Markov Chains and Hidden Markov Models

CS114 Lab 7

March 9, 2018

Kenneth Lai

Markov Chains

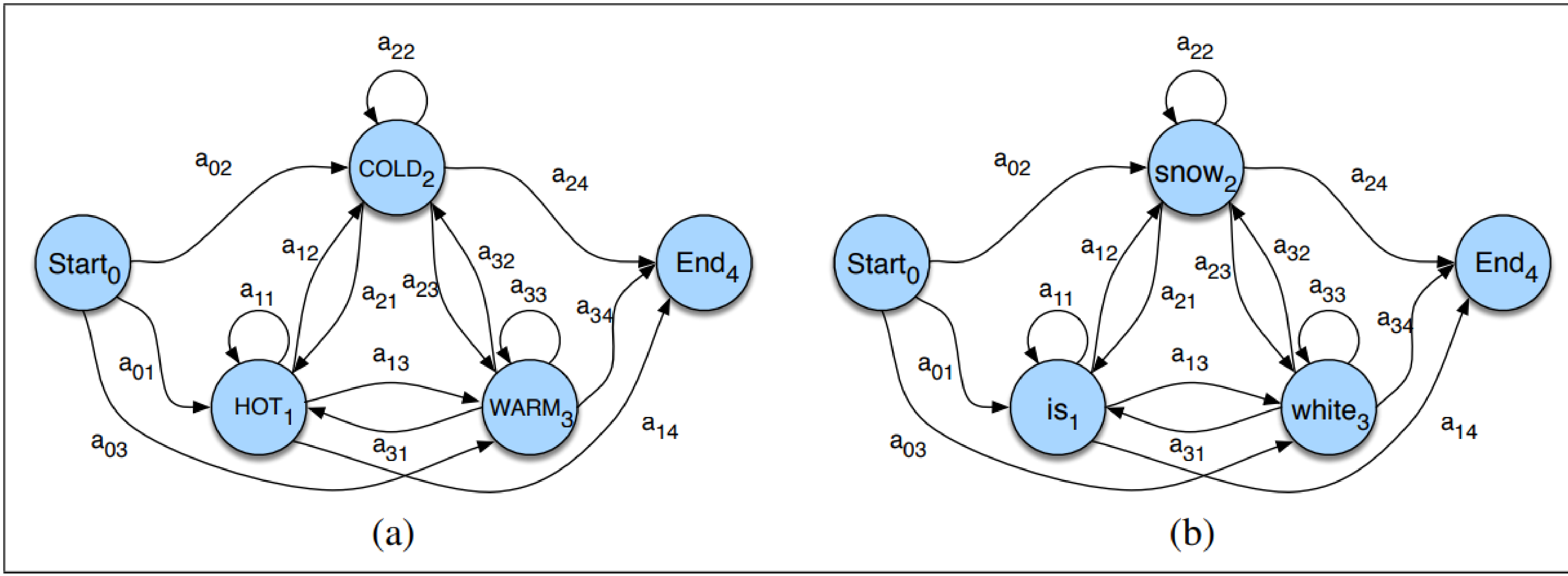

Figure 9.1 A Markov chain for weather (a) and one for words (b). A Markov chain is specified by the structure, the transition between states, and the start and end states.

Markov Chains

- Q: a set of states
- A: a transition matrix

  a_{ij} = P(moving from state i to state j)
- π: initial probability distribution
- Q_A: a set of accept states
  - May be empty

Markov Chains

- (first-order) Markov assumption

  $P(q_i|q_1...q_{i-1}) = P(q_i|q_{i-1})$

  - Where have we seen this before?

Hidden Markov Models

- Q: a set of states
- A: a transition matrix

  a_{ij} = P(moving from state i to state j)
- B: an emission matrix

  $b_i(o_t)$ = P(observation o_t generated by state i)
- π: initial probability distribution
- Q_A: a set of accept states
  - May be empty

Hidden Markov Models

- (first-order) Markov assumption

  $P(q_i|q_1...q_{i-1}) = P(q_i|q_{i-1})$

  – Where have we seen this before?

- Output independence

  $P(o_i|q_1...q_i...q_T o_1...o_T) = P(o_i|q_i)$

Hidden Markov Models

- Problem 1
  - Likelihood: P(O)
  - Lab 7 exercises
- Problem 2
  - Decoding: Q
  - HW 4 main assignment
- Problem 3
  - (unsupervised) Learning: A, B
  - HW 4 extra credit

Likelihood Problem

- How many ice creams did Jason Eisner eat?

  (3, 1, 3)

  - What is the probability of this sequence?

Likelihood Problem

$$P(O,Q) = P(O|Q) \times P(Q) = \prod_{i=1}^{T} P(o_i|q_i) \times \prod_{i=1}^{T} P(q_i|q_{i-1}) \qquad (9.10)$$

$$P(O) = \sum_{Q} P(O,Q) = \sum_{Q} P(O|Q)P(Q) \qquad (9.12)$$

Forward Algorithm

- Computing P(O) using dynamic programming

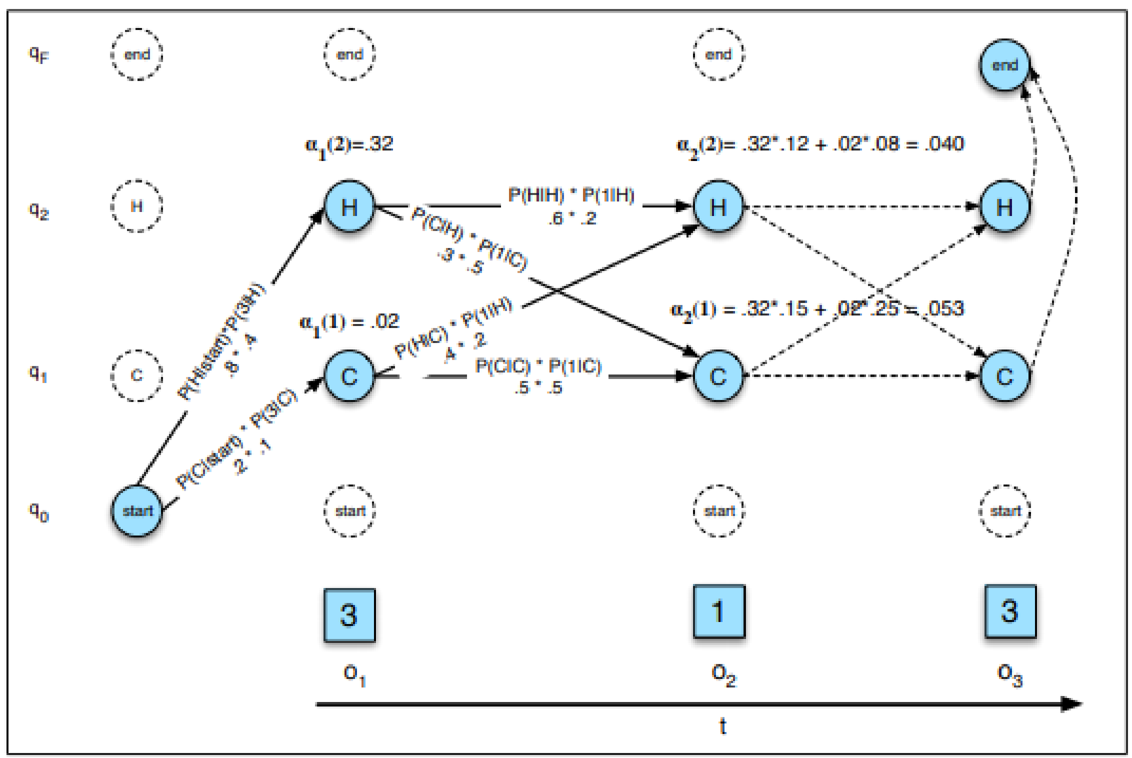

Figure 9.7 The forward trellis for computing the total observation likelihood for the ice-cream events *3 1 3*. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $\alpha_t(j)$ for two states at two time steps. The computation in each cell follows Eq. 9.14: $\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i)a_{ij}b_j(o_t)$. The resulting probability expressed in each cell is Eq. 9.13: $\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$.

Forward Algorithm

$$\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j | \lambda) \qquad (9.13)$$

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t) \qquad (9.14)$$

Forward Algorithm

function FORWARD(*observations* of len T, *state-graph* of len N) **returns** *forward-prob*

create a probability matrix *forward[N+2,T]*
for each state s **from** 1 **to** N **do** ; initialization step
  $forward[s,1] \leftarrow a_{0,s} * b_s(o_1)$
for each time step t **from** 2 **to** T **do** ; recursion step
  for each state s **from** 1 **to** N **do**

$$forward[s,t] \leftarrow \sum_{s'=1}^{N} forward[s',t-1] * a_{s',s} * b_s(o_t)$$

$$forward[q_F,T] \leftarrow \sum_{s=1}^{N} forward[s,T] * a_{s,q_F}$$; termination step

return $forward[q_F,T]$

Figure 9.9 The forward algorithm. We've used the notation $forward[s,t]$ to represent $\alpha_t(s)$.

Decoding Problem

- Jason Eisner ate (3, 1, 3) ice creams
  - What was the weather in Baltimore?

Decoding Problem

- Two options
  - Run the forward algorithm for each possible hidden state sequence
  - Run the Viterbi algorithm once

Viterbi Algorithm

- = forward algorithm, except
  - Max instead of sum
  - Backpointers