

Hidden Markov Models

Aarti Singh

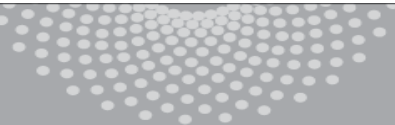
Slides courtesy: Eric Xing

Machine Learning 10-701

Apr 10, 2023



MACHINE LEARNING DEPARTMENT



Carnegie Mellon.
School of Computer Science

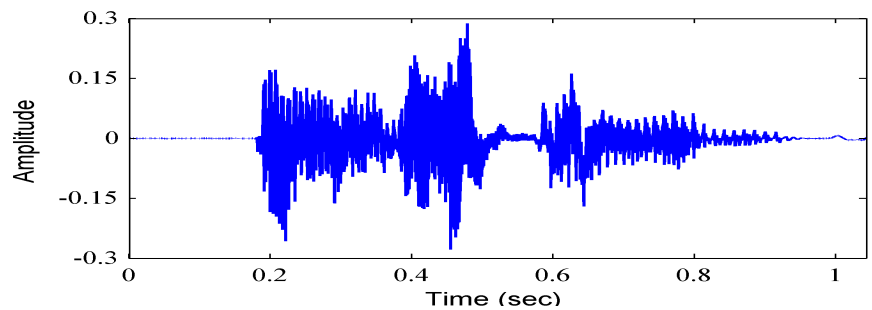
i.i.d to sequential data

- Independent, identically distributed data

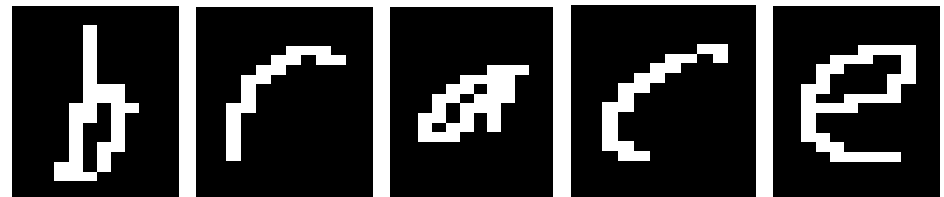
$$\{X_i\}_{i=1}^n \stackrel{iid}{\sim} p(\mathbf{X})$$

- Sequential data

- Time-series data
E.g. Speech



- Characters in a sentence



- Base pairs along a DNA strand



Markov Models

- Joint Distribution

$$\begin{aligned} p(\mathbf{X}) &= p(X_1, X_2, \dots, X_n) \\ &= p(X_1)p(X_2|X_1)p(X_3|X_2, X_1) \dots p(X_i|X_{i-1}, \dots, X_1) \\ &= \prod_{i=1}^n p(X_i|X_{i-1}, \dots, X_1) \end{aligned} \quad \text{Chain rule}$$

- Markov Assumption (m^{th} order)

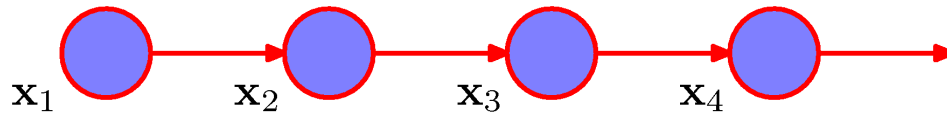
$$p(\mathbf{X}) = \prod_{i=1}^n p(X_i|X_{i-1}, \dots, X_{i-m})$$

Current observation only depends on past m observations

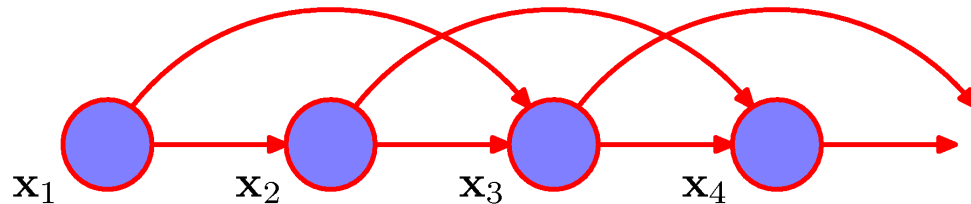
Markov Models

- Markov Assumption

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



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



Markov Models

parameters in
stationary model
d-ary variables

- Markov Assumption

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

mth order $p(\mathbf{X}) = \prod_{i=1}^n p(X_i | X_{i-1}, \dots, X_{i-m}) \quad O(d^{m+1})$

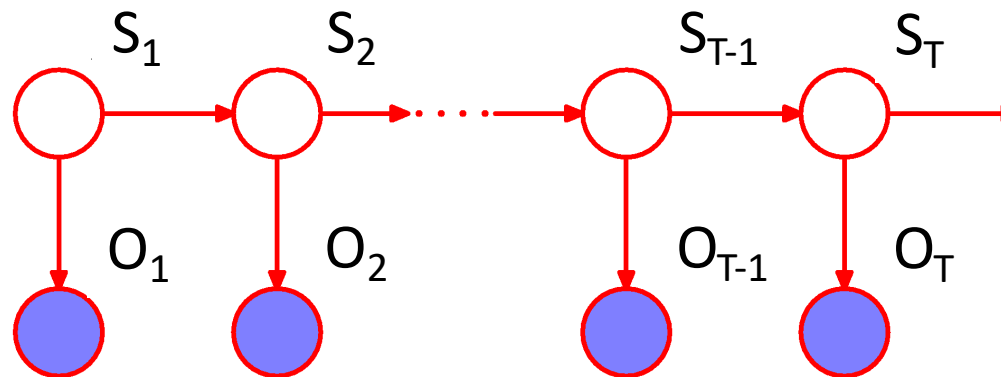
n-1th order $p(\mathbf{X}) = \prod_{i=1}^n p(X_i | X_{i-1}, \dots, X_1) \quad O(d^n)$

≡ no assumptions – complete graph

Homogeneous/stationary Markov model (probabilities of transitioning from a particular state value to another value doesn't depend on i)

Hidden Markov Models

- Distributions that characterize sequential data with few parameters but are not limited by strong Markov assumptions.



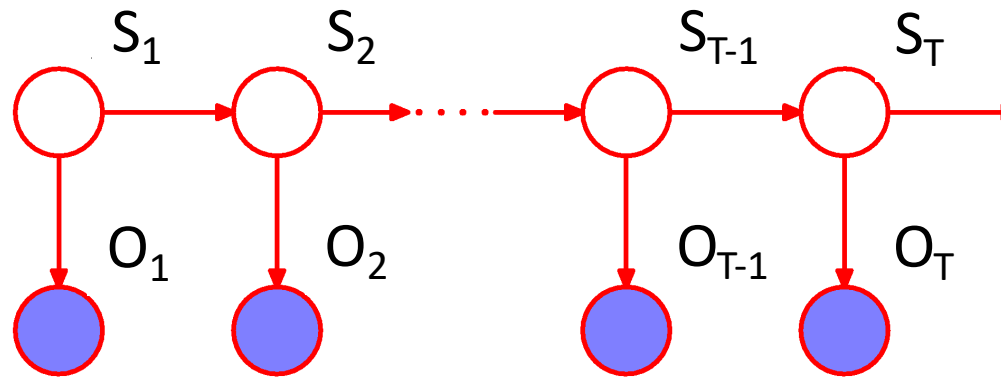
Observation space

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

Hidden states

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

Hidden Markov Models



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

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

Note: O_t depends on all previous observations $\{O_{t-1}, \dots, O_1\}$

Hidden Markov Models

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

Initial probabilities

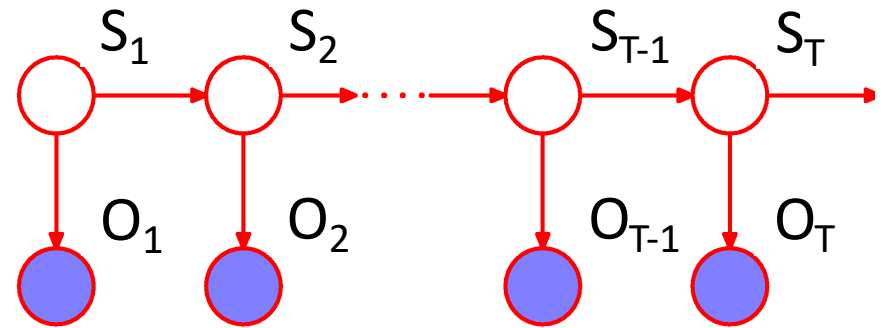
$$p(S_1 = i) = \pi_i$$

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 die:

Fair dice

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

Loaded dice

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

$$P(6) = \frac{1}{2}$$

Casino player switches back-&-forth between fair and loaded die once every 20 turns



HMM Problems

GIVEN: A sequence of rolls by the casino player

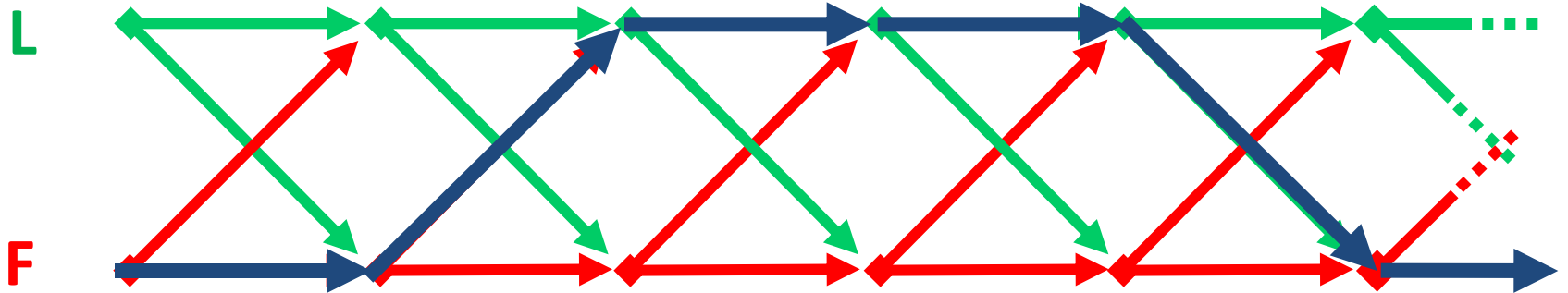
1245526462146146136136661664661636616366163616515615115146123562344

QUESTION

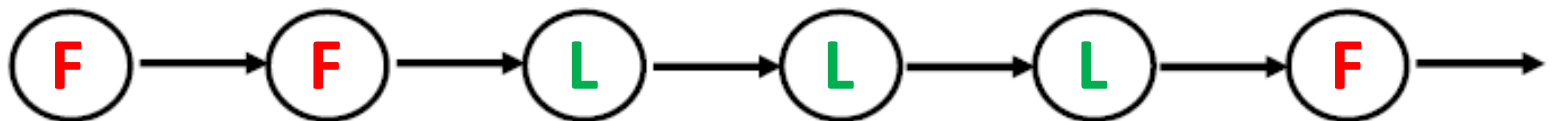
- How likely is this sequence, given our model of how the casino works?
 - This is the **EVALUATION** problem in HMMs
- What portion of the sequence was generated with the fair die, and what portion with the loaded die?
 - This is the **DECODING** question in HMMs
- How “loaded” is the loaded die? How “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

HMM Example

- Observed sequence: $\{O_t\}_{t=1}^T$

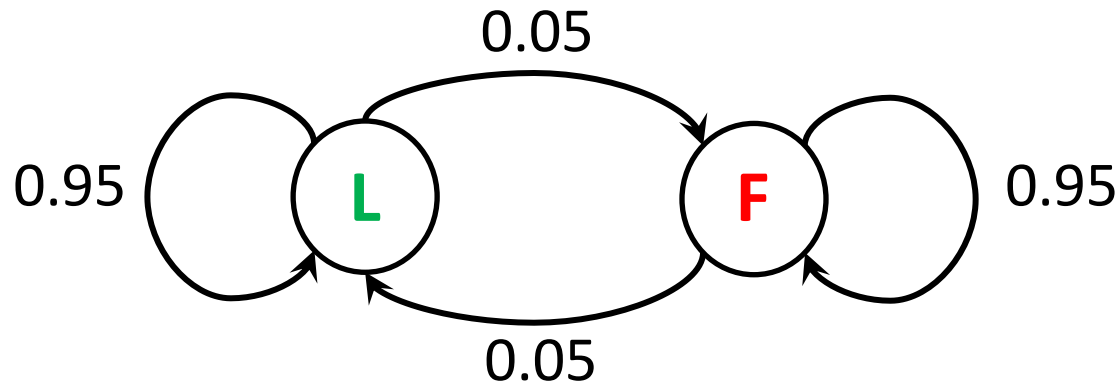


- Hidden sequence $\{S_t\}_{t=1}^T$ (or segmentation):



State Space Representation

- Switch between **F** and **L** once every 20 turns ($1/20 = 0.05$)



- HMM Parameters

Initial probs

$$P(S_1 = \mathbf{L}) = 0.5 = P(S_1 = \mathbf{F})$$

Transition probs

$$P(S_t = \mathbf{L}/\mathbf{F} | S_{t-1} = \mathbf{L}/\mathbf{F}) = 0.95$$

$$P(S_t = \mathbf{F}/\mathbf{L} | S_{t-1} = \mathbf{L}/\mathbf{F}) = 0.05$$

Emission probabilities

$$P(O_t = y | S_t = \mathbf{F}) = 1/6 \quad y = 1, 2, 3, 4, 5, 6$$

$$P(O_t = y | S_t = \mathbf{L}) = 1/10 \quad y = 1, 2, 3, 4, 5$$

$$= 1/2 \quad y = 6$$

Three main problems in HMMs

- **Evaluation** – Given HMM parameters & observation seqn $\{O_t\}_{t=1}^T$
find $p(\{O_t\}_{t=1}^T)$ prob of observed sequence
- **Decoding** – Given HMM parameters & observation seqn $\{O_t\}_{t=1}^T$
find $\arg \max_{s_1, \dots, s_T} p(\{S_t\}_{t=1}^T | \{O_t\}_{t=1}^T)$ 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

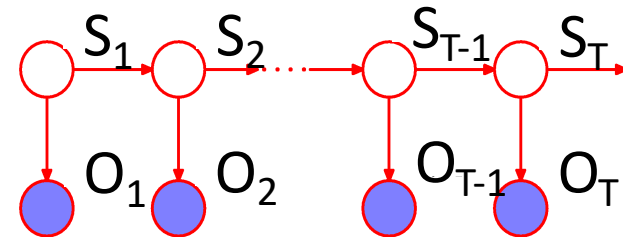
HMM Algorithms

- **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



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

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

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

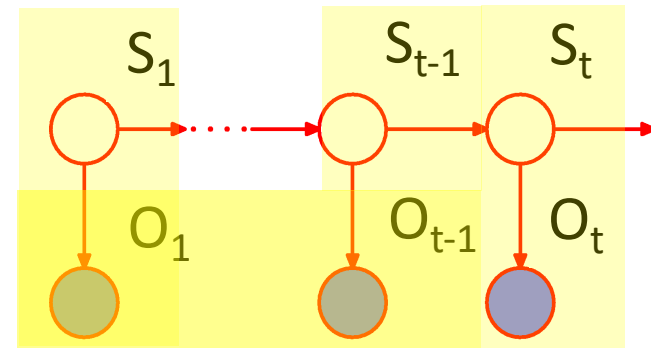
α_T^k Compute recursively

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

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



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 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 $t = 2, \dots, T$

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

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

Decoding Problem 1

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

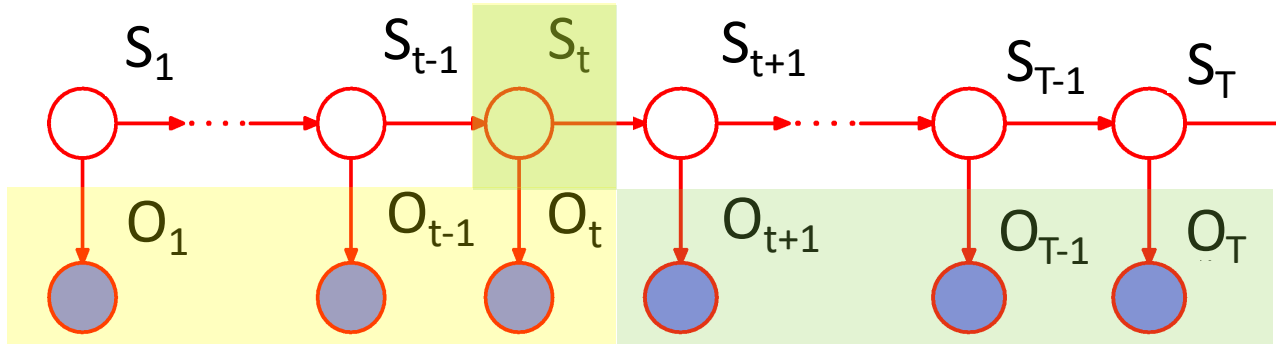
find probability that hidden state at time t was k $p(S_t = k | \{O_t\}_{t=1}^T)$

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

Compute recursively

α_t^k

β_t^k

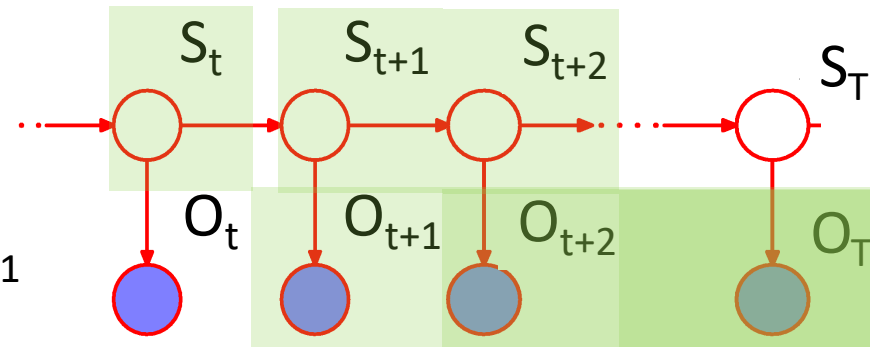


Backward Probability

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

Compute forward probability β_t^k recursively over t

$$\beta_t^k := p(O_{t+1}, \dots, O_T | S_t = k)$$



Introduce S_{t+1}

⋮

Chain rule

⋮

Markov assumption

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

Backward Algorithm

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

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

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

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

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

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

E.g. What was the most likely sequence of die rolls used by the casino given the observed sequence?

Not the same solution !

MLA of x?

MLA of (x,y)?

x	y	P(x,y)
0	0	0.35
0	1	0.05
1	0	0.3
1	1	0.3

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 $S_T = k$

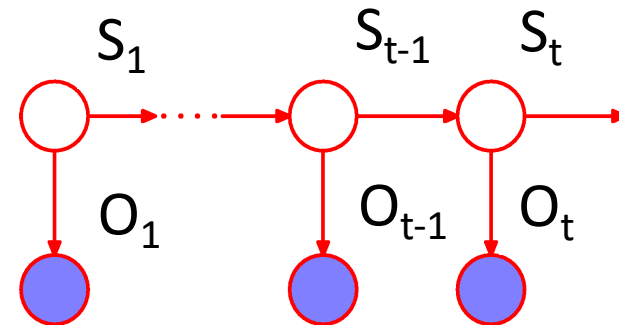
Viterbi Decoding

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

Compute probability V_t^k recursively over t

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

- Chain 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: $S_T^* = \arg \max_k V_T^k$

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

Computational complexity

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

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

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

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

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

Learning Problem

- Given HMM with unknown parameters $\theta = \{\{\pi_i\}, \{p_{ij}\}, \{q_i^k\}\}$ and observation sequence $\mathbf{O} = \{O_t\}_{t=1}^T$

find parameters that maximize likelihood of observed data

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

But likelihood doesn't factorize
since observations not i.i.d.

hidden variables – state sequence $\{S_t\}_{t=1}^T$

EM (Baum-Welch) Algorithm:

E-step – Fix parameters, find expected state assignments

M-step – Fix expected state assignments, update parameters

Baum-Welch (EM) Algorithm

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

$$\gamma_i(t) = p(S_t = i | O, \theta) = \frac{\alpha_t^i \beta_t^i}{\sum_j \alpha_t^j \beta_t^j}$$

Forward-Backward algorithm

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

Baum-Welch (EM) Algorithm

- Start with random initialization of parameters

- **E-step**

$$\gamma_i(t) = p(S_t = i | O, \theta)$$

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

$$\sum_{t=1}^T \gamma_i(t) = \text{expected \# times} \\ \text{in state } i$$

$$\sum_{t=1}^{T-1} \gamma_i(t) = \text{expected \# transitions} \\ \text{from state } i$$

$$\sum_{t=1}^{T-1} \xi_{ij}(t) = \text{expected \# transitions} \\ \text{from state } i \text{ to } j$$

- **M-step**

$$\pi_i = \gamma_i(1)$$

$$p_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

$$q_i^k = \frac{\sum_{t=1}^T \delta_{O_t=k} \gamma_i(t)}{\sum_{t=1}^T \gamma_i(t)}$$

HMMs.. What you should know

- Useful for modeling sequential data with few parameters using discrete hidden states that satisfy Markov assumption
- Representation - initial prob, transition prob, emission prob,
State space representation
- Algorithms for inference and learning in HMMs
 - Computing marginal likelihood of the observed sequence: **forward algorithm**
 - Predicting a single hidden state: **forward-backward**
 - Predicting an entire sequence of hidden states: **viterbi**
 - Learning HMM parameters: an EM algorithm known as **Baum-Welch**