Questions for Flipped Classroom Session of COMS 4705 Week 12, Fall 2014. (Michael Collins)

Question 1 Consider an application of global linear models to parsing. In this scenario each input x is a sentence. We have a fixed context-free grammar; GEN(x) returns the set of all parses allowed for x under the context-free grammar. The feature vector $f(x, y)$ for any sentence x paired with a parse tree y is defined as

$$f(x, y) = \sum_{\alpha \to \beta \in (x,y)} g(\alpha \to \beta)$$

where g is a function that maps a context-free rule $\alpha \to \beta$ to a feature vector, and the notation $\alpha \to \beta \in (x, y)$ refers to a sum over all context-free rules in the parse tree defined by (x, y).

We'd like $f(x, y)$ to be a 3-dimensional feature vector, with the following values for its three components:

$$\begin{aligned} f_1(x, y) &= \text{Number of times } \texttt{S -> NP VP} \text{ is seen in } (x, y) \\ f_2(x, y) &= \text{Number of times } \texttt{N -> dog} \text{ is seen in } (x, y) \\ f_3(x, y) &= \text{Number of times } \texttt{NP -> NP NP} \text{ is seen in } (x, y) \end{aligned}$$

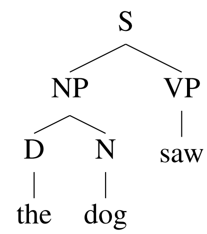
Give a definition of the function g that leads to this definition of $f(x, y)$.

Question 2 In this question we develop a global linear model for parsing with a context-free grammar in Chomsky normal form. The input to the model is a sentence $s = x_1 \ldots x_n$ where x_i is the i'th word in the sentence. We use $\mathcal{T}(s)$ to denote the set of all parse trees for the sentence s. For any parse tree $y \in \mathcal{T}(s)$, for any rule $X \rightarrow Y\ Z$ in the grammar, for any indices i, k, j such that $1 \leq i \leq k < j \leq n$, we define

$$\delta(y, X \rightarrow Y\ Z, i, k, j) = 1$$

if the rule $X \rightarrow Y\ Z$ is seen in the parse tree y, with non-terminal X spanning words $i \ldots j$ inclusive; non-terminal Y spanning words $i \ldots k$ inclusive; and non-terminal Z spanning words $k+1 \ldots j$ inclusive.

For example, for the parse tree



we have $\delta(\text{S} \rightarrow \text{NP VP}, 1, 2, 3) = \delta(\text{NP} \rightarrow \text{D N}, 1, 1, 2) = 1$, with all other δ values being equal to 0.

We also assume that we have a feature vector $g(s, X \rightarrow Y\ Z, i, k, j) \in \mathbb{R}^d$ for any sentence s together with a rule $X \rightarrow Y\ Z, i, k, j$; and a parameter vector $v \in \mathbb{R}^d$. The score for an entire parse tree under parameter values v is

$$\text{score}(y; v) = \sum_{X \rightarrow Y\ Z, i, k, j} \delta(y, X \rightarrow Y\ Z, i, k, j)\, (v \cdot g(s, X \rightarrow Y\ Z, i, k, j))$$

Thus the score for an entire parse tree is a sum of scores for the rules it contains, where each rule receives the score $v \cdot g(s, X \rightarrow Y\ Z, i, k, j)$.

Question 2a Give a dynamic programming algorithm that calculates

$$\max_{y \in \mathcal{T}(s)} \text{score}(y; v)$$

for any input sentence $s = x_1 \ldots x_n$. (For convenience, the CKY parsing algorithm for PCFGs is shown over the page, in figure 1.)

Question 2b Now assume that we have a training set consisting of pairs $s^{(i)}, y^{(i)}$ for $i \in \{1 \ldots M\}$, where each $s^{(i)}$ is a sentence, and each $y^{(i)}$ is a parse tree. We'd like to train the parameters of the model v using the perceptron algorithm for training global linear models. Give pseudo-code for the perceptron algorithm for training the parser below. You can assume that for any $s^{(i)}$, you can calculate

$$\arg \max_{y \in \mathcal{T}(s^{(i)})} \text{score}(y; v)$$

efficiently, where $\mathcal{T}(s^{(i)})$ is the set of all parse trees for the sentence $s^{(i)}$.

Input: a sentence $s = x_1 \ldots x_n$, a PCFG $G = (N, \Sigma, S, R, q)$.
Initialization:
For all $i \in \{1 \ldots n\}$, for all $X \in N$,

$$\pi(i, i, X) = \begin{cases} q(X \rightarrow x_i) & \text{if } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

- For $l = 1 \ldots (n-1)$
  - For $i = 1 \ldots (n-l)$
    - Set $j = i + l$
    - For all $X \in N$, calculate

$$\pi(i, j, X) = \max_{\substack{X \rightarrow Y Z \in R, \\ s \in \{i \ldots (j-1)\}}} \left(q(X \rightarrow Y Z) \times \pi(i, s, Y) \times \pi(s+1, j, Z) \right)$$

Output: Return $\pi(1, n, S) = \max_{t \in \mathcal{T}(s)} p(t)$

Figure 1: The CKY parsing algorithm.

Question 3 Consider a tagging problem where we have a training set with two training examples:

$$x^{(1)} = \texttt{a b c}, \qquad y^{(1)} = \texttt{A B C}$$
$$x^{(2)} = \texttt{a b e}, \qquad y^{(2)} = \texttt{A D E}$$

Now say we define the following features $f_j(h, y)$ for $j = 1 \ldots 9$, where h is a history and y is a tag:

$$\begin{aligned}
f_1(h, y) &= 1 \text{ if } x_i = \texttt{a} \text{ and } y = \texttt{A}, 0 \text{ otherwise} \\
f_2(h, y) &= 1 \text{ if } x_i = \texttt{b} \text{ and } y = \texttt{B}, 0 \text{ otherwise} \\
f_3(h, y) &= 1 \text{ if } x_i = \texttt{b} \text{ and } y = \texttt{D}, 0 \text{ otherwise} \\
f_4(h, y) &= 1 \text{ if } x_i = \texttt{c} \text{ and } y = \texttt{C}, 0 \text{ otherwise} \\
f_5(h, y) &= 1 \text{ if } x_i = \texttt{e} \text{ and } y = \texttt{E}, 0 \text{ otherwise} \\
f_6(h, y) &= 1 \text{ if } y_{-1} = \texttt{A} \text{ and } y = \texttt{B}, 0 \text{ otherwise} \\
f_7(h, y) &= 1 \text{ if } y_{-1} = \texttt{A} \text{ and } y = \texttt{D}, 0 \text{ otherwise} \\
f_8(h, y) &= 1 \text{ if } y_{-1} = \texttt{B} \text{ and } y = \texttt{C}, 0 \text{ otherwise} \\
f_9(h, y) &= 1 \text{ if } y_{-1} = \texttt{D} \text{ and } y = \texttt{E}, 0 \text{ otherwise}
\end{aligned}$$

Question 3a: Say we train a perceptron-based model with these features. Show that the algorithm will converge to a solution that recovers the correct tag sequence on both examples. (For this you just need to come up with parameter values for $v_1 \ldots v_9$ that recover the correct tag sequences on both examples.)

Question 3b: Now say we train a log-linear tagger (an MEMM). Show that the model cannot give $p(y^{(1)}|x^{(1)}) = 1$ and $p(y^{(2)}|x^{(2)}) = 1$.