Conditional Random Field 的一些相关推导

夏庆荣　李正华

December 15, 2015

1　符号定义

$\mathcal{D} = \{S^j, Y^j\}_{j=1}^N$：表示一个数据集，包含 N 个句子和对应的 N 个人工标注的词性序列。

$S^j = w_1^j...w_i^j...x_{n_j}^j$：表示第 j 个句子，由 n_j 个词语组成。

$Y^j = y_1^j...y_i^j...y_{n_j}^j$：表示第 j 个句子对应的词性序列。

$\mathcal{T}$：表示词性集合，即隐状态的所有可能取值，$y_i^j \in \mathcal{T}$。

$\mathcal{V}$：表示词表 (vocabulary)，即数据 $\mathcal{D}$ 所有词语的集合，$w_i^j \in \mathcal{V}$。

2　CRF(Conditional Random Field) 模型定义

CRF，主要用于结构化分类问题（structured classifciation），即分类问题的类别是存在结构的，或者类别数目不确定（指数级）。

一个序列标注问题为：给定一个句子 $S = w_1...w_n$，要求确定其词性序列。如：“我 喜欢　我的　手机”，即给定句子，要求模型预测该句子的词性序列。

CRF 中，定义句子 S 标注为序列 Y 的概率为：

$$p(Y|S) = \frac{e^{\textit{Score}(S,Y)}}{Z(S)} \tag{1}$$

其中

$$Z(S) = \sum_{Y' \in \mathcal{T}^n} e^{\textit{Score}(S,Y')} \tag{2}$$

Score(S, Y) 的定义和 global linear model 里面的定义一样：

$$\begin{aligned} \textit{Score}(S,Y) &= \sum_{i=1}^{n} \textit{Score}(S,i,y_{i-1},y_i) \\ &= \sum_{i=1}^{n} \mathbf{w} \cdot \mathbf{f}(S,i,y_{i-1},y_i) \\ &= \mathbf{w} \cdot \mathbf{f}(S,Y) \end{aligned} \tag{3}$$

进而获得分值最高的词性序列：

$$Y^* = \arg\max_{Y \in \mathcal{T}^n} \textit{Score}(S,Y) \tag{4}$$

3 CRF 似然函数

$$
\begin{aligned}
LL(\mathcal{D};\mathbf{w}) &= \sum_{j=1}^{N} \text{log} p(Y^j|S^j;\mathbf{w}) \\
&= \sum_{j=1}^{N} \text{log} \frac{e^{\textit{Score}(S^j,Y^j)}}{Z(S^j)} \\
&= \sum_{j=1}^{N} \left[\text{log} e^{\textit{Score}(S^j,Y^j)} - \text{log} Z(S^j) \right] \\
&= \sum_{j=1}^{N} \left[\textit{Score}(S^j,Y^j) - \text{log} Z(S^j) \right]
\end{aligned}
\tag{5}
$$

4 CRF 似然函数求解

$$
\frac{\partial LL(\mathcal{D};\mathbf{w})}{\partial \mathbf{w}} = \sum_{j=1}^{N} \left[\mathbf{f}(S^j,Y^j) - \frac{\partial \text{log} Z(S^j)}{\partial \mathbf{w}} \right]
\tag{6}
$$

其中，公式 (6) 的后项：

$$
\begin{aligned}
\frac{\partial \text{log} Z(S^j)}{\partial \mathbf{w}} &= \frac{1}{Z(S^j)} \cdot Z'(S^j) \\
&= \frac{\sum_{Y' \in \mathcal{T}^n} e^{\textit{Score}(S,Y')} \cdot \mathbf{f}(S^j,Y')}{\sum_{Y' \in \mathcal{T}^n} e^{\textit{Score}(S,Y')}} \\
&= \sum_{Y' \in \mathcal{T}^n} p(Y'|S) \cdot \mathbf{f}(S^j,Y')
\end{aligned}
\tag{7}
$$

定义边缘概率：

$$
\begin{aligned}
p(y_{i-1}=t', y_i=t|S) = p(i,t',t|S) \\
&= \sum_{\substack{Y_{\neq\{i-1,i\}} \in \mathcal{T}^{n-2} \\ y_i=t \\ y_{i-1}=t'}} p(Y|S)
\end{aligned}
\tag{8}
$$

公式 (7) 可以表示为：

$$
\begin{aligned}
\mathsf{E}_{Y|S^j;\mathbf{w}} \cdot \mathbf{f}(S^j, Y) &= \sum_{Y\in\mathcal{T}^n} p(Y|S^j) \cdot \mathbf{f}(S^j, Y) \\
&= \sum_{Y\in\mathcal{T}^n} \left\{ p(Y|S^j) \cdot \left[\sum_{i=1}^{n} \mathbf{f}(S^j, i, y_{i-1}, y_i) \right] \right\} \\
&= \sum_{Y\in\mathcal{T}^n} \left[\sum_{i=1}^{n} p(Y|S^j) \cdot \mathbf{f}(S^j, i, y_{i-1}, y_i) \right] \\
&= \sum_{i=1}^{n} \sum_{y_{i-1}\in\mathcal{T}} \sum_{y_i\in\mathcal{T}} \sum_{Y_{\neq\{i-1,i\}}\in\mathcal{T}^{n-2}} p(Y|S^j) \cdot \mathbf{f}(S^j, i, y_{i-1}, y_i) \\
&= \sum_{i=1}^{n} \sum_{\substack{y_{i-1}\in\mathcal{T} \\ y_i\in\mathcal{T}}} \mathbf{f}(S, i, y_{i-1}, y_i) \left[\sum_{Y_{\neq\{i-1,i\}}\in\mathcal{T}^{n-2}} p(Y|S^j) \right] \\
&= \sum_{i=1}^{n} \sum_{\substack{t\in\mathcal{T} \\ t'\in\mathcal{T}}} \mathbf{f}(S^j, i, t', t) \cdot p(i, t', t|s)
\end{aligned} \tag{9}
$$

其中，$Y_{\neq\{i-1,i\}}$ 表示 Y 中不考虑 y_{i-1}, y_i 的子序列。

4.1 Forward

Forward 算法的目的是，给定句子 S，求解其所有可能的词性序列的得分之和。前向得分由左向右进行计算。我们令 $\alpha(k,t)$ 为第 k 个词 w_k 的词性为 t 的所有部分路径（后面的词不考虑）的得分之和，即：

$$
\begin{aligned}
\alpha(k,t) &= \sum_{\substack{y_{1\ldots k-1}\in\mathcal{T}^{k-1} \\ y_k=t}} e^{\mathsf{Score}(S, y_1\ldots y_k)} \\
&= \sum_{\substack{y_{1\ldots k-1}\in\mathcal{T}^{k-1} \\ y_k=t}} e^{\sum_{i=1}^{k} \mathsf{Score}(S,i,y_{i-1},y_i)} \\
&= \sum_{\substack{y_k=t \\ y_{k-1}\in\mathcal{T}}} \sum_{y_{1\ldots k-2}\in\mathcal{T}^{k-2}} \left(e^{\sum_{i=1}^{k-1} \mathsf{Score}(S,i,y_{i-1},y_i)} \right) \times \left(e^{\mathsf{Score}(S,k,y_{k-1},y_k)} \right) \\
&= \sum_{\substack{y_k=t \\ t'\in\mathcal{T}}} e^{\mathsf{Score}(S,k,t',t)} \cdot \sum_{\substack{y_{1\ldots k-2}\in\mathcal{T}^{k-2} \\ y_{k-1}=t'}} e^{\sum_{i=1}^{k-1} \mathsf{Score}(S,i,y_{i-1},y_i)} \\
&= \sum_{t'\in\mathcal{T}} e^{\mathsf{Score}(S,k,t',t)} \cdot \alpha(k-1, t')
\end{aligned} \tag{10}
$$

4.2 Backward

Backward 算法和 Forward 算法基本一致，区别在于 Backward 算法从右向左计算。我们令 $\beta(k,t)$ 为第 k 个词 w_k 的词性为 t（并以此为条件）的所有可能的后续词性路径

（前面的词不考虑）的得分之和，即：

$$
\begin{aligned}
\beta(k,t) &= \sum_{\substack{y_k=t \\ y_{k+1\dots n}\in\mathcal{T}^{n-k}}} e^{\textit{Score}(S,y_{k+1}\dots y_n|y_k)} \\
&= \sum_{\substack{y_k=t \\ y_{k+1\dots n}\in\mathcal{T}^{n-k}}} e^{\sum_{i=k+1}^{n}\textit{Score}(S,i,y_{i-1},y_i)} \\
&= \sum_{\substack{y_k=t \\ y_{k+1}\in\mathcal{T}}} \sum_{y_{k+2\dots n}\in\mathcal{T}^{n-k-1}} e^{\sum_{i=k+2}^{n}\textit{Score}(S,i,y_{i-1},y_i)} \cdot e^{\textit{Score}(S,k+1,y_k,y_{k+1})} \\
&= \sum_{\substack{y_k=t \\ t'\in\mathcal{T}}} e^{\textit{Score}(S,k+1,t,t')} \cdot \sum_{\substack{y_{k+2\dots n}\in\mathcal{T}^{n-k-1} \\ y_{k+1}=t'}} e^{\sum_{i=k+2}^{n}\textit{Score}(S,i,y_{i-1},y_i)} \\
&= \sum_{t'\in\mathcal{T}} e^{\textit{Score}(S,k+1,t,t')} \cdot \beta(k+1,t')
\end{aligned}
\tag{11}
$$

4.3 Forward-Backward 相互合作

利用 Forward-Backward 的结果，我们可以得到很多有用的信息：

公式 (9) 中后项可以表示为：

$$
\begin{aligned}
p(i,t',t|S) &= \frac{\sum_{\substack{Y_{\neq\{i-1,i\}}\in\mathcal{T}^{n-2} \\ y_{i-1}=t' \\ y_i=t}} e^{\textit{Score}(S,Y)}}{Z(S)} \\
&= \frac{\alpha(i-1,t')\cdot e^{\textit{Score}(S,i,t',t)}\cdot\beta(i,t)}{Z(S)}
\end{aligned}
\tag{12}
$$

$$
Z(S) = \alpha(n,\textit{STOP}) = \beta(0,\textit{START}) = \sum_{t\in\mathcal{T}} \alpha(k,t)\beta(k,t) \tag{13}
$$