[翻译]Introduction to Conditional Random Fields

Translated by hittlle(QQ:519234757)

source: http://blog.echen.me/2012/01/03/introduction-to-conditional-random-fields/

假设你有 Justin Bieber（译注：加拿大著名歌手和作词人）一天的生活照片，你想根据每张照片的内容给照片打标签（诸如吃饭，睡觉，开车等标签）。应该怎么做呢？

一个办法就是忽略照片之间的顺序关系，给每一张照片构建一个分类器。譬如，根据标注好的一个月数量的照片，你可以得出早上 6 点左右拍的照片可能是表示睡觉，照片上亮色比较多的可能表示跳舞，照片里有车可能表示正在开车等等。

这样做，你会丢掉很多有用的信息。譬如，如果有张照片里有一张紧闭的嘴，那它表示什么呢？是表示吃饭还是唱歌呢？如果你知道这张照片的前一张照片是 Justin Bieber 在吃饭或者在做饭，那么这张照片很有可能就是在吃饭。但是，如果前一张照片是 Justin Bieber 在唱歌或者跳舞，那么这张照片很有可能就是在唱歌。

所以，为了增加标签的可靠性，我们应该把相邻的照片考虑进来，这正是条件随机场（Conditional Random Field, CRF）做的事。

词性标注

让我们用比较常用的词性标注为例进一步说明。

词性标注的目标就是给句子（一些列单词或记号）标上诸如形容词、名词、代词、动词、介词，副词、冠词等标签。

譬如，给这样一个句子

Bob drank coffee at Starbucks

可能的标注会是：Bob(名词) drank(动词) coffee(名词) at（介词）Starbucks(名词)

让我们构建一个条件随机场来对句子做词性标注。和其它分类器一样，我们首先要做的就是构建一系列特征函数 fi。

CRF 中的特征函数

在条件随机场中，每个特征函数接受如下的输入参数

* 句子
* 句子中单词的位置 i
* 当前单词的标签 l_i
* 前一个单词的标签 l_{i-1}

每个特征函数输出一个实值数(这个值经常要么是 0 要么是 1)

（注意：将特征函数限制成当前词的标注只和前一个词有关，而不是句子中所有的词的标签有关，我实际上是在构建一种特殊的条件随机场-线性链条条件随机场。为了简化起见，我会跳过一般条件随机场的相关内容）

特征函数的一个例子就是在前一个单词是 very 的情况下，当前单词应该被标注为形容词的可能性有多大。

从特征到概率

接下来，给每个特征函数 f_j 一个权重 λ_j (后面我会说明如何从数据中学习这些权重值)。给

定一个句子 s, 现在我们可以给句子的标签打分，即对所有特征函数对句子进行加权求和

$$score(l|s) = \sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s, i, l_i, l_{i-1})$$

(第一个求和符号遍历所有的特征函数 f_j，第二个求和遍历句子 s 所有的位置 i 的单词)

最后，我们通过指数化和归一化将这个分数转化成取值在 0 和 1 之间的概率 p(l|s)

$$p(l|s) = \frac{exp[score(l|s)]}{\sum_{l'} exp[score(l'|s)]} = \frac{exp[\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s,i,l_i,l_{i-1})]}{\sum_{l'} exp[\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s,i,l'_i,l'_{i-1})]}$$

特征函数例子

那么，这些特征函数是什么样的呢？词性标注的特征函数可能包括：

* 如果 l_i =ADVERB(副词)并且第 i 个单词以 ly 结尾，那么 f_1 (s, i, l_i, l_{i-1}) = 1，否则 f_1 (s, i, l_i, l_{i-1}) = 0；如果对应的权重 λ_1 为正且足够大，那就是说我们倾向于将以 ly 结尾的单词标识为 ADVERB(副词)

* 如果 i = 1, 并且 l_i =VERB(动词)，同时句子以问题结束，f_2 (s, i, l_i, l_{i-1}) = 1， 否则 f_2 = 0; 同样，如果对应的权重 λ_2 为正且足够大，那就是说我们倾向于把疑问句的第一个单词标注为动词（譬如 “Is this a sentence beginning with a verb?”）

* 如果 l_{i-1} =ADJECTIVE(形容词)并且 l_i =NOUN(名词), f_3 (s, i, l_i, l_{i-1}) = 1, 否则为 0；同样，如果对应的权重为正且足够大，那表示形容词后面倾向于跟名词

* 如果 l_{i-1} =PREPOSITION(介词)并且 l_i =PREPOSITION(介词)，f_4 (s, i, l_i, l_{i-1})=1, 否则为 0，如果对应的权重 λ_4 为负，那意味着介词后面一般不跟介词，我们应该避免在这种情况下进行标注。

这就是特征函数！总结一下：要构建条件随机场，你需要定义一堆特征函数（特征函数可以作用在整个句子上，当前位置上和相邻位置的标注上），并给特征函数分配权重，然后，按权重进行加权求和，最后将结果转化成概率。

现在我们来比较一下条件随机场和其它常用的机器学习算法。

有点类似于逻辑回归

CRF 求概率的公式有点类似于逻辑回归的公式

$$p(l|s) = \frac{exp[score(l|s)]}{\sum_{l'} exp[score(l'|s)]} = \frac{exp[\sum_{j=1}^{m}\sum_{i=1}^{n}\lambda_j f_j(s,i,l_i,l_{i-1})]}{\sum_{l'} exp[\sum_{j=1}^{m}\sum_{i=1}^{n}\lambda_j f_j(s,i,l'_i,l'_{i-1})]}$$

这是因为 CRF 其实是逻辑回归的序列版本；逻辑回归是分类的线性对数模型，而 CRF 是序列标签的线性对数模型

和 HMM 类似...

回想一下隐马尔可夫(HMM)也是另外一种词性标注模型（也是一种通用的序列标注模型）。当然，CRF 组合一堆特征函数来给标注打分，HMM 通过生成模型的方式来进行标注和定义

$$p(l,s) = p(l_1)\prod_i p(l_i|l_{i-1})p(w_i|l_i)$$

这儿

p(l_i | l_{i-1})表示转移概率(譬如介词后面跟名词的概率)

p(w_i | l_i)表示发射概率(譬如名词中出现 dad 的概率)

怎么比较 HMM 和 CRF 呢？CRF 更强大，它能解决 HMM 所能解决的问题，并且还能解决更多其它问题。可以从下面这个角度来看 HMM 和 CRF 的区别

注意到 HMM 计算概率的公式是

$$\log p(l,s) = \log p(l_0) + \sum_i \log p(l_i|l_{i-1}) + \sum_i \log p(w_i|l_i)$$

如果我们将这儿的对数概率看成是二值转移和发射概率指标函数，这和线链 CRF 的形式相同。所以，我们可以通过如下方式构建等价 CRF 模型

* 对 HMM 中每一个转移概率 p(l_i =y| l_{i-1} =x), 定义一系列 CRF 转移特征函数 $f_{x,y}$ (s, i, l_i, l_{i-1})=1，前提是 l_i = y 并且 l_{i-1} =x. 给每个特征函数一个权重 $w_{x,y}$ = logp(l_i =y| l_{i-1} =x)

*同样的，对每一个 HMM 的发射概率 p(w_i =z| l_i =x)， 定义一系列 CRF 发射特征函数 $g_{x,y}$ (s, i, l_i , l_{i-1})=1 ， 前 提 是 w_i =z 并 且 l_i =x, 否 则 为 0 ； 给 每 个 特 征 函 数 一 个 权 重 $w_{x,y}$ =logp(w_i =z| l_i =x)

通过这种方式，CRF 通过 p(l|s)计算出来的分数就和 HMM 计算出来的成比例，这样每一个 HMM 都和某个 CRF 等价。

但是，CRF 可以建模更加复杂的分布，这是因为下面两个原因

* CRF 可以定义更加多样的特征，而 HMM 却受限于此，这是因为 HMM 都是二值转移概率和发射概率特征函数，这就让当前单词只取决于当前标签而当前标签只取决于前一个标签，而 CRF 可以使用一些全局性的特征。例如，在我们上面的词性标注的例子中，有一个特征函数可以增加以动词开头以问题结尾的句子的概率权重。

*CRF 的权重可以是任意值，但 HMM 的概率必须满足限制条件；譬如：

$$0 <= p(w_i|l_i) <= 1, \sum_w p(w_i = w|l_1) = 1)$$

但是 CRF 的权重却没有这样的限制。

权重学习

现在我们回到如何学习 CRF 权重这个问题上来。一个毫不惊奇的办法就是使用梯度下降算法。

假设我们有一堆训练样本（句子和句子对应的词性标注）。先随机初始化我们的 CRF 模型的权重。为了把这些随机化的权重变成正确的权重，对每一个训练样本，我们需要

* 遍历每一个特征函数 fi, 计算训练样本的对数概率相对于 λ_i 的梯度，得

$$\frac{\partial}{\partial w_j} \log p(l|s) = \sum_{j=1}^{m} f_i(s, j, l_j, l_{j-1}) - \sum_{l'} p(l'|s) \sum_{j=1}^{m} f_i(s, j, l'_j, l'_{j-1})$$

* 注意到上面梯度的第一项是 fi 在真实标签下的的贡献度，第二项是 fi 在当前模型下的期望贡献度，这也是梯度上升所具备的形式

* 让 λ_i 沿梯度上升的方向移动：

$$\lambda_i = \lambda_i + \alpha[\sum_{j=1}^{m} f_i(s, j, l_j, l_{j-1}) - \sum_{l'} p(l'|s) \sum_{j=1}^{m} f_i(s, j, l'_j, l'_{j-1})]$$

这儿，α 是学习率

* 重复这个过程至到我们到达某些停止条件（例如，更新值小于某个阈值）

换句话说，上述每进行一步都会将现有模型向期望模型靠拢，而 λ_i 是靠拢期望模型的步伐。

找到最优的标注序列

假设现在我们已经训练好了 CRF 模型，现在有一个新的句子来了，我们应该怎么标注它呢？

一个直接的办法是对每种可能的标注都计算 $p(l|s)$，然后选择让 p 最大的标注序列。但是，对一个有 k 个标签和长度为 m 的句子来说，有 k^m 种可能性需要处理，这是指数级别的运算。一个更好的办法是注意到线性链 CRF 具备子结构优化的属性，这就允许我们采用动态规划的办法在多项时间内找到最优的标注序列，和 HMM 的维特比算法类似。