Global Linear Model（结构化分类）讲课要点

夏庆荣，李正华

December 6, 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 将词性标注作为序列标注问题

和 LinearModel 不同，本节课中假设词语之间的词性是有关联的，当前词的词性和前一个词的词性相关（bigram 的情况，即表1中的第一个词性）。

Global linear model 又称为 structured perceptron，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。

线性模型中，定义句子 S 标为词性序列 Y 的分值（不是概率）为：

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

其中，$\mathbf{f}(S, i, y_{i-1}, y_i)$ 是一个特征抽取函数，根据表1中列出的特征模板，抽取出将第 i 个词标为 y_i，并且前一个词标为 y_{i-1} 时对应的特征向量。注意是稀疏特征向量，即只有很少的特征对应的值为 1，绝大多数都为 0。$\mathbf{w}$ 是特征权重向量，和 $\mathbf{f}(i, y_{i-1}, y_i)$ 的维度一致。每一个特征都有一个唯一个权重。

$\mathbf{f}(S, Y)$ 是一个聚合（accumulated）特征向量，将句子所有位置对应的特征向量累加起来得到，因此向量某些维度可能大于 1（同一个特征在不同位置重复出现）。换句话说，$\mathbf{f}(S, Y)$ 表示将句子 S 标为词性序列 Y 时对应的特征向量。

模型一次性考虑整个句子对应的词性序列（词性之间互相影响），而不是刻画一个词的词性，所以称为全局线性模型（Global Linear Model）。

进而获得分值最高的词性序列：

$$Y^* = \arg\max_{Y \in \mathcal{T}^n} \textit{Score}(S, Y) \tag{2}$$

01: $t_i \circ t_{i-1}$	02: $t_i \circ w_i$
03: $t_i \circ w_{i-1}$	04: $t_i \circ w_{i+1}$
05: $t_i \circ w_i \circ c_{i-1,-1}$	06: $t_i \circ w_i \circ c_{i+1,0}$
07: $t_i \circ c_{i,0}$	08: $t_i \circ c_{i,-1}$
09: $t_i \circ c_{i,k}, 0 < k < \#c_i - 1$	
10: $t_i \circ c_{i,0} \circ c_{i,k}, 0 < k < \#c_i - 1$	
11: $t_i \circ c_{i,-1} \circ c_{i,k}, 0 < k < \#c_i - 1$	
12: **if** $\#c_i = 1$ **then** $t_i \circ w_i \circ c_{i-1,-1} \circ c_{i+1,0}$	
13: **if** $c_{i,k} = c_{i,k+1}$ **then** $t_i \circ c_{i,k} \circ$ *"consecutive"*	
14: $t_i \circ \text{prefix}(w_i, k), 1 \leq k \leq 4, k \leq \#c_i$	
15: $t_i \circ \text{suffix}(w_i, k), 1 \leq k \leq 4, k \leq \#c_i$	

表 1: POS tagging feature templates $\mathbf{f}(S, i, t)$。$\circ$ means string concatenation;t_i denotes the corresponding tag of w_i; $c_{i,k}$ denotes the k^{th} Chinese character of w_i; $c_{i,0}$ is the first Chinese character; $c_{i,-1}$ is the last Chinese character; $\#c_i$ is the total number of Chinese characters contained in w_i; prefix/suffix(w_i, k) denote the k-Character prefix/suffix of w_i.

3 特征模板

在做分类的时候，我们希望把有用的信息都加入到模型中，这些信息即特征。特征模板的作用是描述我们想用哪些信息。表1给出了一个目前学术界比较常用的特征模板列表，即把 w_i 标为词性 t 时用到的特征集合。可以看到，我们主要用到了词信息，以及词内部的汉字信息。

4 线性模型的编码流程

4.1 训练阶段

给定数据集 $\mathcal{D}$，设计好特征模板列表，训练阶段的结果是输出模型：一个特征集合（字符串）和每个特征的权重，格式如：

02:NN○ 手机　2
03:NN○ 我的　5
...
07:NN○ 手　1
...

前面是具体的一个特征，后面是特征对应的权重（如果特征权重为 0，可以不输出）。

1) 确定 feature space，即收集训练数据中所有的特征。

具体做法是，顺序处理 $\mathcal{D}$ 中的所有训练实例（instance）。对于每一个实例，根据特征模板列表，得到具体的特征，添加到特征空间（集合）中。

2) 估计特征权重向量 **w**。

后面会介绍一种感知器在线学习方法。

3) 输出模型（特征集合和权重），写到磁盘上。

4.2 测试阶段

目标是，给一个新的句子，确定这个句子的词性序列。

利用公式 (2) 求解。即尝试所有的词性分别构成特征向量，计算对应分值，将所有分值累加，最后选择分值最大的词性序列。

5 全局线性模型的训练过程

本部分介绍一种学术界常用的简单有效的训练方法，称为感知器在线学习 Perceptron Online training。其思想是，每次取一个训练实例，用当前模型进行预测，然后根据预测结果对特征权重进行更新。见算法1。

5.1 Viterbi

Viterbi 算法的目的是，给定句子 S，求解其最优的词性序列 Y^*：

$$Y^* = \arg\max_{Y \in \mathcal{T}^n} \textit{Score}(S, Y) \tag{3}$$

我们令 $\pi(k,t)$ 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）的最大得分，即：

$$\begin{aligned}
\pi(k,t) &= \max_{\substack{y_k = t \\ y_1 \cdots y_{k-1} \in \mathcal{T}^{k-1}}} \textit{Score}(S, y_1 \ldots y_k) \\
&= \max_{\substack{y_k = t \\ y_1 \cdots y_{k-1} \in \mathcal{T}^{k-1}}} \sum_{i=1}^{k} \textit{Score}(S, i, y_{i-1}, y_i) \\
&= \max_{\substack{y_k = t \\ y_1 \cdots y_{k-1} \in \mathcal{T}^{k-1}}} \left\{ \sum_{i=1}^{k-1} \textit{Score}(S, i, y_{i-1}, y_i) + \textit{Score}(S, k, y_{k-1}, y_k) \right\} \\
&= \max_{\substack{y_k = t \\ t' \in \mathcal{T}}} \left\{ \textit{Score}(S, k, t', y_k) + \max_{\substack{y_{k-1} = t' \\ y_1 \cdots y_{k-2} \in \mathcal{T}^{k-2}}} \sum_{i=1}^{k-1} \textit{Score}(S, i, y_{i-1}, y_i) \right\} \\
&= \max_{\substack{y_k = t \\ t' \in \mathcal{T}}} \left\{ \textit{Score}(S, k, t', y_k) + \pi(k-1, t') \right\}
\end{aligned} \tag{4}$$

可以看到，这个求解过程符合动态规划算法的一些性质：如最优子结构、子问题结果重用。初始条件为：

$$\begin{aligned}
\pi(0, t = \textit{START}) &= 0 \\
\pi(0, t \neq \textit{START}) &= -\infty
\end{aligned} \tag{5}$$

如上公式中，只是求出最大概率，可以通过回溯（backward-trace）指针，获得对应最大概率的词性序列。

6 编程作业

编写一个 Global Linear Model，包括训练、测试、评价等程序。数据和 HMM 模型使用的数据（训练数据 train.conll、开发数据 dev.conll）相同。满分 10 分。

Algorithm 1 Perceptron Online training.

Input: Labeled data $\mathcal{D}$, and Feature space $\mathcal{E}$
Output: Feature weights **w**
Initialization: $\mathbf{v} = \mathbf{0}, \mathbf{w}^0 = \mathbf{0}$, $k = 0$;
for $m = 1$ **to** M **do** {*iterations*}
  for $j = 1$ **to** N **do** {*for sentence* S^j}
    $Y^* = \arg\max_{Y \in \mathcal{T}^n} \mathit{Score}(S^j, Y; \mathbf{w}^k)$
    if $Y^* \neq Y^j$ **then** {Y^j is the correct tag sequence}
      $\mathbf{w}^{k+1} = \mathbf{w}^k + \mathbf{f}(S^j, Y^j) - \mathbf{f}(S^j, Y^*)$
      $\mathbf{v} = \mathbf{v} + \mathbf{w}^{k+1}$
      $k = k + 1$
    end if
  end for
  Here, after each iteration, we can evaluate the current model **w** on some validation (development) dataset.
  我们可以使用 **v** 作为特征权重，对开发数据集进行处理，评价得到对应的准确率，并和使用 **w** 做比较。
end for
Output **w**