Handout 3/8/18 (UMass CS 690N)
HMM example from the Jurafsky and Martin textbook (Jason Eisner's ice cream example)

Model

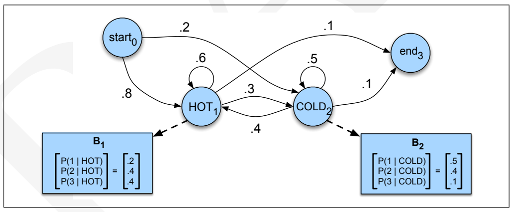


Figure 7.3 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

Forward algorithm

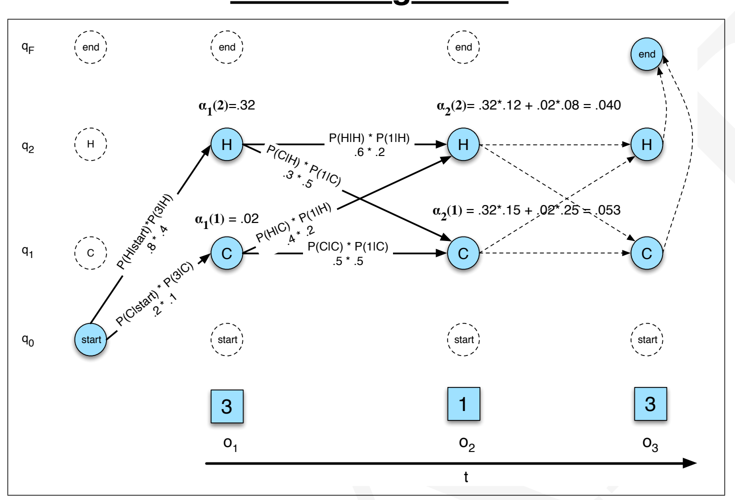


Figure 7.7 The forward trellis for computing the total observation likelihood for the ice-cream events *3 1 3*. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $\alpha_t(j)$ for two states at two time steps. The computation in each cell follows Eq. 7.14: $\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$. The resulting probability expressed in each cell is Eq. 7.13: $\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j | \lambda)$.

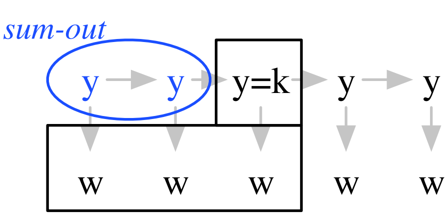
Forward-Backward

Declaratively:

(note: the backward algo. is a slightly different formulation than what I did on the blackboard on 3/8)

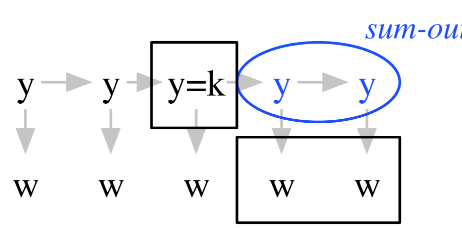
Forward probs

$$\alpha_t[k] = \sum_{y_1 \cdots y_{t-1}} P(y_t = k, w_1..w_t, y_1..y_{t-1})$$



Backward probs

$$\beta_t[k] = \sum_{y_{t+1} \cdots y_n} P(y_t = k, w_{t+1}..w_n, y_{t+1}..y_n)$$



Forward Algo.: for each *t=1..N,* for each *k*,

$$\alpha_t[k] := \sum_{j=1..K} \left(\alpha_{t-1}[j] \; P_{trans}(k \mid j) \; P_{emit}(w_t \mid k) \right)$$

Backward Algo.: for each *t=N..1*, for each *j*,

$$\beta_t[j] := \sum_{k=1..K} \left(\beta_{t+1}[j] \; P_{trans}(k \mid j) \; P_{emit}(w_{t+1} \mid k) \right)$$

Tag Marginals:

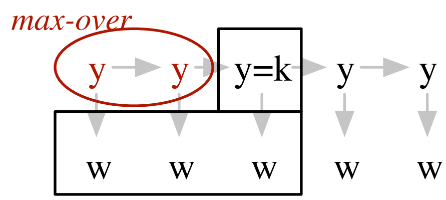
$$P(y_t = k \mid w_1..w_n) \propto \alpha_t[k] \; \beta_t[k]$$

$$P(y_{t-1} = j, y_t = k \mid w_1..w_n) \propto \alpha_t[j] \; P_{trans}(k \mid j) \; \beta_t[k]$$

Viterbi algorithm (for HMMs)

Declaratively:

$$V_t[k] = \max_{y_1 \dots y_{t-1}} P(y_t = k, \quad y_1..y_{t-1}, w_1..w_t)$$



Algorithm, for each *t=1..N,*

$$V_t[k] := \max_{j=1..K} \Big(V_{t-1}[j] \; P_{trans}(k \mid j) \; P_{emit}(w_t \mid k) \Big)$$

$$B_t[k] := \arg\max_{j=1..K} \Big(\dots \Big)$$

For solution: choose best tag at last position.
Trace backpointers to find best tag at second-to-last, e tc.

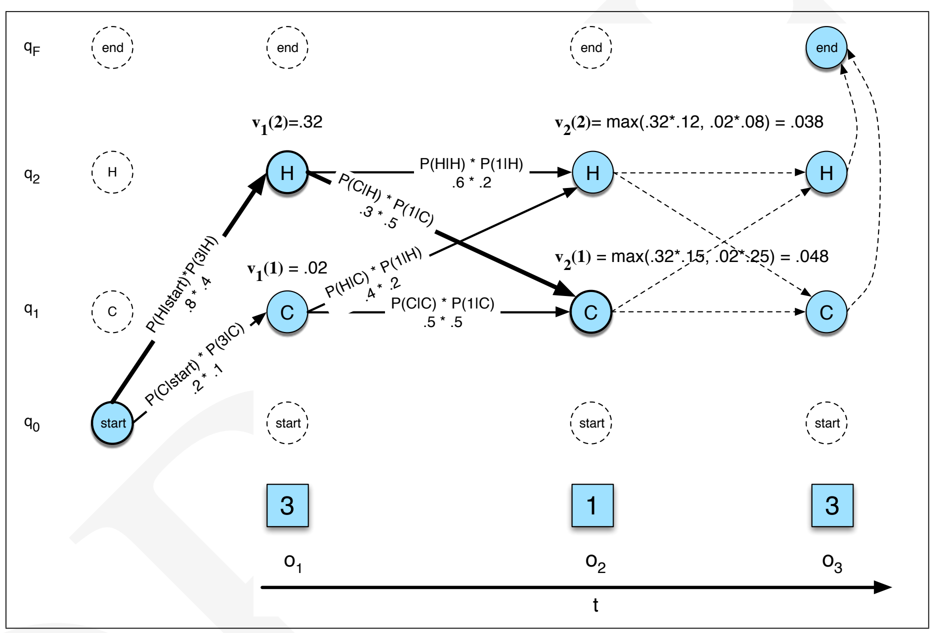


Figure 7.10 The Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events *3 1 3*. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $v_t(j)$ for two states at two time steps. The computation in each cell follows Eq. 7.19: $v_t(j) = \max_{1 \leq i \leq N-1} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$. The resulting probability expressed in each cell is Eq. 7.18: $v_t(j) = P(q_0, q_1, \dots, q_{t-1}, o_1, o_2, \dots, o_t, q_t = j|\lambda)$.