Lecture 4, COMS E6998-3: The Structured Perceptron

Michael Collins

February 9, 2011

Conditional Random Fields (CRFs)

- Notation: for convenience we'll use $\underline{x}$ to refer to the sequence of input words, $x_1 \ldots x_m$, and $\underline{s}$ to refer to a sequence of possible states, $s_1 \ldots s_m$. The set of possible states is $\mathcal{S}$. We use $\mathcal{Y}$ to refer to the set of *all possible state sequences* (we have $|\mathcal{Y}| = |S|^m$).
- We're again going to build a model of

$$p(s_1 \ldots s_m | x_1 \ldots x_m) = p(\underline{s} | \underline{x})$$

CRFs

- We use $\underline{\Phi}(\underline{x}, \underline{s}) \in \mathbb{R}^d$ to refer to a feature vector for an *entire* state sequence
- We then build a *giant* log-linear model,

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

- The model is "giant" in the sense that: 1) the space of possible values for $\underline{s}$, i.e., $\mathcal{Y}$, is huge. 2) The normalization constant (denominator in the above expression) involves a sum over a huge number of possibilities (i.e., all members of $\mathcal{Y}$).

CRFs (continued)

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

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

$$\underline{\Phi}(\underline{x}, \underline{s}) = \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s_{j-1}, s_j)$$

where $\underline{\phi}(\underline{x}, j, s_{j-1}, s_j)$ are the same as the feature vectors used in MEMMs.

Decoding with CRFs

- The decoding problem: find

$$
\begin{aligned}
\arg\max_{\underline{s}\in\mathcal{Y}} p(\underline{s}|\underline{x};\underline{w}) &= \arg\max_{\underline{s}\in\mathcal{Y}} \frac{\exp\left(\underline{w}\cdot\underline{\Phi}(\underline{x},\underline{s})\right)}{\sum_{\underline{s}'\in\mathcal{Y}}\exp\left(\underline{w}\cdot\underline{\Phi}(\underline{x},\underline{s}')\right)} \\
&= \arg\max_{\underline{s}\in\mathcal{Y}} \exp\left(\underline{w}\cdot\underline{\Phi}(\underline{x},\underline{s})\right) \\
&= \arg\max_{\underline{s}\in\mathcal{Y}} \underline{w}\cdot\underline{\Phi}(\underline{x},\underline{s}) \\
&= \arg\max_{\underline{s}\in\mathcal{Y}} \underline{w}\cdot\sum_{j=1}^{m}\underline{\phi}(\underline{x},j,s_{j-1},s_j) \\
&= \arg\max_{\underline{s}\in\mathcal{Y}} \sum_{j=1}^{m}\underline{w}\cdot\underline{\phi}(\underline{x},j,s_{j-1},s_j)
\end{aligned}
$$

- Again, we can use the Viterbi algorithm...

The Viterbi Algorithm for CRFs

- Initialization: for $s \in \mathcal{S}$

$$\pi[1, s] = \underline{w} \cdot \underline{\phi}(\underline{x}, 1, s_0, s)$$

where s_0 is a special "initial" state.

- For $j = 2 \dots m$, $s = 1 \dots k$:

$$\pi[j, s] = \max_{s' \in \mathcal{S}} \left[\pi[j-1, s'] + \underline{w} \cdot \underline{\phi}(\underline{x}, j, s', s) \right]$$

- We then have

$$\max_{s_1 \dots s_m} \sum_{j=1}^{m} \underline{w} \cdot \underline{\phi}(\underline{x}, j, s_{j-1}, s_j) = \max_s \pi[m, s]$$

- The algorithm runs in $O(mk^2)$ time. As before (see HMM lecture slides), we can use backpointers to recover the most likely sequence of states.

Parameter Estimation in CRFs

- To estimate the parameters, we assume we have a set of n labeled examples, $\{(\underline{x}^i, \underline{s}^i)\}_{i=1}^n$. Each $\underline{x}^i$ is an input sequence $x_1^i \ldots x_m^i$, each $\underline{s}^i$ is a state sequence $s_1^i \ldots s_m^i$.
- We then proceed in exactly the same way as for regular log-linear models
- The *regularized log-likelihood function* is

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

- Our parameter estimates are

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

- We find the optimal parameters using gradient-based methods

The Structured Perceptron

- Input: labeled examples, $\{(\underline{x}^i, \underline{s}^i)\}_{i=1}^{n}$.
- Initialization: $\underline{w} = \underline{0}$
- For $t = 1 \ldots T$, for $i = 1 \ldots n$:
  - Use the Viterbi algorithm to calculate

$$\underline{s}^* = \arg\max_{\underline{s} \in \mathcal{Y}} \; \underline{w} \cdot \underline{\Phi}(\underline{x}^i, \underline{s}) = \arg\max_{\underline{s} \in \mathcal{Y}} \; \sum_{j=1}^{m} \underline{w} \cdot \underline{\phi}(\underline{x}, j, s_{j-1}, s_j)$$

  - Updates:

$$\begin{aligned} \underline{w} &= \underline{w} + \underline{\Phi}(\underline{x}^i, \underline{s}^i) - \underline{\Phi}(\underline{x}^i, \underline{s}^*) \\ &= \underline{w} + \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s^i_{j-1}, s^i_j) - \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s^*_{j-1}, s^*_j) \end{aligned}$$

- Return $\underline{w}$

The Structured Perceptron with Averaging

- Input: labeled examples, $\{(\underline{x}^i, \underline{s}^i)\}_{i=1}^n$.
  Initialization: $\underline{w} = \underline{0}$, $\underline{w}_a = \underline{0}$
- For $t = 1 \dots T$, for $i = 1 \dots n$:
  - Use the Viterbi algorithm to calculate

$$\underline{s}^* = \arg\max_{\underline{s} \in \mathcal{Y}} \quad \underline{w} \cdot \underline{\Phi}(\underline{x}^i, \underline{s}) = \arg\max_{\underline{s} \in \mathcal{Y}} \quad \sum_{j=1}^{m} \underline{w} \cdot \underline{\phi}(\underline{x}, j, s_{j-1}, s_j)$$

  - Updates:

$$\begin{aligned} \underline{w} &= \underline{w} + \underline{\Phi}(\underline{x}^i, \underline{s}^i) - \underline{\Phi}(\underline{x}^i, \underline{s}^*) \\ &= \underline{w} + \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s_{j-1}^i, s_j^i) - \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s_{j-1}^*, s_j^*) \\ \underline{w}_a &= \underline{w}_a + \underline{w} \end{aligned}$$

- Return $\underline{w}_a / nT$

Convergence of the Structured Perceptron

- **Definition:** The training set $\{(\underline{x}^i, \underline{s}^i)\}_{i=1}^n$ is separable with margin $\delta > 0$, if there exists some parameter vector $\underline{w}$ such that:
  1. $||\underline{w}||^2 = 1$
  2. For all $i = 1 \ldots n$, for all $s_1 \ldots s_m$ such that $s_j \neq s_j^i$ for some j,
  $$\underline{w} \cdot \underline{\Phi}(\underline{x}^i, \underline{s}^i) - \underline{w} \cdot \underline{\Phi}(\underline{x}^i, \underline{s}) \geq \delta$$

- **Theorem:** if a training set is separable with margin δ, the structured perceptron makes at most
  $$\frac{R^2}{\delta^2}$$
  mistakes before convergence, where R is related to the norm of the feature vectors $\underline{\Phi}(\underline{x}^i, \underline{s})$