Computational Linguistics 1

CMSC/LING 723, LBSC 744

Kristy Hollingshead Seitz
Institute for Advanced Computer Studies
University of Maryland

Lecture 11: 6 October 2011

Agenda

- Homework
  - HW1 – graded! (sending by email)
  - HW2 – graded by next Tuesday (maybe Thursday)
  - HW3 – due next Thursday 10/13
- Questions, comments, concerns?
- Re-visit Viterbi & Forward Algorithms
- Forward-Backward Algorithm

Viterbi Algorithm

- Use an $N \times T$ trellis $[v_{tj}]$
  - Just like in forward algorithm
- v_{tj} or $v_t(j)$
  - = P(in state j after seeing t observations and passing through the most likely state sequence so far)
  - = $P(q_1, q_2, \ldots q_{t-1}, q_{t=j}, o_1, o_2, \ldots o_t)$
- Each cell = extension of most likely path from other cells
  $v_t(j) = \max_i v_{t-1}(i)\, a_{ij}\, b_j(o_t)$
  - $v_{t-1}(i)$: Viterbi probability until $(t-1)$
  - a_{ij}: transition probability of going from state i to j
  - $b_j(o_t)$: probability of emitting symbol o_t in state j
- $P = \max_i v_T(i)$

Viterbi Algorithm: Formal Definition

- Initialization

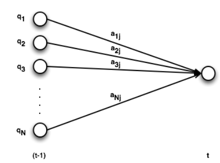
$$v_1(j) = \pi_i b_i(o_1); 1 \le i \le N$$
$$BT_1(i) = 0$$

- Recursion

$$v_t(j) = \max_{i=1}^{N} [v_{t-1}(i) a_{ij}]\, b_j(o_t); 1 \le i \le N, 2 \le t \le T$$
$$BT_1(i) = \arg\max_{i=1}^{N} [v_{t-1}(i) a_{ij}]$$

- Termination

$$P^* = \max_{1=1}^{N} v_T(j)$$
$$q_T^* = \arg\max_{1=i}^{N} v_T(j)$$



HMM Tagger – Initialization (v2)

word sequence: $W = w_1 \ldots w_n$, for time $1 \le t \le n$
total corpus size: N
input (word) vocabulary: $v_i \in V$ for $1 \le i \le k$
output (tag) vocabulary: $\tau_j \in T$ for $1 \le j \le m$
Let $b_j(v_i) = P(v_i \mid \tau_j) = c(\tau_j, v_i)/c(\tau_j)$
Let $a_{ij} = P(\tau_j \mid \tau_i) = [c(\tau_i, \tau_j)+1]/[c(\tau_i)+m]$
Let $a_{0j} = \pi(\tau_j) = P(\tau_j) = c(\tau_j)/N$
Let $\alpha_0(0) = 1$ and $\alpha_j(t) = \max_i [\alpha_i(t-1) * a_{ij}] * b_j(w_t)$
$\zeta_j(t) = \text{argmax}_i [\alpha_i(t-1) * a_{ij}]$
(backtrace)

Viterbi Algorithm (version 2)

word sequence: $W = w_1 \ldots w_n$, size of tagset $|T| = m$
for t = 1 to n
  for j = 1 to m
    $\zeta_j(t) \leftarrow \text{argmax}_i [\alpha_i(t-1) * a_{ij}]$
    $\alpha_j(t) \leftarrow \max_i [\alpha_i(t-1) * a_{ij}] * b_j(w_t)$
$\zeta_0(n+1) \leftarrow \text{argmax}_i (\alpha_i(n))$
$\rho(n+1) \leftarrow 0$
for $t = n$ to 1
  $\rho(t) \leftarrow \zeta_{\rho(t+1)}(t+1)$
  $\hat{\tau}(t) \leftarrow \tau_{\rho(t)}$

Viterbi Algorithm (version 3)

- pseudocode for the Viterbi algorithm is also given in the textbook
  - Just be sure to initialize as defined on slide 41 of lecture 9

Forward Algorithm

- Use an $N \times T$ trellis or chart $[\alpha_{tj}]$
- Forward probabilities: α_{tj} or $\alpha_t(j)$
  - = *P*(being in state *j* after seeing *t* observations)
  - = $P(o_1, o_2, \ldots o_t, q_t{=}j)$
- Each cell = ∑ extensions of all paths from other cells
  $\alpha_t(j) = \sum_i \alpha_{t-1}(i)\, a_{ij}\, b_j(o_t)$
  - $\alpha_{t-1}(i)$: forward path probability until (*t-1*)
  - a_{ij}: transition probability of going from state *i* to *j*
  - $b_j(o_t)$: probability of emitting symbol o_t in state *j*
- $P(O|\lambda) = \sum_i \alpha_T(i)$

Forward Algorithm: Formal Definition

- Initialization

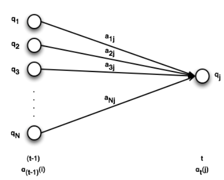
$$\alpha_1(j) = \pi_j b_j(o_1), 1 \le j \le N$$

- Recursion

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t); 1 \le j \le N, 2 \le t \le T$$

- Termination

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$



Forward-Backward (Baum-Welsch) Algorithm

- What if, instead of wanting to know:
  - *P*(being in state *j* after seeing *t* observations) (Forward Algorithm)
  - *P*(in state *j* after seeing *t* observations and passing through the most likely state sequence so far) (Viterbi Algorithm)
- We want to know:
  - *P*(being in state *j* at time *t* given the entire observation sequence)
  - *P*(being in state *j* at time *t* *and* being in state *k* at time *t*+1 given the entire observation sequence)
- Our forward probability $\alpha_j(t)$ is insufficient to calculate these *conditional* probabilities
- Also need a *backward* probability

Forward and Backward Probabilities

word sequence: $W = w_1 \ldots w_n$, **for time** $1 \le t \le n$

Forward probability:

(probability of seeing initial sequence $w_1 \ldots w_t$ and having tag j at time t)

$$\alpha_0(0) = 1 \qquad \alpha_j(t) = \left(\sum_{i=1}^{m} \alpha_i(t-1) a_{ij}\right) b_j(w_t)$$

Backward probability:

(probability of seeing remaining sequence $w_{t+1} \ldots w_n$ given tag i at time t)

$$\beta_i(n) = a_{i0} \qquad \beta_i(t) = \sum_{j=1}^{m} \beta_j(t+1) a_{ij} b_j(w_{t+1})$$

$$\mathrm{P}(w_1 \ldots w_n) = \beta_0(0) = \sum_{i=1}^{m} \alpha_i(n) a_{i0}$$

New Parameters for Forward-Backward

Probability of having tag i at time t given $w_1 \ldots w_n$

$$\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{m} \alpha_j(t)\beta_j(t)}$$

Probability of having tag i at time t and tag j at time $t+1$, given $w_1 \ldots w_n$

$$\xi_{ij}(t) = \frac{\gamma_i(t) a_{ij} b_j(w_{t+1}) \beta_j(t+1)}{\beta_i(t)}$$

Forward-Backward Algorithm

word sequence: $W = w_1 \dots w_n$, size of tagset $|\mathcal{T}| = m \quad \alpha_0(0) \leftarrow 1$

for $t = 1$ to n

  for $j = 1$ to m

    $\alpha_j(t) \leftarrow \left(\sum_{i=0}^{m} \alpha_i(t-1) a_{ij}\right) b_j(w_t)$

for $i = 1$ to m

  $\beta_i(n) \leftarrow a_{i0}$

for $i = 1$ to m

  $\gamma_i(n) \leftarrow \frac{\alpha_i(n)\beta_i(n)}{\sum_{j=1}^{m} \alpha_j(n)\beta_j(n)}$

for $t = n-1$ to 1

  for $i = 1$ to m

    $\beta_i(t) \leftarrow \sum_{j=1}^{m} \beta_j(t+1) a_{ij} b_j(w_{t+1})$

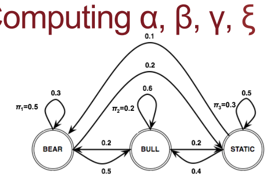
  for $i = 1$ to m

    $\gamma_i(t) \leftarrow \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{m} \alpha_j(t)\beta_j(t)}$

    for $j = 1$ to m

      $\xi_{ij}(t) \leftarrow \frac{\gamma_i(t) a_{ij} b_j(w_{t+1}) \beta_j(t+1)}{\beta_i(t)}$

Computing α, β, γ, ξ



t: 1 2 3 4 5 6

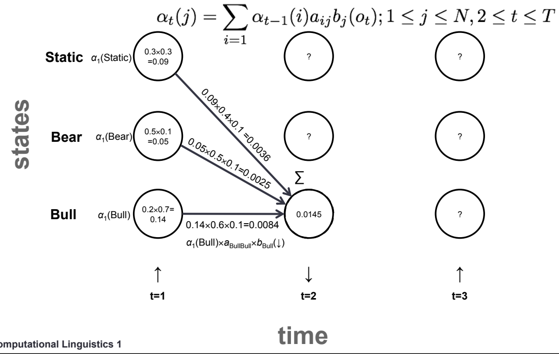
O: ↑ ↓ ↑ ↔ ↓ ↔

$$\begin{bmatrix} P(\uparrow|Bear) = 0.1 \\ P(\downarrow|Bear) = 0.6 \\ P(\leftrightarrow|Bear) = 0.3 \end{bmatrix} \quad \begin{bmatrix} P(\uparrow|Bull) = 0.7 \\ P(\downarrow|Bull) = 0.1 \\ P(\leftrightarrow|Bull) = 0.2 \end{bmatrix} \quad \begin{bmatrix} P(\uparrow|Static) = 0.3 \\ P(\downarrow|Static) = 0.3 \\ P(\leftrightarrow|Static) = 0.4 \end{bmatrix}$$

λ_{stock}

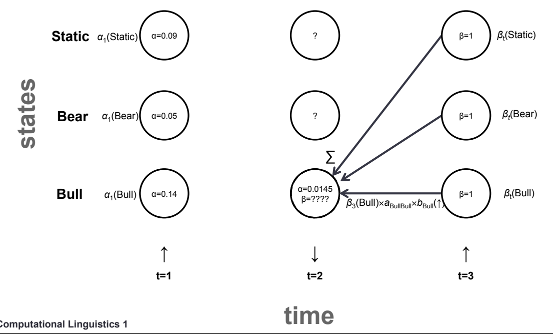
Forward-Backward Algorithm: α

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t); 1 \leq j \leq N, 2 \leq t \leq T$$



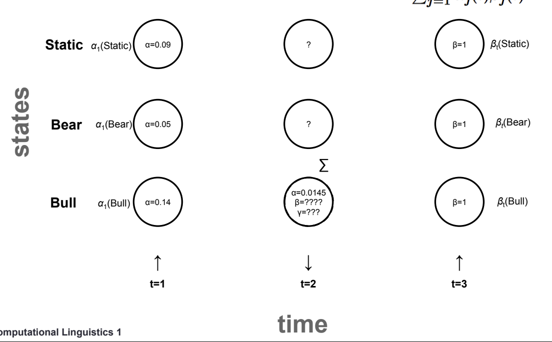
Forward-Backward Algorithm: β

$$\beta_i(t) = \sum_{j=1}^{m} \beta_j(t+1) a_{ij} b_j(w_{t+1})$$



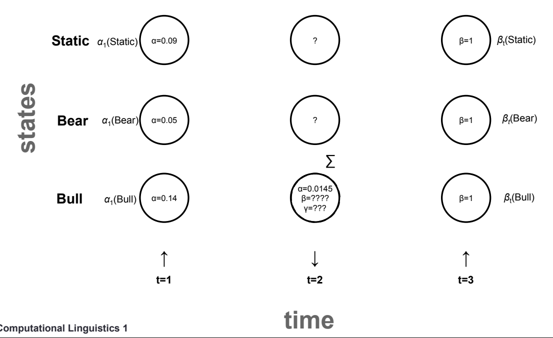
Forward-Backward Algorithm: Ƴ

$$\gamma_i(t) \leftarrow \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{m} \alpha_j(t)\beta_j(t)}$$



Forward-Backward Algorithm: ξ

$$\xi_{ij}(t) \leftarrow \frac{\gamma_i(t) a_{ij} b_j(w_{t+1}) \beta_j(t+1)}{\beta_i(t)}$$



Forward-Backward Algorithm, E-step

word sequence: $W = w_1 \ldots w_n$, size of tagset $|\mathcal{T}| = m$ $\quad \alpha_0(0) \leftarrow 1$

for $t = 1$ to n

  for $j = 1$ to m

    $\alpha_j(t) \leftarrow \left(\sum_{i=0}^{m} \alpha_i(t-1) a_{ij}\right) b_j(w_t)$

for $i = 1$ to m

  $\beta_i(n) \leftarrow a_{i0}$

for $i = 1$ to m

  $\gamma_i(n) \leftarrow \frac{\alpha_i(n)\beta_i(n)}{\sum_{j=1}^{m} \alpha_j(n)\beta_j(n)}$

for $t = n-1$ to 1

  for $i = 1$ to m

    $\beta_i(t) \leftarrow \sum_{j=1}^{m} \beta_j(t+1) a_{ij} b_j(w_{t+1})$

  for $i = 1$ to m

    $\gamma_i(t) \leftarrow \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{m} \alpha_j(t)\beta_j(t)}$

    for $j = 1$ to m

      $\xi_{ij}(t) \leftarrow \frac{\gamma_i(t) a_{ij} b_j(w_{t+1}) \beta_j(t+1)}{\beta_i(t)}$

Forward-Backward, M-step

corpus of N sentences, $W_s = w_1^s \ldots w_{|W_s|}^s$, size of tagset $|\mathcal{T}| = m$

initialize a_{ij}, a_{0j}, a_{j0}, and $b_j(v_k)$ to 0 for all i, j, k

for $i = 1$ to m

  $c(i) \leftarrow \sum_{s=1}^{N} \sum_{t=1}^{|W_s|} \gamma_i^s(t)$

  $a_{0i} \leftarrow \frac{1}{N} \sum_{s=1}^{N} \gamma_i^s(1)$

  $a_{i0} \leftarrow \frac{1}{c(i)} \sum_{s=1}^{N} \gamma_i^s(|W_s|)$

  for $j = 1$ to m

    $a_{ij} \leftarrow \frac{1}{c(i)} \sum_{s=1}^{N} \sum_{t=1}^{|W_s|-1} \xi_{ij}^s(t)$

  for $k = 1$ to $|V|$

    $b_i(v_k) \leftarrow \frac{1}{c(i)} \sum_{s=1}^{N} \sum_{t=1}^{|W_s|} \delta_{w_t^s, v_k} \gamma_i^s(t)$

Agenda: Summary

- Review Viterbi, Forward Algorithms
- Forward-Backward (Baum-Welsch) Algorithm
- Midterm