

- Transformation-based tagger
- HMM-based tagger

## Maximum Entropy and Log-linear Models

Regina Barzilay  
EECS Department  
MIT

October 1, 2004

- Initialization: a list of allowable part of speech tags
- Transformations: Change the tag of a word from  $\chi$  to  $Y$  in context  $C$ , where  $\gamma \in \chi$ .  
*Example: "From NN\_VBP to VBP if previous tag is NNS"*
- Scoring criterion:

$$R = \operatorname{argmax}_{Z \in \chi, Z \neq Y} \frac{\operatorname{freq}(X)}{\operatorname{freq}(Z) * \operatorname{incontext}(Z, C)}$$

$$\operatorname{score} = \operatorname{incontext}(\gamma, C) - \frac{Y}{\operatorname{freq}(R)} * \operatorname{incontext}(R, C)$$

## Leftovers: POS distribution

---

The number of word types in Brown corpus by degree of ambiguity

<b>Unambiguous (1 tag)</b>	<b>35,340</b>
<b>Ambiguous (2-7 tags)</b>	<b>4,100</b>
2 tags	3,764
3 tags	264
4 tags	61
5 tags	12
6 tags	2
7 tags	1

## The General Problem

---

- We have some **input domain**  $\chi$
- We have some **label set**  $\gamma$
- Goal: learn a **conditional probability**  $P(y|x)$  for any  $x \in \chi$  and  $y \in \gamma$

## Today

---

- Maximum entropy models
- Connection to log-linear models
- Optimization methods

## POS tagging: Representation

---

*Our/PRP\$ enemies/NNS are/VBP innovative/JJ and/CC resourceful/JJ ,/, and/CC so/RB are/VB we/PRP ?/?.*

- History is a 4-tuples  $(t_1, t_2, w_{[1:n]}, i)$
- $t_1, t_2$  are the previous two tags
- $w_{[1:n]}$  are the  $n$  words in the input sentence
- $i$  is the index of the word being tagged

$\chi$  is the set of all possible histories

## POS tagging

---

*Our/PRP\$ enemies/NNS are/VBP innovative/JJ and/CC resourceful/JJ ,/, and/CC so/RB are/VB we/PRP ?/?.*

- Input domain:  $\chi$  is the set of possible histories
- Label set:  $\gamma$  is the set of all possible tags
- Goal: learn a **conditional probability**  $P(tag|history)$

## POS Representation

- Word/tag features for all word/tag pairs:

$$f_{55}(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ is "our" and } t = PRP \\ 0 & \text{otherwise} \end{cases}$$

- Spelling features for all prefixes/suffixes of certain length:

$$f_{70}(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ ends in "ing" and } t = VBG \\ 0 & \text{otherwise} \end{cases}$$

- Contextual features:

$$f_{112}(h, t) = \begin{cases} 1 & \text{if previous word } w_i \text{ is "the" and } t = Vt \\ 0 & \text{otherwise} \end{cases}$$

## Feature Vector Representation

- A **feature** is a function  $f : \chi * \gamma \rightarrow 0$

$$f(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ is "are" and } t = VBP \\ 0 & \text{otherwise} \end{cases}$$

- We have  $m$  features  $f_k$  for  $k = 1 \dots m$

## Maximum Entropy: Motivating Example

Estimate probability distribution  $p(a, b)$ , given the constraint:  
 $p(x, 0) + p(y, 0) = 0.6$ , where  $a \in \{x, y\}$  and  $b \in 0, 1$ .

$p(a, b)$	0	1	
x	?	?	
y	?	?	
total	0.6		1.0

## POS Representation

For a given history  $x \in X$ , each label in  $\gamma$  is mapped to a different feature vector

$$f((PRP, NNS, [Our, \dots], 2), VBP) = 100110010$$

$$f((PRP, NNS, [Our, \dots], 2), JJ) = 111001111$$

$$f((PRP, NNS, [Our, \dots], 2), NP) = 001011010$$

...

Goal: learn a **conditional probability**  $P(\text{tag}|\text{history})$

## Another Way To Satisfy Constraints

$p(a, b)$	0	1	
x	0.3	0.2	
y	0.3	0.2	
total	0.6		1.0

## Representing Evidence

Constraint: *observed expectation* of each feature has to be the same as *the model's expectation* of the feature:

$$E_p f_j = E_{p'} f_j (j = 1 \dots m),$$

$$E_p f_j = \sum_{x \in \{\chi^* \gamma\}} p(x) f_j(x) \text{ (model's expectation)}$$

$$E_{p'} f_j = \sum_{x \in \{\chi^* \gamma\}} p'(x) f_j(x) \text{ (observed expectation)}$$

## One Way To Satisfy Constraints

$p(a, b)$	0	1	
x	0.5	0.1	
y	0.1	0.3	
total	0.6		1.0

## Maximum Entropy Modeling

Given a set of training examples, we wish to find a distribution which:

- satisfies the input constraints
- maximizes the uncertainty

... in making inference on the basis of partial information we must use the probability distribution which has maximum entropy subject to whatever is known. This is the only unbiased assignment we can make; to use any other would amount to arbitrary assumption of information which by hypothesis we do not have. *Jaynes, 1957*

## Outline

---

- We will first show that  
 $p^*(x) = \pi \prod_{j=1}^k \alpha_j^{f_j(x)}$ ,  $0 < \alpha_j < \infty$ , where  $\pi$  is a normalization constant and the  $\alpha$ 's are the model parameters
- Then, we will consider an estimation procedure for finding the  $\alpha$ 's

## Principle of Maximum Entropy

---

$$P = \{p | E_p f_j = E_{p'} f_j, j = \{1 \dots m\}\}$$
$$p^* = \operatorname{argmax}_{p \in P} H(p)$$

## Relative Entropy (Kullback-Liebler Distance)

- **Definition:** The relative entropy  $D$  between two probability distributions  $p$  and  $q$  is given by:  
$$D(p, q) = \sum_{x \in \chi \times \gamma} p(x) \log \frac{p(x)}{q(x)}$$
- **Lemma 1:** For any two probability distributions  $p$  and  $q$ ,  $D(p, q) \geq 0$ , and  $D(p, q) = 0$  if and only if  $p = q$

## Notations

---

$\chi$  is the set of possible histories  
 $\gamma$  is the set of all possible tags  
 $S$  finite training sample of events  
 $p'(x)$  observed probability of  $x$  in  $S$   
 $p(x)$  the model's probability of  $x$   
 $f_j$  function of type  $\chi \times \gamma \rightarrow \{0, 1\}$   
 $E_p f_j = \sum_{x \in \{\chi \times \gamma\}} p(x) f_j(x)$   
 $E_{p'} f_j = \sum_{x \in \{\chi \times \gamma\}} p'(x) f_j(x)$   
 $P = \{p | E_p f_j = E_{p'} f_j, j = \{1 \dots m\}\}$   
 $Q = \{p | p(x) = \pi \prod_{j=1}^k \alpha_j^{f_j(x)}\}$   
 $H(p) = - \sum_{x \in \chi \times \gamma} p(x) \log p(x)$   
 $L(p) = \sum_{x \in \chi \times \gamma} p'(x) \log p(x)$

Let  $p \in P$ ,  $q \in Q$  and  $p^* \in P \cap Q$ :

$$\begin{aligned}
 & D(p, p^*) + D(p^*, q) \\
 &= \sum_x p(x) \log p(x) - \sum_x p(x) \log p^*(x) + \sum_x p^*(x) \log p^*(x) - \sum_x p^*(x) \log q(x) \\
 &= \sum_x p(x) \log p(x) - \sum_x p(x) \log p^*(x) + \sum_x p(x) \log p^*(x) - \sum_x p(x) \log q(x) \\
 &= \sum_x p(x) \log p(x) - \sum_x p(x) \log q(x) = D(p, q)
 \end{aligned}$$

## Pythagorean Property

**Lemma 2 (Pythagorean Property):** If  $p \in P$  and  $q \in Q$ , and  $p^* \in P \cap Q$ , then

$$D(p, q) = D(p, p^*) + D(p^*, q).$$

Proof: For any  $r, s \in P$ , and  $t \in Q$

$$\begin{aligned}
 & \sum_x r(x) \log t(x) \\
 &= \sum_x r(x) [\log \pi + \sum_j f_j(x) \log \alpha_j] \\
 &= \log \pi \left[ \sum_x r(x) \right] + \left[ \sum_j \log \alpha_j \sum_x r(x) f_j(x) \right] \\
 &= \log \pi \left[ \sum_x s(x) \right] + \left[ \sum_j \log \alpha_j \sum_x s(x) f_j(x) \right] \\
 &= \sum_x s(x) [\log \pi + \sum_j f_j(x) \log \alpha_j] = \sum_x s(x) \log t(x)
 \end{aligned}$$

## The Maximum Likelihood Solution

**Theorem 2** If  $p^* \in P \cap Q$ , then  $p^* = \operatorname{argmax}_{q \in Q} L(q)$ . Furthermore  $p^*$  is unique.

**Proof.** Let  $\bar{p}(x)$  be the observed distribution of  $x$  in the sample  $S$ ,  $\forall x \in \epsilon$ . Clearly  $\bar{p} \in P$ .

Suppose  $q \in Q$  and  $p^* \in P \cap Q$ .

- Show that  $L(q) \leq L(p^*)$ :

By Lemma 2,

$$D(\bar{p}, q) = D(\bar{p}, p^*) + D(p^*, q)$$

and by Lemma 1,

$$D(\bar{p}, q) \geq D(\bar{p}, p^*)$$

$$-H(\bar{p}) - L(q) \geq -H(\bar{p}) - L(p^*)$$

$$L(q) \leq L(p^*)$$

- Show  $p^*$  is unique:

$$L(q) = L(p^*) \implies D(\bar{p}, q) = D(\bar{p}, p^*) \implies D(p^*, q) = 0 \implies p^* = q$$

## The Maximum Entropy Solution

**Theorem 1** If  $p^* \in P \cap Q$  then  $p^* = \operatorname{argmax}_{p \in P} H(p)$ . Furthermore,  $p^*$  is unique.

**Proof.** Suppose  $p \in P$  and  $p^* \in P \cap Q$ . Let  $u \in Q$  be the uniform distribution so that  $\forall x \in \epsilon u(x) = \frac{1}{|\epsilon|}$ .

- Show that  $H(p) \leq H(p^*)$ :

By Lemma 2,

$$D(p, u) = D(p, p^*) + D(p^*, u)$$

and by Lemma 1,

$$D(p, u) \geq D(p^*, u)$$

$$-H(p) - \log \frac{1}{|\epsilon|} \geq -H(p^*) - \log \frac{1}{|\epsilon|}$$

$$H(p) \leq H(p^*)$$

- Show  $p^*$  is unique:

$$H(p) = H(p^*) \implies D(p, u) = D(p^*, u) \implies D(p, p^*) = 0 \implies p = p^*$$

## Generative Iterative Scaling

(Darroch&Ratcliff, 1972)

- Goal: Find distribution of the form  $\pi \prod_{j=1}^k \alpha_j^{f_j}(x)$  that obeys the following constraints:

$$E_p f_j = E_{p'} f_j$$

- GIS constraints:
  - $\forall x \in \chi \times \gamma \sum_{j=1}^m f_j(x) = C$ , where C is a constant (add correctional feature)
  - $\forall x \in \chi \times \gamma \exists f_j f_j(x) = 1$

## Duality Theorem

- There is a unique distribution  $p^*$ 
  1.  $p^* \in P \cap Q$
  2.  $p^* = \operatorname{argmax}_{p \in P} H(p)$  (Max-ent solution)
  3.  $p^* = \operatorname{argmax}_{q \in Q} L(q)$  (Max-likelihood solution)
- Implications:
  - The maximum entropy solution can be written in log-linear form
  - Finding the maximum-likelihood solution also gives the maximum entropy solution

## Computation

- $E_{p'} f_j = \sum_{i=1}^N p'(a_i, b_i) f_j(a_i, b_i) = \frac{1}{N} * \sum_{i=1}^N f_j(a_i, b_i)$ , where  $S = \{(a_1, b_1), \dots, (a_N, b_N)\}$  is a training sample
- $E^{(n)} f_j = \sum_{x \in \chi \times \gamma} p^{(n)}(a, b) f_j(a, b)$  (dreadful!)  
 $E^{(n)} f_j \approx \sum_{i=1}^N p'(b_i) \sum_{a \in \mathcal{S}} p^{(n)}(a|b_i) f_j(a, b_i)$  (approximation)

Running time:  $O(NPA)$ , where  $N$  is the training set size,  $P$  is the number of predictions, and  $A$  is the average number of features that are active for a given event  $(a, b)$

## GIS (cont.)

**Theorem:** The following procedure will converge to  $p^* \in P \cap Q$

$$\alpha_j^{(0)} = 1$$

$$\alpha_j^{(n+1)} = \alpha_j^{(n)} \left[ \frac{E'(f_j)}{E^{(n)}(f_j)} \right]^{\frac{1}{C}},$$

where  $E^{(n)} f_j = \sum_{x \in \{\chi * \gamma\}} p^{(n)}(x) f_j(x)$

$$p^{(n)}(x) = \pi \prod_{j=1}^{m+1} (\alpha_j^{(n)})^{f_j(x)}$$

## Summary

---

- Modeling conditional probabilities with log-linear models
- Maximum-entropy properties of log-linear models
- Optimization via iterative scaling

Some implementations:

- <http://nlp.stanford.edu/downloads/classifier.shtml>
- <http://maxent.sourceforge.net>

## ME classifiers

---

- Can handle lots of features
- Sparsity is an issue
  - apply smoothing and feature selection
- Feature interaction
  - ME classifiers do not assume feature independence
  - However, they do not explicitly model feature interaction