SFU NatLangLab

Natural Language Processing

Anoop Sarkar
anoopsarkar.github.io/nlp-class

Simon Fraser University

October 5, 2023

Natural Language Processing

Anoop Sarkar
anoopsarkar.github.io/nlp-class

Simon Fraser University

Part 1: Linear models for Tagging

Tagging tasks in NLP

Log-linear models for Tagging

Tagging Tasks

Tagged Sequences

a b e e a f h j $\Rightarrow$ a/Y b/Z e/Y e/Y a/Z f/X h/Z j/Y

Example 1: Part-of-speech tagging

Profits/N soared/V at/P Boeing/N Co./N ,/, easily/ADV topping/V forecasts/N on/P Wall/N Street/N ,/, as/P their/POSS CEO/N Alan/N Mulally/N announced/V first/ADJ quarter/N results/N ./.

Example 2: Named Entity Recognition

Profits/O soared/O at/O Boeing/B-CO Co./I-CO ,/O easily/O topping/O forecasts/O on/O Wall/B-LOC Street/I-LOC ,/O as/O their/O CEO/O Alan/B-PER Mulally/I-PER announced/O first/O quarter/O results/O ./O

Notation for Tagging Tasks

- Set of possible input words: $\mathcal{V}$
- Set of possible tags: $\mathcal{T}$
- Word sequence: $x_{[1:n]} = [x_1, \ldots, x_n]$
- Tag sequence: $t_{[1:n]} = [t_1, \ldots, t_n]$
- Training data is N tagged sentences, the i^{th} sentence has length n_i:

$$(x^{(i)}_{[1:n]}, t^{(i)}_{[1:n]}) \text{ for } i = 1, \ldots, n$$

Independence Assumptions for Tagging

Chain Rule

$$P(t_{[1:n]} \mid x_{[1:n]}) = \prod_{j=1}^{n} P(t_j \mid t_{j-1}, \ldots, t_1, x_{[1:n]}, j)$$

Make independence assumptions

$$P(t_{[1:n]} \mid x_{[1:n]}) \approx \prod_{j=1}^{n} P(t_j \mid t_{j-1}, x_{[1:n]}, j)$$

j is the word being tagged.

We model the conditional probability directly: no Bayes Rule here.

Questions

- Split up $P(t_j \mid t_{j-1}, x_{[1:n]}, j)$ into parameters?
- How to find $\arg\max_{t_{[1:n]}} P(t_{[1:n]} \mid x_{[1:n]})$?

Tagging tasks in NLP

Log-linear models for Tagging

Representation: finding the right parameters

Problem: Predict ?? using context, $P(?? \mid \texttt{context})$

Profits/N soared/V at/P Boeing/?? Co. , easily topping forecasts on Wall Street , as their CEO Alan Mulally announced first quarter results .

Representation: history

- A history is a 3-tuple: $(t_{-1}, x_{[1:n]}, i)$
- t_{-1} is the previous tag (we are assuming a bigram model)
- $x_{[1:n]}$ are the n words in the input
- i is the index of the word being tagged
- For example, for $x_4 =$ Boeing:
  - $t_{-1} =$ P
  - $x_{[1:n]} =$ (Profits, soared, ..., results, .)
  - $i = 4$

Feature-vectors over history-tag pairs

Take a history, tag pair (h, t)

$f_k(h, t)$ for $k = 1, \ldots, m$ are **feature functions** representing the tagging decision.

Example: Part-of-speech tagging [Ratnaparkhi 1996]

$$f_{100}(h, t) = \begin{cases} 1 & \text{if current word } x_i \text{ is } \texttt{Boeing} \text{ and } t = \texttt{N} \\ 0 & \text{otherwise} \end{cases}$$

$$f_{101}(h, t) = \begin{cases} 1 & \text{if } t_{-1} \text{ is } \texttt{P} \text{ and } t = \texttt{N} \\ 0 & \text{otherwise} \end{cases}$$

Log linear model for Tagging

- Let there be m features, $f_k(\mathbf{x}, \mathbf{y})$ for $k = 1, \ldots, m$
- $\mathbf{x} = x_{[1:n]}$ and $\mathbf{y} = t_{[1:n]}$
- Define a parameter vector $\mathbf{w} \in \mathbb{R}^m$
- Each $(\mathbf{x}, \mathbf{y})$ pair is mapped to score:

$$s(\mathbf{x}, \mathbf{y}) = \sum_k w_k \cdot f_k(\mathbf{x}, \mathbf{y})$$

- Using inner product notation:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, \mathbf{y}) &= \sum_k w_k \cdot f_k(\mathbf{x}, \mathbf{y}) \\ s(\mathbf{x}, \mathbf{y}) &= \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, \mathbf{y}) \end{aligned}$$

- To get a probability from the score: Renormalize!

$$\Pr(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) = \frac{exp\left(s(\mathbf{x}, \mathbf{y})\right)}{\sum_{\mathbf{y'}} exp\left(s(\mathbf{x}, \mathbf{y'})\right)}$$

Feature functions for Tagging

Problem

- We have defined a log-linear model using feature functions: $\mathbf{f(x,y)}$
- We have defined parameters using a history h so feature functions are: $\mathbf{f}(h, t)$

Locally normalized log-linear taggers

Conditional Distribution over history, tag pair (h, t)

$$\log \Pr(t \mid h) = \mathbf{w} \cdot \mathbf{f}(h, t) - log \sum_{t'} \exp\left(\mathbf{w} \cdot \mathbf{f}(h, t')\right)$$

- $\mathbf{f}(h, t)$ is a vector of feature functions
- $\mathbf{w}$ is the weight vector

Local normalization for tagging

- Word sequence: $x_{[1:n]}$ and tag sequence: $t_{[1:n]}$
- Histories $h_i = (t_{i-1}, x_{[1:n]}, i)$

$$\log \Pr(t_{[1:n]} \mid x_{[1:n]}) = \sum_{i=1}^{n} \log \Pr(t_i \mid h_i)$$

Globally normalized log-linear taggers

Global feature function $\mathbf{\Phi}(\mathbf{x}, \mathbf{y})$

- Word sequence: $\mathbf{x} = x_{[1:n]}$ and tag sequence: $\mathbf{y} = t_{[1:n]}$
- From *local* histories $h_i = (t_{i-1}, x_{[1:n]}, i)$ to global Φ values:

$$\Phi_k(x_{[1:n]}, t_{[1:n]}) = \sum_{i=1}^{n} f_k(h_i, t_i)$$

- $\mathbf{\Phi}(\mathbf{x}, \mathbf{y}) = (\Phi_1, \Phi_2, \ldots, \Phi_m)$ is a *global* feature vector
- $\mathbf{w}$ is the weight vector for $\mathbf{\Phi}$

Global normalization for tagging

$$\log \Pr(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) = \mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}, \mathbf{y}) - \log \sum_{\mathbf{y'}} exp\left(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}, \mathbf{y'})\right)$$

Conditional Random Field

Global normalization for tagging

$$\log \Pr(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) = \mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}, \mathbf{y}) - \log \sum_{\mathbf{y'}} exp\left(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}, \mathbf{y'})\right)$$

- This model is also called a conditional random field (CRF)

Algorithms for training and decoding

- Global normalization could be expensive: requires sum over exponentially many terms $\mathbf{y'}$
- Finding $\arg\max_{\mathbf{y}} \log \Pr(\mathbf{y} \mid \mathbf{x})$ can be accomplished using the Viterbi algorithm.
- Training: finding the weight vector $\mathbf{w}$ can be done using a variant of the Forward algorithm.

Acknowledgements

Many slides borrowed or inspired from lecture notes by Michael Collins, Chris Dyer, Kevin Knight, Chris Manning, Philipp Koehn, Adam Lopez, Graham Neubig, Richard Socher and Luke Zettlemoyer from their NLP course materials.

All mistakes are my own.

A big thank you to all the students who read through these notes and helped me improve them.