NLP: Maximum Entropy Markov Models

Dan Garrette
dhg@cs.utexas.edu

October 24, 2013

1 Features

Why do we like feature?

- Give us additional, useful information.
- Parts of speech: prefixes, suffixes, capitalization, word shape, is a number, ...

Why do we like sequence models?

- Nouns and Adjectives more likely to follow Determiners
- "I enjoy walks". Typically, "walks" is a verb, but not here since "enjoy" is definitely a verb.

2 Linear Regression

- Each feature f_i has an associated weight w_i
- Assign a real value $y \in (-\infty, \infty)$ based on features

$$
\begin{aligned}
y &= w_0 + w_1 \times f_1 + w_2 \times f_2 + w_3 \times f_3 + ... \\
&= \sum_{i=0}^{N} w_i \times f_i = \vec{\mathbf{w}} \cdot \vec{\mathbf{f}} \quad \text{(dot product)}
\end{aligned}
$$

- For a particular instance j:

$$y_{pred}^{(j)} = \sum_{i=0}^{N} w_i \times f_i^{(j)}$$

- Learning: choose weights W that minimize the sum-squared error:

$$cost(W) = \sum_{j=0}^{M} (y_{pred}^{(j)} - y_{obs}^{(j)})$$

3 Logistic Regression

- For many NLP applications, we don't want a real value output, we want a *classification.*
- Moreover, we want to assign a *probability* to each class
- Want to be able to use wighted features
- But, can't simply apply linear regression because it doesn't give us probabilities

Binary Classification

- Need $p(y = \text{true} \mid x)$
- For instance x, we want to make use of $\sum_{i=0}^{N} w_i \times f_i$
- Maybe a ratio? $\frac{p(y=\text{true}|x)}{p(y=\text{false}|x)} = \frac{p(y=\text{true}|x)}{1-p(y=\text{true}|x)}$, but this yields a value between 0 (definitely false) and ∞ (definitely true)
- Logarithm gets us a value between $-\infty$ and ∞: $\ln(\frac{p(y=\text{true}|x)}{1-p(y=\text{true}|x)}) = \vec{\mathbf{w}} \cdot \vec{\mathbf{f}}$
- Exponentiating both sides gives us: $\frac{p(y=\text{true}|x)}{1-p(y=\text{true}|x)} = e^{\vec{\mathbf{w}} \cdot \vec{\mathbf{f}}}$
- Classify with

$$\begin{aligned}
p(y = \text{true} \mid x) &> p(y = \text{false} \mid x) \\
\frac{p(y = \text{true} \mid x)}{p(y = \text{false} \mid x)} &> 1 \\
\frac{p(y = \text{true} \mid x)}{1 - p(y = \text{true} \mid x)} &> 1 \\
e^{\vec{\mathbf{w}} \cdot \vec{\mathbf{f}}} &> 1 \quad \text{from above} \\
\vec{\mathbf{w}} \cdot \vec{\mathbf{f}} &> 0 \\
\sum_{i=0}^{N} w_i \times f_i &> 0
\end{aligned}$$

Learning

$$\begin{aligned}
\hat{w} &= argmax_w \prod_i p(y^{(i)} \mid x^{(i)}) \\
&= argmax_w \prod_i \{p(y^{(i)} = 1 \mid x^{(i)}) \text{ for } y^{(i)} = 1 \quad \text{OR} \quad p(y^{(i)} = 0 \mid x^{(i)}) \text{ for } y^{(i)} = 0
\end{aligned}$$

- convex optimization
- gradient ascent or L-BFGS