条件随机场

1. 概念

无向图模型，是指节点之间的连线没有方向，又称作马尔科夫随机场（MRF）或马尔科夫网络。（有向图模型称作贝叶斯网络）。

设$X=(X_1,X_2,...,X_n)$和$Y=(Y_1,Y_2,...,Y_n)$都是联合随机变量。若随机变量Y构成一个无向图$G=(V,E)$，表示马尔科夫随机场，则条件概率分布$P(Y|X)$称为条件随机场。其中X表示输入变量，为观测序列。Y表示输出变量，为状态序列、标记序列（和HMM中的概念保持一致）。

1.1 回顾逻辑回归

首先是Logistic函数

$$g(z)=\frac{1}{1+e^{-z}}$$

它的导数是

$$g^{,}(z)=\frac{d}{dz}\frac{1}{1+e^{-z}}=\frac{1}{(1+e^{-z})^2}(e^{-z})=\frac{1}{1+e^{-z}}(1-\frac{1}{1+e^{-z}})=g(z)(1-g(z))$$

假设$z=\theta^Tx$，定义$h_\theta(x)=g(\theta^Tx)=\frac{1}{1+e^{-\theta^Tx}}$，用$h_\theta(x)$表示以下概率

$$\begin{aligned}P(y=1|x;\theta)=h_\theta(x)\\P(y=0|x;\theta)=1-h_\theta(x)\end{aligned}$$

将其写在一起为

$$p(y|x;\theta)=(h_\theta(x))^y(1-h_\theta(x))^{1-y}$$

因此似然函数是

$$\begin{aligned}L(\theta)&=p(Y|X;\theta)\\&=\prod_{i=1}^m p\left(y^{(i)}|x^{(i)};\theta\right)\\&=\prod_{i=1}^m\left(h_\theta\left(x^{(i)}\right)\right)^{y^{(i)}}\left(1-h_\theta\left(x^{(i)}\right)\right)^{1-y^{(i)}}\end{aligned}$$

对似然函数取对数得到

$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))
\end{aligned}
$$

通过对对数似然函数进行极大似然估计。假设x是n维向量，因此变量θ也是n维，下面的式子是对θ的第j个维度求偏导数，将$h_\theta(x) = g(\theta^T x)$代入，

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} \ell(\theta) &= \left(y \frac{1}{g\left(\theta^T x\right)} - (1-y) \frac{1}{1-g\left(\theta^T x\right)}\right) \frac{\partial}{\partial \theta_j} g\left(\theta^T x\right) \quad \text{代入g的导数} \\
&= \left(y \frac{1}{g\left(\theta^T x\right)} - (1-y) \frac{1}{1-g\left(\theta^T x\right)}\right) g(\theta^T x)(1-g(\theta^T x)) \frac{\partial}{\partial \theta_j} \theta^T x \quad \text{只对}\theta\text{的第}j\text{维求导} \\
&= \left(y\left(1-g\left(\theta^T x\right)\right) - (1-y) g\left(\theta^T x\right)\right) x_j \\
&= \left(y - h_\theta(x)\right) x_j
\end{aligned}
$$

有了偏导数，就可以对变量进行优化

$$
\theta_j := \theta_j + \alpha\left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right) x_j^{(i)}
$$

1.2 对数线性模型

定义事件发生的几率odds，为事件发生的概率除以事件不发生的概率。

定义logit函数为几率的对数形式。

$$
\operatorname{logit}(p) = \log \frac{p}{1-p} = \log \frac{h_\theta(x)}{1-h_\theta(x)} = \log \left(\frac{\frac{1}{1+e^{-\theta^T x}}}{\frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}} \right) = \theta^T x
$$

发现几率取对数后，是一个线性模型，因此称为对数线性模型。

更进一步说明对数线性模型的一般形式。令x为样本，y为标记，在Logistic回归中，特征是样本的各维度$x = (x_1, x_2, \ldots, x_n)$，我们定义做$F_j(x, y)$，表示和$x$与$y$相关的第$j$个特征。因此将对数线性模型的一般形式定义为

$$
p(y|x; w) = \frac{1}{Z(x, w)} \exp\left(\sum_j w_j F_j(x, y)\right)
$$

其中，归一化因子为

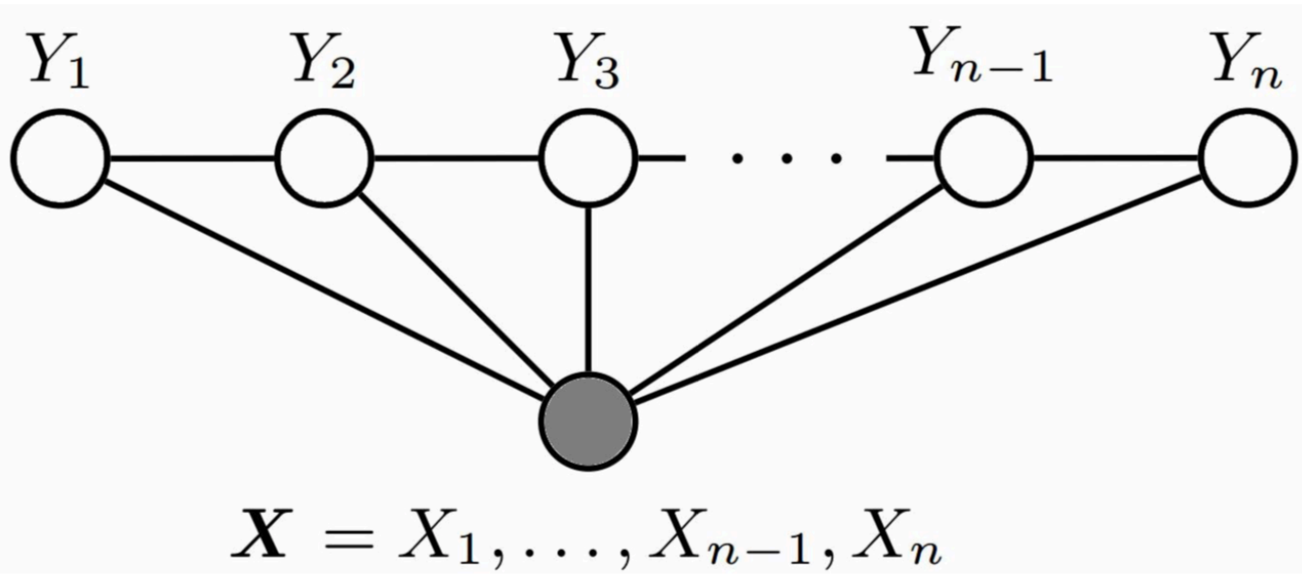
$$
Z(x, w) = \sum_y \exp \sum_j w_j F_j(x, y)
$$

因此，如果给定了x，并且训练好了w，预测x最可能的标记y

$$\hat{y} = \arg\max_y p(y|x;w) = \arg\max_y \sum_j w_j F_j(x,y)$$

1.3 特征函数的选择

特征函数几乎可以任意选择。在NLP中，可以从前缀、后缀、词典位置、前后单词、前置/后置标点的角度设计特征。特征的数量可以非常多。在本文中，假设每个特征只与当前**词性**与相邻**词性**有关。但是特征可以与所有的**词**有关。下图中可以看到，标记Y表示词性，X表示词，词性只与前后相邻的有关，而与所有的词X都有联系。相邻的标记互相影响，非独立，如果每个单词分别预测，将丢失信息。这就是线性链的条件随机场模型。



除了上述问题以外，该模型还将会解决不同的句子长度不同的问题，以及标记序列解集与句子长度呈现指数级增长，无法穷举的问题。

1.4 线性链条件随机场

线性链条件随机场可以使用对数线性模型表达。

定义$\bar{x}$表示n个词的序列，$\bar{y}$表示相应的词性序列，假设训练好了权重w，定义

$$p(\overline{y}|\overline{x};w) = \frac{1}{Z(\overline{x},w)} \exp\left(\sum_j w_j F_j(\overline{x},\overline{y})\right)$$

次特征

定义句子$\bar{x}$的第个j特征$F_j(\bar{x},\bar{y})$由若干个次特征$f_j(y_{i-1},y_i,\bar{x},i)$组合而成，这里的$f_j$依赖全部或部分词$\bar{x}$、当前和前一个词性$y_i$和$y_{i-1}$，以及当前词所在句子中的位置$i$

$$F_j(\bar{x},\bar{y}) = \sum_i f_j(y_{i-1},y_i,\bar{x},i)$$

可以发现，无论句子有多长，最终的特征F_j是由若干个次特征求和得到的，因此解决了训练样本变长的问题。

参数训练

给定样本$\bar{x}$和$\bar{y}$，学习问题，学w

$$\bar{y}^{\star}=\arg\max_{\bar{y}} P(\bar{y}|\bar{x},w)$$

使得$\bar{y}^{\star}$与给定的$\bar{y}$接近

概率计算

给定w，计算概率$P(y|x,w)$。

根据概率计算的公式$p(\overline{y}|\overline{x};w)=\frac{1}{Z(\overline{x},w)}\exp\left(\sum_j w_j F_j(\overline{x},\overline{y})\right)$

可以发现，归一化因子$Z(x,w)=\sum_y \exp\sum_j w_j F_j(x,y)$的外层循环是对$y$遍历，由于句子中的每个词的词性有$a$中可能，因此暴力求解的复杂度是$a^n$，是指数级别。

预测问题

给定w和$\bar{x}$，哪个$\bar{y}$最好。

同样是这个式子，在遍历$\bar{y}$的时候，暴力求解的复杂度是a^n，是指数级别。

$$\bar{y}^{\star}=\arg\max_{\bar{y}} P(\bar{y}|\bar{x},w)$$

要解决上述三个问题中，首先要克服遍历y带来的计算复杂度的难点，下面介绍解决方案。

1.5 状态关系矩阵

根据特征与次特征的关系

$$F_j(\overline{x},\overline{y})=\sum_i f_j\left(y_{i-1},y_i,\overline{x},i\right)$$

代入$\bar{y}^{\star}$的求解过程中，可得

$$\begin{aligned}
y^{*} &= \arg\max_{\bar{y}} \sum_j w_j \sum_i f_j\left(y_{i-1},y_i,\overline{x},i\right)=\arg\max_{\bar{y}} \sum_i \sum_i w_j f_j\left(y_{i-1},y_i,\overline{x},i\right) \\
&= \arg\max_{\bar{y}} \sum_i \sum_j w_j f_j\left(y_{i-1},y_i,\overline{x},i\right) \quad \text{交换求和的顺序} \\
&= \arg\max_{\bar{y}} \sum_i g_j\left(y_{i-1},y_i\right) \quad \text{定义} g_j\left(y_{i-1},y_i\right)=\sum_j w_j f_j\left(y_{i-1},y_i,\overline{x},i\right)
\end{aligned}$$

对于g_j而言，它是一个状态转移方阵，阶次是a。

定义前向概率$\alpha_k(v)$，表示第k个词的标记为v的最大得分值（这里不是概率，因为没有做归一化）

$$\alpha_k(v)=\max_{y_1,y_2,\cdots,y_{k-1}}\left(\sum_{i=1}^{k-1}g_i\left(y_{i-1},y_i\right)+g_k\left(y_{k-1},v\right)\right)$$

max里面的式子分两部，第一项是$\sum_{i=1}^{k-1}$，表示前$k-1$个词的词性，随便什么都可以。到了第k个词，其词性要从y_{k-1}转移到v。

因此，递推公式为

$$\alpha_k(v)=\max_{y_{k-1}}\left(\alpha_{k-1}\left(y_{k-1}\right)+g_k\left(y_{k-1},v\right)\right)$$

这样的算法复杂度为(a^2n)，a是标记数目，n是句子的词个数。这样，在每一个词k的时候，分别取$v=1,2,\ldots,a$，找到使得当前步骤$\alpha_k(v=?)$最大的v即可。

接下来，给定x和w，计算概率，归一化因子不好算，怎么办

$$\begin{aligned}&Z(\bar{x},w)=\sum_y\exp\sum_j w_jF_j(\bar{x},\bar{y})\\&=\sum_{\bar{y}}\exp\sum_i g_j\left(y_{i-1},y_i\right)\\&=\sum_y\prod_b\exp(g_j\left(y_{i-1},y_i\right))\end{aligned}$$

注意到g是一个$a\times a$的矩阵中的一个元素，因此定义矩阵M，其中$M_t(u,v)=\exp(g_t(u,v))$。

同时，定义起始状态$M_1(u,v)$中的$u=start$，终止状态$M_{n+1}(u,v)$中的$v=stop$。

那么

$$\begin{aligned}&M_{12}(\text{ start},v)\quad\text{前两个时刻，词性从start变化到v}\\&=\sum_q M_1(\text{start},q)M_2(q,v)\quad\text{1时刻从start随便变化到词性q，2时刻从词性q变化到v，然后把q积分掉}\\&=\sum_q\exp(g_1(start,q))\cdot\exp(g_2(q,v))\end{aligned}$$

同理，对于三个词的情况

$$\begin{aligned}&M_{123}(\text{ start},v)=\sum_q M_{12}(start,q)M_3(q,v)\\&=\sum_q\left(\sum M_1(start,r)M_2(r,q)\right)M_3(q,v)\\&=\sum_q M_1(start,r)M_2(r,q)M_3(q,v)\end{aligned}$$

因此，如果考虑n个词，

$$\begin{aligned}&M_{1,2,3\ldots n+1}(start,stop)=\sum M_1\left(start,y_1\right)M_2\left(y_1,y_2\right)\cdots M_{n+1}\left(y_n,stop\right)\\&=\sum_y\prod_i\exp(g_j\left(y_{i-1},y_i\right))\end{aligned}$$

即得到了归一化因子Z。这是n个矩阵连乘，时间复杂度为(a^3n)

1.6 参数训练

求对数目标函数的偏导数。

首先目标函数是

$$p(y|x;w) = \frac{1}{Z(x,w)}\exp\left(\sum_j w_j F_j(x,y)\right)$$

取了对数之后为

$$\Rightarrow \log p(y|x;w) = \log\frac{1}{Z(x,w)} + \log\exp\left(\sum_j w_j F_j(x,y)\right)$$
$$= -\log Z(x,w) + \sum_j w_j F_j(x,y)$$

接下来计算对数似然函数的梯度

$Z(x,w) = \sum_y \exp\sum_j w_j F_j(x,y)$

$$\begin{aligned}\frac{\partial}{\partial w_j}\log p(y|x;w) &= -\frac{\partial}{\partial w_j}\log Z(x,w) + F_j(x,y)\\ &= F_j(x,y) - \frac{1}{Z(x,w)}\sum_{\tilde{y}}\frac{\partial}{\partial w_j}\exp(\sum_l w_l F_l(x,\tilde{y}))\\ &= F_j(x,y) - \frac{1}{Z(x,w)}\sum_{\tilde{y}}[\exp\sum_l w_l F_l(x,\tilde{y})]F_j(x,\tilde{y})\\ &= F_j(x,y) - \sum_{\tilde{y}}F_j(x,\tilde{y})\frac{\exp\sum_l w_l F_l(x,\tilde{y})}{\sum\exp\sum_l w_l F_l(x,\tilde{y})}\\ &= F_j(x,y) - \sum_{\tilde{y}}F_j(x,\tilde{y})p(\tilde{y}|x;w)\\ &= F_j(x,y) - E_{\tilde{y}\sim p(\tilde{y}|x;w)}[F_j(x,\tilde{y})]\end{aligned}$$

所以优化参数w_j可以写成

$$w_j := w_j + \alpha\left(F_j(x,y) - E_{\tilde{y}\sim p(\tilde{y}|x;w)}[F_j(x,\tilde{y})]\right)$$