Computational Linguistics 1

CMSC/LING 723, LBSC 744

Kristy Hollingshead Seitz
Institute for Advanced Computer Studies
University of Maryland

Lecture 12: 11 October 2011

Agenda

- Homework
  - HW2 – graded by Thursday
  - HW3 – due Thursday
- Questions, comments, concerns?
- Unsupervised Learning
  - Expectation Maximization
- Supervised Learning – Discriminative Training
  - Perceptron
  - CRFs

Forward and Backward Probabilities

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

Forward probability:

(probability of seeing initial sequence $w_1 \dots w_t$ 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 $w_{t+1} \dots w_n$ given tag i at time t)

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

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

New Parameters for Forward-Backward

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

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

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

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

Forward-Backward Algorithm, E-step

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

Forward-Backward Algorithm, new model

$$\tilde{b}_i(v_k) = \frac{\sum_{t=1}^{n} \delta_{w_t, v_k} \gamma_i(t)}{\sum_{t=1}^{n} \gamma_i(t)}$$

$$\tilde{a}_{ij} = \frac{\sum_{t=1}^{n-1} \xi_{ij}(t)}{\sum_{t=1}^{n} \gamma_i(t)}$$

$$\tilde{a}_{0j} = \gamma_j(1)$$

$$\tilde{a}_{i0} = \frac{\gamma_i(n)}{\sum_{t=1}^{n} \gamma_i(t)}$$

where $\delta w_t, v_k$ is an indicator function indicating that the word at time t was v_k.

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 $|V|$

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

EM example

fruit *flies* *fast*
NN NNS VB
VB RB
JJ

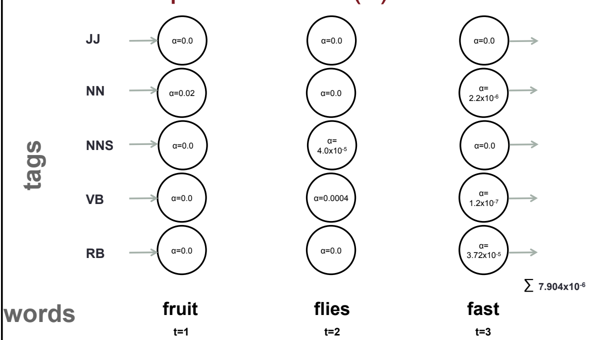
$b_j(w)$

b_2(*fruit*) = P(*fruit* | NN) = 0.1
b_3(*flies*) = P(*flies* | NNS) = 0.01
b_4(*flies*) = P(*flies* | VB) = 0.1
b_4(*fast*) = P(*fast* | VB) = 0.01
b_5(*fast*) = P(*fast* | RB) = 0.3
b_1(*fast*) = P(*fast* | JJ) = 0.05

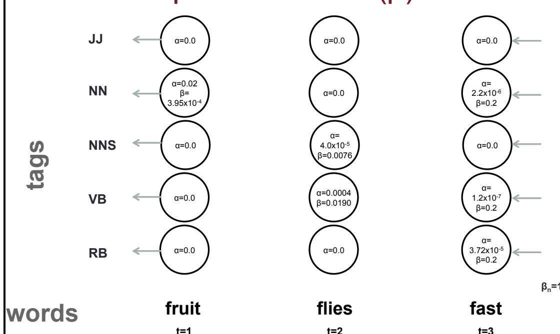
$a_{ij} = \mathrm{P}(\tau_j | \tau_i)$

	j:	0	1	2	3	4	5
i		</s>	JJ	NN	NNS	VB	RB
0	<s>	0	0.3	0.2	0.2	0.2	0.1
1	JJ	0.2	0.1	0.3	0.2	0.1	0.1
2	NN	0.2	0.1	0.2	0.2	0.2	0.1
3	NNS	0.2	0.1	0.1	0.2	0.3	0.1
4	VB	0.2	0.1	0.2	0.2	0	0.3
5	RB	0.2	0.1	0.2	0.1	0.2	0.2

EM example: Forward (α)



EM example: Backward (β)



What is $\beta_0(0)$? $\beta_2(1)a_{02}b_2(\textit{fruit})$ $= 7.904\text{x}10^{-6}$

EM example: Υ

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

t	j	lbl	$\alpha_i(t)$	$\beta_i(t)$	$\alpha_i(t)\beta_i(t)$	$\gamma_i(t)$
1	2	NN	0.02	0.0003952	0.000007904	1
2	3	NNS	0.00004	0.0076	0.000000304	0.038
	4	VB	0.0004	0.019	0.0000076	0.962
3	1	JJ	0.0000022	0.2	0.00000044	0.056
	4	VB	0.00000012	0.2	0.000000024	0.003
	5	RB	0.0000372	0.2	0.00000744	0.941

EM example: ξ

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

t	i	i lbl	j	j lbl	$\gamma_i(t)$	a_{ij}	$b_j(w_{t+1})$	$\beta_j(t+1)$	$\beta_i(t)$	$\xi_{ij}(t)$
1	2	NN	3	NNS	1	0.2	0.01	0.0076	0.0003952	0.0385
			4	VB	1	0.2	0.1	0.019	0.0003952	0.9615
2	3	NNS	1	JJ	0.038	0.1	0.05	0.2	0.0076	0.005
			4	VB	0.038	0.3	0.01	0.2	0.0076	0.003
			5	RB	0.038	0.1	0.3	0.2	0.0076	0.03
	4	VB	1	JJ	0.962	0.1	0.05	0.2	0.019	0.051
			4	VB	0.962	0	0.01	0.2	0.019	0
			5	RB	0.962	0.3	0.3	0.2	0.019	0.911

Forward-Backward Algorithm, new model

$$\tilde{b}_i(v_k) = \frac{\sum_{t=1}^{n} \delta_{w_t,v_k} \gamma_i(t)}{\sum_{t=1}^{n} \gamma_i(t)}$$

$$\tilde{a}_{ij} = \frac{\sum_{t=1}^{n-1} \xi_{ij}(t)}{\sum_{t=1}^{n} \gamma_i(t)}$$

$$\tilde{a}_{0j} = \gamma_j(1)$$

$$\tilde{a}_{i0} = \frac{\gamma_i(n)}{\sum_{t=1}^{n} \gamma_i(t)}$$

where $\delta w_t, v_k$ is an indicator function indicating that the word at time t was v_k.

Forward-Backward, M-step

corpus of N sentences, $W_s = w_1^s \dots w_{|W_s|}^s$, 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 $|V|$
    $b_i(v_k) \leftarrow \frac{1}{c(i)} \sum_{s=1}^{N} \sum_{t=1}^{|W_s|} \delta_{w_t^s, v_k} \gamma_i^s(t)$

Training versus held-out data

- Train a model from training data
- Perform EM until convergence
- If training data is used, this doesn't work (λ ← 1)
  - Already have maximum likelihood solution for training data
  - If we now try to maximize the likelihood . . .
- Hold aside some data from training
  - Converge on held-out data
- Prevents over-training

Agenda

- Homework
- Unsupervised Learning
  - Expectation Maximization
- Supervised Learning – Discriminative Training
  - Perceptron
  - CRFs

Discriminative Training

- Statistical model training involves maximizing some *objective* function
- For an HMM, we use maximum likelihood training
  - Maximize the probability of the training set
- Reduction in errors is the true objective of learning
- Another option is to try to directly optimize error rate or some other closely related objective
- Consider not just truth, but also other candidates

Perceptron

- One approach that has been around since late 60s is the perceptron
- Basic idea:
  - Find the best scoring analysis (e.g. POS tag sequence)
  - Make its score lower, by penalizing its *features*
  - Make the score of the truth better, by rewarding its features
  - Go onto the next example

Perceptron Initialization

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

input (word) vocabulary: $v_i \in V$ for $1 \leq i \leq k$

output (tag) vocabulary: $\tau_j \in \mathcal{T}$ for $1 \leq j \leq m$

$$\begin{aligned}
\text{Let } b_j(v_i) &= \mathrm{P}(v_i|\tau_j) &= 0\\
\text{Let } a_{ij} &= \mathrm{P}(\tau_j|\tau_i) &= 0\\
\text{Let } a_{0j} &= \mathrm{P}(\tau_j|\text{<s>}) &= 0\\
\text{Let } a_{i0} &= \mathrm{P}(\text{</s>}|\tau_i) &= 0
\end{aligned}$$

Let $\Phi(x, y) = [b_j(v_i), a_{ij}]$

Perceptron example

- Training set:
  (PRP Her) (NN cat) (VB loves) (NN fruit) (. .)
  (PRP I) (RB always) (VB fast) (IN during) (NNP Lent) (. .)
  (DT That) (NN airplane) (VB flies) (RB fast) (. .)
  (PRP His) (NN kitchen) (VB has) (NN fruit) (NNS flies) (. .)

Perceptron Example

truth (y_i): (PRP Her) (NN cat) (VB loves) (NN fruit) (. .)
guess (z_i): (AUX Her) (AUX cat) (AUX loves) (AUX fruit) (AUX .)

Perceptron Example

truth (y_i): (PRP Her) (NN cat) (VB loves) (NN fruit) (. .)
guess (z_i): (AUX Her) (AUX cat) (AUX loves) (AUX fruit) (AUX .)

Increment features of truth, decrement features of guess.

$\mathrm{P}(\text{AUX}|\text{<s>}) = -1$	$\mathrm{P}(\text{</s>}|\text{AUX}) = -1$	
$\mathrm{P}(\text{PRP}|\text{<s>}) = 1$	$\mathrm{P}(\text{</s>}|.) = 1$	
$\mathrm{P}(\text{AUX}|\text{AUX}) = -4$	$\mathrm{P}(\mathit{Her}|\text{AUX}) = -1$	$\mathrm{P}(\mathit{Her}|\text{PRP}) = 1$
$\mathrm{P}(\text{NN}|\text{PRP}) = 1$	$\mathrm{P}(\mathit{cat}|\text{AUX}) = -1$	$\mathrm{P}(\mathit{cat}|\text{NN}) = 1$
$\mathrm{P}(\text{VB}|\text{NN}) = 1$	$\mathrm{P}(\mathit{loves}|\text{AUX}) = -1$	$\mathrm{P}(\mathit{loves}|\text{VB}) = 1$
$\mathrm{P}(\text{NN}|\text{VB}) = 1$	$\mathrm{P}(\mathit{fruit}|\text{AUX}) = -1$	$\mathrm{P}(\mathit{fruit}|\text{NN}) = 1$
$\mathrm{P}(.|\text{NN}) = 1$	$\mathrm{P}(.|\text{AUX}) = -1$	$\mathrm{P}(.|.) = 1$

Perceptron Example

truth (y_i): (PRP I) (RB always) (VB fast) (IN during) (NNP Lent) (. .)
guess (z_i): (PRP I) (AUX always) (DT fast) (AUX during) (DT Lent) (. .)

Perceptron Example

truth (y_i): (PRP I) (RB always) (VB fast) (IN during) (NNP Lent) (. .)
guess (z_i): (PRP I) (AUX always) (DT fast) (AUX during) (DT Lent) (. .)

Increment features of truth, decrement features of guess.

$\mathrm{P}(\text{PRP}|\text{<s>}) = 0$	$\mathrm{P}(\text{</s>}|.) = 0$
$\mathrm{P}(\text{PRP}|\text{<s>}) = 1$	$\mathrm{P}(\text{</s>}|.) = 1$
$\mathrm{P}(\text{AUX}|\text{PRP}) = -1$	
$\mathrm{P}(\text{RB}|\text{PRP}) = 1$	
$\vdots$	

etc.

Perceptron Example

truth (y_i): (PRP Her) (NN cat) (VB loves) (NN fruit) (. .)
guess (z_i): (PRP Her) (RB cat) (VB loves) (NN fruit) (. .)

Perceptron: Notes

- Because this technique is optimizing (sequence) error rate, it does not involve a normalization factor
- Thus, it will overtrain
  - i.e. it will do very well on the training set, but not so well on new data, like unsmoothed maximum likelihood
  - Techniques exist for controlling overtraining, such as regularization, voting, and averaging
- Perceptron models outperform maximum likelihood–optimized models on a range of tasks
  - POS-tagging, NP-chunking

Agenda: Summary

- Review Forward-Backward algorithm
- Unsupervised learning
  - Apply EM
- Begin discussion of discriminative supervised learning
  - Perceptron
- Midterm – review next lecture