ILLINOIS INSTITUTE OF TECHNOLOGY

Unsupervised sequence labeling (EM)

CS-585

Natural Language Processing

Derrick Higgins

Recall Our Tagging Questions

- Compute the probability of a text:

$$P_m(W_{1,N})$$

- Compute maximum probability tag sequence:

$$\underset{T_{1,N}}{\operatorname{argmax}} P_m(T_{1,N} | W_{1,N})$$

- Compute maximum likelihood model

$$\underset{\boldsymbol{m}}{\mathbf{arg}\boldsymbol{max}} \boldsymbol{P_m}(\boldsymbol{W_{1,N}})$$

Notation

- N = length of the corpus
- N_t = number of distinct tags
- λ_{ij} = Estimate of $P(t^i \rightarrow t^j)$
  (transition probabilities)
- ϕ_{jk} = Estimate of $P(w^k \mid t^j)$
  (emission probabilities)
- $a_k(i) = P(w_{1,k-1}, t_k = t^i)$
  (from Forward algorithm)
- $b_k(i) = P(w_{k+1,N} | t_k = t^i)$
  (from Backwards algorithm)

Recall: Forward Algorithm

Define $a_k(i) = P(w_{1,k}, t_k = t^i)$

for i in $[1, \ldots, N_t]$ **:**

$a(i) \leftarrow P_m(t_0 \rightarrow t^i) P_m(w_1 | t^i)$

for k in $[2, \ldots, N]$

for j in $[1, \ldots, N_t]$ **:**

$a_k(j) \leftarrow \left(\sum_i a_{k-1}(i) P_m(t^i \rightarrow t^j) \right) P_m(w_k | t^j)$

$P_m\left(W_{1,N}\right) = \sum_i a_N(i)$

Complexity $= O(N_t{}^2 N)$

Recall: Backward Algorithm

Define $b_k(i) = P(w_{k+1,N} | t_k = t^i)$

`for` i in $[1, \ldots, N_t]$ `:`

$\quad b_N(i) \leftarrow 1$

`for` k in $[N-1, \ldots, 1]$

$\quad$ `for` j in $[1, \ldots, N_t]$ `:`

$\qquad b_k(j) \leftarrow \sum_i P_m(t^j \rightarrow t^i)\ P_m(w_{k+1} | t^i)\ b_{k+1}(i)$

$$P_m\left(W_{1,N}\right) = \sum_i P_m(t_0 \rightarrow t^i)\ P_m(w_1 | t^i)\ b_1(i)$$

$$\text{Complexity} = O(N_t^{\,2} N)$$

EM for POS Tagging

1. Start with some initial model (HMM)
2. Compute the probability of each state (tag) for each output symbol, using the current model
3. Use this tagging to revise the model, increasing the probability of the most likely transitions and outputs
4. Repeat until convergence

*Note: **No** labeled training required!*

Estimating transition probabilities

Define $p_k(i,j)$ as prob. of traversing arc $t^i \rightarrow t^j$ at position k given the observations:

Probability of being in state i at index k

$$p_k(i,j) = P(t_k = t^i, t_{k+1} = t^j | W, m)$$

Probability of traversing from state i to state j at index k+1

$$= \frac{P(t_k = t^i, t_{k+1} = t^j, W | m)}{P(W|m)}$$

$$= \frac{a_k(i)\lambda_{ij}\phi_{jW_{k+1}}b_{k+1}(j)}{\sum_{r=1}^{N_t}\sum_{s=1}^{N_t} a_k(r)\lambda_{rs}\phi_{sW_{k+1}}b_{k+1}(s)}$$

Probability of traversing from state j at index k+1 to the end of the sequence

Estimating transition probabilities

Define $p_k(i,j)$ as prob. of traversing arc $t^i \to t^j$ at position k given the observations:

$$p_k(i,j) = P(t_k = t^i, t_{k+1} = t^j | W, m)$$

$$= \frac{P(t_k = t^i, t_{k+1} = t^j, W | m)}{P(W|m)}$$

$$= \frac{a_k(i)\lambda_{ij}\phi_{jW_{k+1}} b_{k+1}(j)}{\sum_{r=1}^{N_t}\sum_{s=1}^{N_t} a_k(r)\lambda_{rs}\phi_{sW_{k+1}} b_{k+1}(s)}$$

Sum over all tag pairs at k, k+1

Derivation

$P(t_k = t^i, t_{k+1} = t^j, W|m)$

$= P(W_{1..k}, t_k = t^i)P(t^i \rightarrow t^j)P(w_{k+1}|t^j)P(W_{(k+2)..N}|t_{k+1} = t^j)$

$= a_i(k)\lambda_{ij}\phi_{jW_{k+1}}b_j(k+1)$

$P(W|m)$

$= \sum_{r=1}^{N_t}\sum_{s=1}^{N_t} P(t_k = t^r, t_{k+1} = t^s, W|m)$

$= \sum_{r=1}^{N_t}\sum_{s=1}^{N_t} a_r(k)\lambda_{rs}\phi_{sW_{k+1}}b_s(k+1)$

Expected transitions

- Define $g_k(i) = P(t_k = t^i \mid W, m)$, then:

$$g_k(i) = \sum_{j=1}^{N_t} p_k(i,j)$$

- Now note that:
  - Expected number of transitions from tag i =

$$\sum_{k=1}^{N} g_k(i)$$

  - Expected transitions from tag i to tag j =

$$\sum_{k=1}^{N} p_k(i,j)$$

Reestimation

$$\lambda'_{ij} = \frac{\text{expected \# of transitions from tag } i \text{ to tag } j}{\text{expected \# of transitions from tag } i}$$

$$= \frac{\sum_{r=1}^{N} p_r(i,j)}{\sum_{r=1}^{N} g_r(i)}$$

$$\phi'_{ik} = \frac{\text{expected \# of observations of } k \text{ for tag } i}{\text{expected \# of transitions from tag } i}$$

$$= \frac{\sum_{w:W_r=w^k} g_r(i)}{\sum_{r=1}^{N} g_r(i)}$$

EM Algorithm for HMM POS

1. Choose initial model = $\langle \boldsymbol{\lambda}, \boldsymbol{\phi} \rangle$
2. Repeat until results don't improve much:
    a. Compute p_t using current model and Forward & Backwards algorithms to compute a and b (Expectation)
    b. Compute new model $\langle \boldsymbol{\lambda}', \boldsymbol{\phi}' \rangle$ (Maximization)

Note: Only guarantees a *local* maximum!

Extensions for HMM POS tagging

- Higher-order models:

$$P(t^{i_1}, \ldots, t^{i_n} \rightarrow t^j)$$

- Incorporating text **features**:
  - Output prob = $P(w^i, \boldsymbol{f}^j \mid t^k)$ where f is a vector of features

    (e.g., capitalization, ends in –*d*, etc.)

- Combining labeled and unlabeled training (initialize with labeled training, then do EM)

Chicago streets

[Notebook]