Lecture 4, COMS E6998-3: Discriminative Context-Free Parsing

Michael Collins

February 9, 2011

Context-Free Grammars

- A context-free grammar (CFG) in Chomsky normal form is a tuple (V, Σ, R, S) where:
  - V is a finite set of *non-terminal* symbols
  - Σ is a finite set of *terminal* symbols
  - R is a set of rules: each rule either takes the form

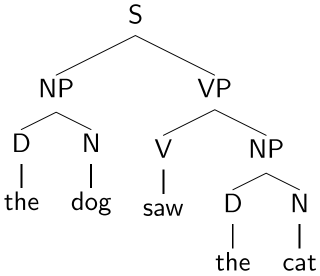
    $$X \rightarrow Y \; Z$$

    where $X, Y, Z \in V$, or

    $$X \rightarrow w$$

    where $X \in V$ and $w \in \Sigma$
  - $S \in V$ is the start symbol

Context-Free Parse Trees



- Each rule is a tuple $\langle X \rightarrow Y\ Z, i, k, j \rangle$ where $X \rightarrow Y\ Z$ is a rule, non-terminal X spans words $i \ldots j$ inclusive, Y spans words $i \ldots k$ inclusive, Z spans words $(k+1) \ldots j$ inclusive.
- Rules in this example:

$$S \rightarrow NP\ VP, 1, 2, 5$$
$$NP \rightarrow D\ N, 1, 1, 2$$
$$VP \rightarrow V\ NP, 3, 3, 5$$
$$NP \rightarrow D\ N, 4, 4, 5$$

Ambiguity

There are many sources of ambiguity: PP attachment, part-of-speech ambiguity, coordination, etc. etc.

Notation

- Assume $\underline{x}$ is a sequence of words $x_1 \ldots x_m$
- A context-free parse is a vector $\underline{y}$
- First, define the *index set* $\mathcal{I}$ to be the set of all possible rules:

  $$\mathcal{I} = \{X \rightarrow Y\ Z, i, k, j : X \rightarrow Y\ Z \in R, 1 \leq i \leq k < j \leq m\}$$

- Then $\underline{y}$ is a vector of values $y(r)$ for all $r \in \mathcal{I}$. $y(r) = 1$ if the structure contains the rule (r), $y(r) = 0$ otherwise.
- We use $\mathcal{Y}$ to refer to the set of all possible well-formed vectors $\underline{y}$

Feature Vectors for Rules

- $\underline{\phi}(\underline{x}, X \rightarrow Y\ Z, i, k, j)$ is a feature vector representing rule

  $$X \rightarrow Y\ Z, i, k, j$$

  for sentence $\underline{x}$
- Example features:
  - Identity of the rule $X \rightarrow Y\ Z$
  - Identity of the rule $X \rightarrow Y\ Z$ in conjunction with words at the boundary points i, k, or j
  - etc. etc.

CRFs for Discriminative Context-Free Parsing

- We use $\underline{\Phi}(\underline{x}, \underline{y}) \in \mathbb{R}^d$ to refer to a feature vector for an *entire* context-free parse tree $\underline{y}$
- We then build a log-linear model, very similar to a CRF

$$p(\underline{y}|\underline{x}; \underline{w}) = \frac{\exp\left(\underline{w} \cdot \underline{\Phi}(\underline{x}, \underline{y})\right)}{\sum_{\underline{y}' \in \mathcal{Y}} \exp\left(\underline{w} \cdot \underline{\Phi}(\underline{x}, \underline{y}')\right)}$$

- How do we define $\underline{\Phi}(\underline{x}, \underline{y})$? Answer:

$$\underline{\Phi}(\underline{x}, \underline{y}) = \sum_{r \in \mathcal{I}} y(r) \underline{\phi}(\underline{x}, r)$$

where $\underline{\phi}(\underline{x}, r)$ is the feature vector for rule r

Decoding

- The decoding problem: find

$$
\begin{aligned}
\arg\max_{\underline{y}\in\mathcal{Y}} p(\underline{y}|\underline{x};\underline{w}) &= \arg\max_{\underline{y}\in\mathcal{Y}} \frac{\exp\left(\underline{w}\cdot\underline{\Phi}(\underline{x},\underline{y})\right)}{\sum_{\underline{y}'\in\mathcal{Y}}\exp\left(\underline{w}\cdot\underline{\Phi}(\underline{x},\underline{y}')\right)} \\
&= \arg\max_{\underline{y}\in\mathcal{Y}} \exp\left(\underline{w}\cdot\underline{\Phi}(\underline{x},\underline{y})\right) \\
&= \arg\max_{\underline{y}\in\mathcal{Y}} \underline{w}\cdot\underline{\Phi}(\underline{x},\underline{y}) \\
&= \arg\max_{\underline{y}\in\mathcal{Y}} \underline{w}\cdot\sum_{r\in\mathcal{I}} y(r)\underline{\phi}(\underline{x},r) \\
&= \arg\max_{\underline{y}\in\mathcal{Y}} \sum_{r\in\mathcal{I}} y(r)\left(\underline{w}\cdot\underline{\phi}(\underline{x},r)\right)
\end{aligned}
$$

- This problem can be solved using dynamic programming, in $O(m^3)$ time, where m is the length of the sentence

Decoding using the CKY Algorithm

- For convenience, define

$$\theta(r) = \underline{w} \cdot \underline{\phi}(\underline{x}, r)$$

  The decoding problem is to find

$$\arg\max_{\underline{y} \in \mathcal{Y}} \sum_{r \in \mathcal{I}} y(r)\theta(r)$$

- Dynamic programming algorithm: define

$$\pi[X, i, j]$$

  for $X \in V$, $1 \leq i \leq j \leq m$ to be the highest score for any subtree rooted in non-terminal X, spanning words $i \ldots j$ inclusive

Decoding using the CKY Algorithm (continued)

- Initialization: for $i = 1 \ldots m$, $X \in V$, define $\pi[X, i, i] = 0$ if $X \rightarrow x_i$ is a valid rule, $-\infty$ otherwise. (Recall that x_i is the i'th word in the input sentence.)

- Recursive case: for $X \in V$, for $1 \leq i < j \leq n$,

$$\pi[X, i, j] = \max_{\substack{X \rightarrow Y \ Z \in R, \\ k \in \{i \ldots j-1\}}} \left(\theta(X \rightarrow Y \ Z, i, k, j) + \pi[Y, i, k] + \pi[Z, k+1, j]\right)$$

- The highest scoring tree has score $\pi[S, 1, m]$. Backpointers can be used to recover the identity of the highest scoring tree.

Parameter Estimation

- To estimate the parameters, we assume we have a set of n labeled examples, $\{(\underline{x}^i, \underline{y}^i)\}_{i=1}^n$. Each $\underline{x}^i$ is an input sequence $x_1^i \ldots x_m^i$, each $\underline{y}^i$ is a context-free tree
- We then proceed in exactly the same way as for CRFs
- The *regularized log-likelihood function* is

$$L(\underline{w}) = \sum_{i=1}^{n} \log p(\underline{y}^i | \underline{x}^i; \underline{w}) - \frac{\lambda}{2} ||\underline{w}||^2$$

- The *parameter estimates* are

$$\underline{w}^* = \arg\max_{\underline{w} \in \mathbb{R}^d} \sum_{i=1}^{n} \log p(\underline{y}^i | \underline{x}^i; \underline{w}) - \frac{\lambda}{2} ||\underline{w}||^2$$

The gradient of $L(\underline{w})$ can again be calculated efficiently, using dynamic programming algorithms