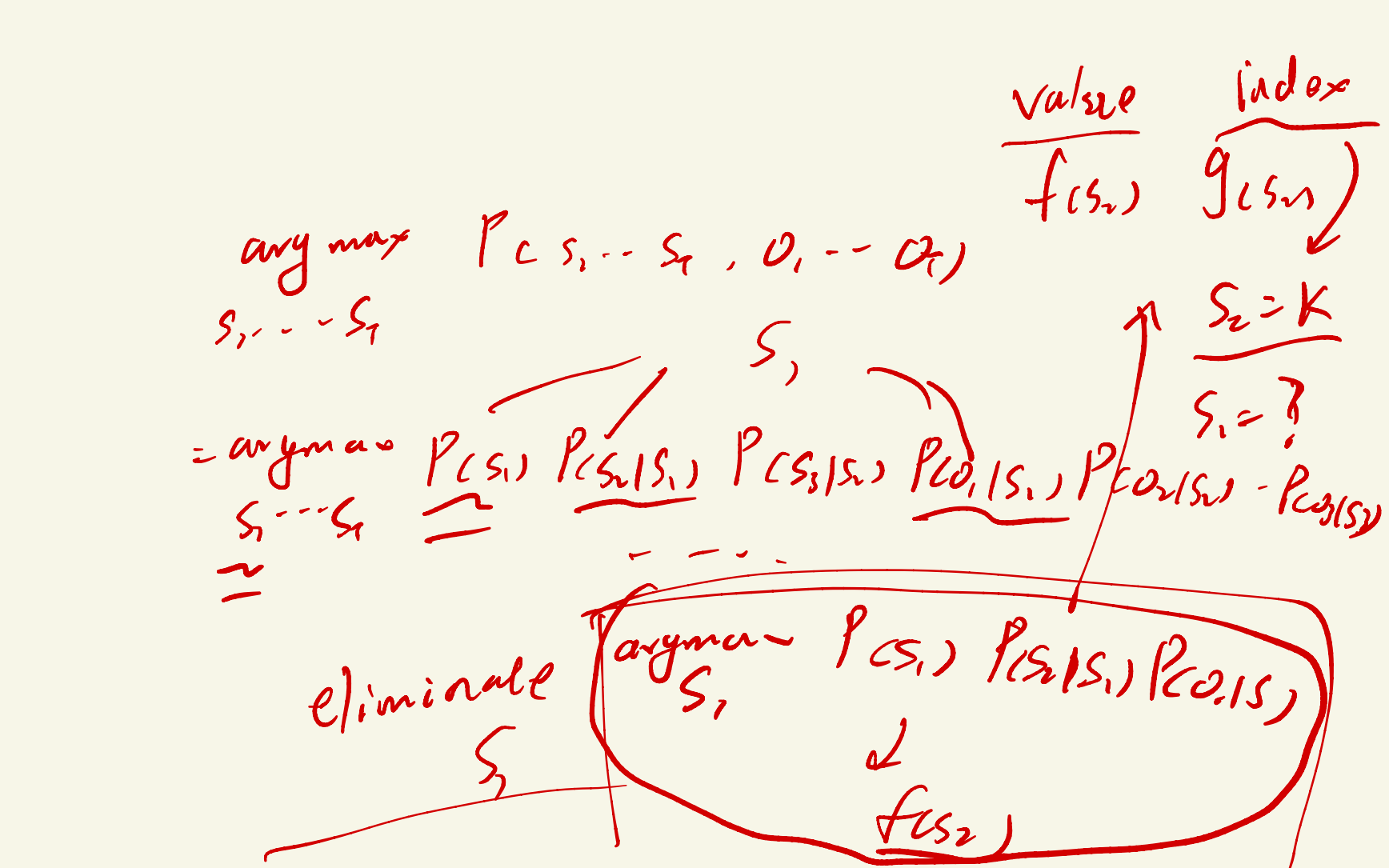
.
.
.

Bayes rule

Markov assumption



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


value
index
$f(s_2)$
$g(s_2)$
$s_2 = k$
$s_1 = ?$
$\underset{s_1 \cdots s_t}{\arg\max}\ P(s_1 \cdots s_t, o_1 \cdots o_t)$
$= \underset{s_1 \cdots s_t}{\arg\max}\ P(s_1) P(s_2|s_1) P(s_3|s_2) P(o_1|s_1) P(o_2|s_2) \cdots P(o_3|s_3)$
$\cdots$
eliminate s_1
$\underset{s_1}{\arg\max}\ P(s_1) P(s_2|s_1) P(o_1|s_1)$
$f(s_2)$

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

forward

- 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

max ELBO

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

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)

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)
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

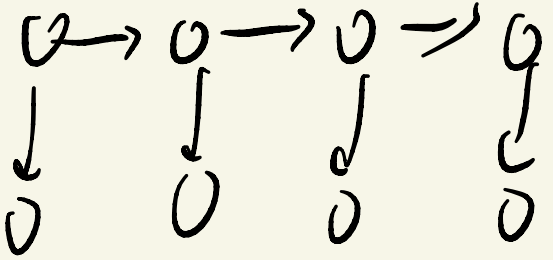
Expectation Maximization is to address the MLE optimization problem

EM because

backward ( )

$P(x) = \sum_t P(x, t)$ intractable

gradient descent



forward

$O(K^2 T)$

$P(O_1 \cdots O_T) = \sum_{S_1} \sum_{S_2} \cdots \sum_{S_T} P(S_1 S_2 S_3 \cdots O_1 \cdots O_T)$

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

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

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

Wait, HMM has closed-form likelihood?

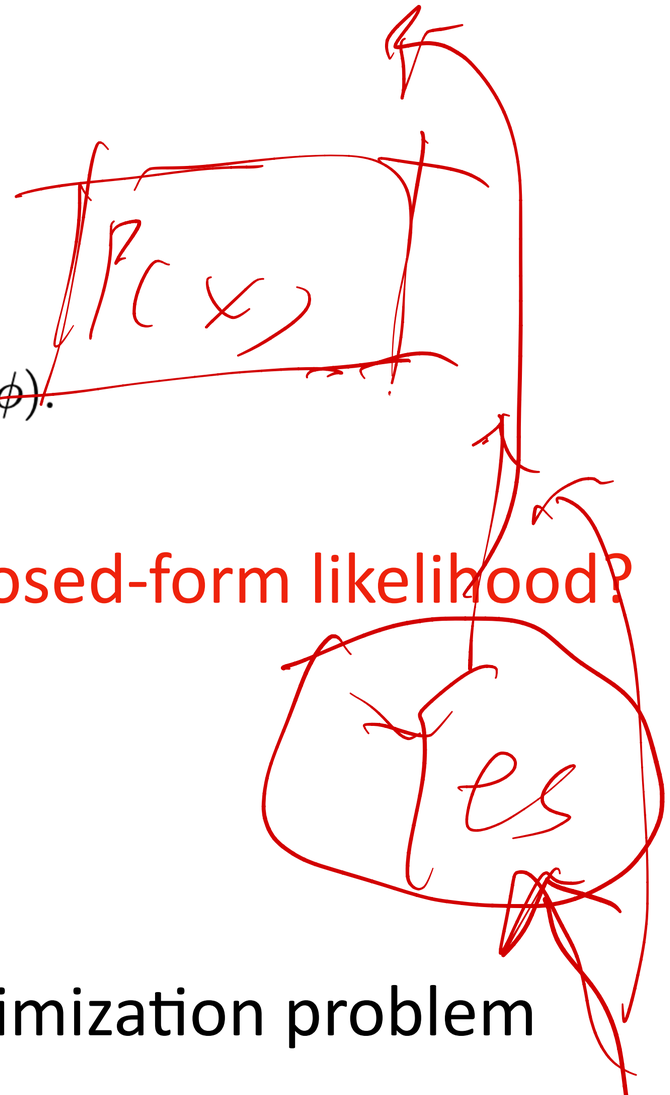
1. Intractable (no closed-form for the solution)
2. Large variance in gradient descent

Expectation Maximization is to address the MLE optimization problem

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

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

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



Wait, HMM has closed-form likelihood?

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

Expectation Maximization is to address the MLE optimization problem

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

Thank You!