Natural Language Processing
MEMMs and CRFs

Felipe Bravo-Marquez

September 5, 2019

MEMMs

- Maximum-entropy Markov models (MEMMs) make use of log-linear models for sequence labeling tasks.
- Our goal will be to model the conditional distribution

$$P(s_1, s_2 \ldots, s_m | x_1, \ldots, x_m)$$

  where each x_j for $j = 1 \ldots m$ is the j'th input symbol (for example the j'th word in a sentence), and each s_j for $j = 1 \ldots m$ is the j'th state.
- We'll use S to denote the set of possible states; we assume that S is a finite set. [1]

[1] These slides are based on lecture notes of Michael Collins http://www.cs.columbia.edu/~mcollins/

MEMMs

- For example, in part-of-speech tagging of English, S would be the set of all possible parts of speech in English (noun, verb, determiner, preposition, etc.).
- Given a sequence of words $x_1, \ldots, x_m$, there are k^m possible part-of-speech sequences $s_1, \ldots, s_m$, where $k = |S|$ is the number of possible parts of speech.
- We'd like to estimate a distribution over these k^m possible sequences.

MEMMs

- In a first step, MEMMs use the following decomposition:

$$\begin{aligned} P(s_1, s_2 \ldots, s_m | x_1, \ldots, x_m) &= \prod_{i=1}^{m} P(s_i | s_1 \ldots, s_{i-1}, x_1, \ldots, x_m) \\ &= \prod_{i=1}^{m} P(s_i | s_{i-1}, x_1, \ldots, x_m) \end{aligned} \tag{1}$$

- The first equality is exact (it follows by the chain rule of conditional probabilities).
- The second equality follows from an independence assumption, namely that for all i,

$$P(s_i | s_1 \ldots, s_{i-1}, x_1, \ldots, x_m) = P(s_i | s_{i-1}, x_1, \ldots, x_m)$$

MEMMs

- Hence we are making an assumption here that is similar to the Markov assumption in HMMs.
- The state in the i'th position depends only on the state in the $(i-1)$'th position.
- Having made these independence assumptions, we then model each term using a log-linear model (or Softmax):

$$P(s_i|s_{i-1}, x_1, \ldots, x_m) = \frac{\exp(\vec{w} \cdot \vec{\phi}(x_1, \cdots, x_m, i, s_{i-1}, s_i))}{\sum_{s' \in S} \exp(\vec{w} \cdot \vec{\phi}(x_1, \cdots, x_m, i, s_{i-1}, s'))} \tag{2}$$

MEMMs

Here $\vec{\phi}(x_1, \cdots, x_m, i, s_{i-1}, s_i)$ is a feature vector where:

- $x_1, \cdots, x_m$ is the entire sentence being tagged.
- i is the position to be tagged (can take any value from 1 to m)
- s is the previous state value (can take any value in S).
- s' is the new state value (can take any value in S)

Example of Features used in Part-of-Speech Tagging

1. $\vec{\phi}(x_1, \cdots, x_m, i, s_{i-1}, s_i)_1 = 1$ if s_i = ADVERB and word x_i ends in "-ly"; 0 otherwise.
   If the weight $\vec{w}_1$ associated with this feature is large and positive, then this feature is essentially saying that we prefer labelings where words ending in -ly get labeled as ADVERB.
2. $\vec{\phi}(x_1, \cdots, x_m, i, s_{i-1}, s_i)_2 = 1$ if $i = 1$, s_i= VERB, and x_m=?; 0 otherwise.
   If the weight $\vec{w}_2$ associated with this feature is large and positive, then labelings that assign VERB to the first word in a question (e.g., "Is this a sentence beginning with a verb?") are preferred.

[2] Source: `https://blog.echen.me/2012/01/03/introduction-to-conditional-random-fields/`

Questions?

Thanks for your Attention!

References I