Handout 9/28/17 (UMass CS 585)
From J&M text — Jason Eisner’s ice cream / weather HMM example.

Model

![Image](https://example.com/image.jpg)

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 (sum-product)

Declaratively:

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

Recursive Algo.: for each \( t=1..N \),

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

![Image](https://example.com/image2.jpg)

Figure 7.4 The forward trellis for computing the total observation likelihood for the ice-cream events 3 I 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: \( \alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) \alpha_t(j) \). The resulting probability expressed in each cell is Eq. 7.13: \( \alpha_t(j) = P(o_1, o_2, \ldots, o_t, o_t = j | \lambda) \).
**Viterbi algorithm** (max-product)

**Declaratively:**

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

**Algorithm**, for each \( t=1..N \),

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

---

**Figure 7.10**  The Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events \( 3 \) \& \( 3 \). 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: \( v_t(j) = \max_{1 \leq i \leq N} v_{t-1}(i) \ a_{ij} \ b_j(o_t) \). The resulting probability expressed in each cell is Eq. 7.18: \( v_t(j) = P(q_0, q_1, ..., q_{t-1}, o_1, o_2, ..., o_t; q_t = j | \lambda) \).
Sticky-favoring model over hidden state vocab \{0,1\}. Factor scores are “goodness points” are in log-scale additive form. (They’re positive, though for an HMM they would all be negative.)

$$\log P(y \mid w) = (\text{constant}) + G(y_1, y_2, y_3)$$

$$G(y_1, y_2, y_3) = A(y_1, y_2) + A(y_2, y_3) + B_1(y_1) + B_2(y_2) + B_3(y_3)$$

Additive Viterbi

For t=1..T,
For k in \{0,1\},

$$V_t[k] := \max_j (V_{t-1}[j] + A(j, k) + B_t(k))$$

$$B[k] := \arg \max_j (\ldots)$$

For t=1, set $$A_0(\text{anything})=0$$ and $$V_0(\text{anything})=0$$

Final backtrace step: take best-scoring from last $$V_T$$, follow the backpointers all the way back

Run Viterbi and fill out the trellis with arcs like in the textbook’s HMM example.

Most probable sequence $$y^* = (\ , \ , \ , \ )$$ $$G(y^*) =$$