从点到线：逻辑回归到条件随机场

原创 夕小瑶 夕小瑶的卖萌屋 2017-07-23

开篇高能预警！本文前置知识：

1、理解**特征函数/能量函数、配分函数**的概念及其**无向图**表示，见《逻辑回归到受限玻尔兹曼机》和《解开玻尔兹曼机的封印》；

2、理解**特征函数形式的逻辑回归模型**，见《逻辑回归到最大熵模型》。

从逻辑回归出发，我们已经经过了朴素贝叶斯、浅层神经网络、最大熵等**分类模型**。显然，分类模型是不考虑时间的，仅仅计算当前的一堆特征对应的类别。因此，分类模型是“**点状**”的模型。

想一下，如果我们有一个**词性标注（POS）**的任务，在这个任务中，类别有动词、名词、形容词、副词、介词、连词等有限个类别。样本呢，当然就是自然语言序列啦，例如“夕小瑶喜欢 狗狗”这个序列就对应着“名词 动词 名词”这三个对应类别。

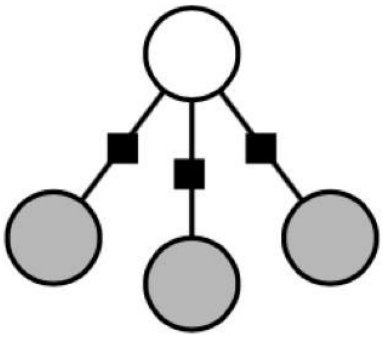
这时我们如果用“点状”模型，也就是分类模型来做这个任务，会产生什么现象呢？

假如我们选取的特征就是当前位置词，那么我们将分类器训练完成后，分类器遇到“夕小瑶”就会输出“名词”这个类别，也就是说它是不考虑上下文的，预测每个词的词性的时候才不会考虑整个句子的情况呢。在这里简单例子中看似没有什么影响，然而实际上非常多的词在不同的句子中会表现出不同的词性。比如“谷歌”一词，在“我今天参观了谷歌”中就是名词，在“你谷歌一下”中就是动词。可以看出，词性不仅取决于它自己，还取决于它的上下文（它两边的词）！

那么，有没有可能让逻辑回归、朴素贝叶斯这类点状模型利用好上下文信息呢？最容易想到的做法就是将上下文信息编码成特征啦！

比如加入当前词的2-gram上下文作为特征，这时在“你谷歌一下”中去预测“谷歌”的词性的时候，特征就是三维的：1、“谷歌”2、“你　　谷歌”3、“谷歌　　一下”。而在“我今天参观了谷歌”中，特征是1、“谷歌”2、“了　　谷歌”3、“谷歌　　<EOS>”这样就能根据不同的特征值在不同的句子中更精确的分类“谷歌”的词性啦～

在《逻辑回归到最大熵模型》中，小夕详细讲了如何将逻辑回归的传统形式转换成**特征函数/能量函数描述的形式**，而如《解开玻尔兹曼机的封印》所示，这种形式很容易画成有向图或无向图的形式：



Logistic Regression

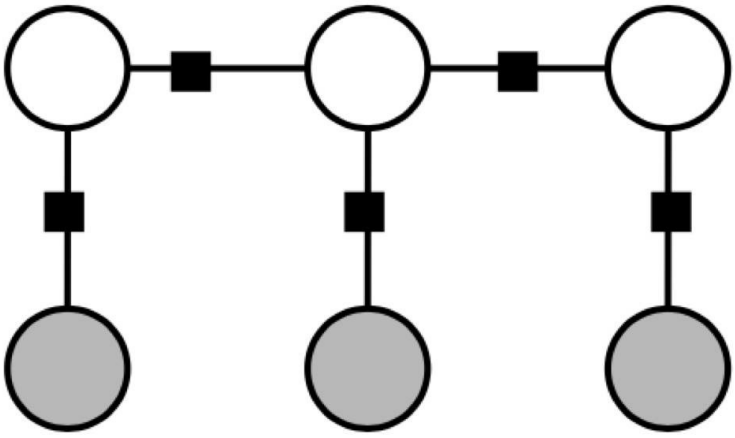
（上面白色圈圈是类别，下面灰色圈圈是各个特征，小黑框表示这一类别-特征对的能量函数/特征函数）

问题来了。对于一些更复杂的句子，可能决定某个词的词性的关键词距离该词有好长的距离，那怎么办呢？难道要扩展到10gram？

我们知道，ngram越长，训练数据就越稀疏，导致模型容易过拟合，泛化能力明显变差。显然点状的机器学习模型是很难在当前分类点利用到长距离信息的，也就是说，**其最多能通过加入短距离上下文特征来做到局部最优分类，而无法做到整个序列的最优分类**。

显然呐，自然语言文本的词性标注任务本来就是个“线状”的任务，你非要用“点状”的模型去做，肯定很差劲啦～那么我们能不能基于逻辑回归这个经典的判别式点状分类器来改良成“线状”，或者说“**链状**”模型呢？如果让你去改造，你会怎么改呢？

最简单的做法当然就是将序列前一时刻/位置的输出连到当前时刻到输出阿，也就是同时用**当前时刻的输入**和**前一时刻的输出**来决定当前时刻的输出(类别)，画出图来就是：



（当然啦，这里白色圈圈依然代表类别，灰色圈圈代表特征。为了画图简单，这里只画出了一个灰色圈圈（三个特征的时候应该在每个时刻画出三个灰色圈圈哦））

看，是不是超级简单的就改完了呢？这样在判断每一时刻的类别的时候就会不得不去参考前一时刻的类别，而参考前一时刻的类别的时候就隐含的包含了更早时刻的类别，这样就把整个句子串起来啦。

画起来容易，但是这个模型该如何用数学语言描述呢？

回顾一下可以直接画出上面逻辑回归的无向图的逻辑回归假设函数：

$$h = \frac{e^{\sum_i \lambda_i \cdot f_i(x,y)}}{Z}$$

扩展到多个类别的话，就是：

$$h(t,i) = P(y_t = i|x_t) = \frac{e^{\sum_o \lambda_{i,o} f(x_o(t),y_i(t))}}{Z}$$

从假设函数也可以看出，逻辑回归是个点状模型，当前时刻的类别预测不依赖任何其他时刻。

那么根据上面我们画的判别式链状模型图，我们唯一需要做的就是加入前一时刻y与当前时刻y的特征函数就可以啦～所以假设函数就很简单的变为了：

$$h = P(y|x) = \frac{e^{\sum_t \sum_{i,j} \lambda_{ij} f(y_i(t),y_j(t-1)) + \sum_t \sum_{o,i} \mu_{oi} f(y_i(t),x_o(t))}}{Z}$$

只是看起来有点长而已，而本质上还不是用特征函数描述了我们画的线状图嘛～

仔细观察，可以发现相比较点状模型，链状模型考虑了全部时间点，对全部时间点下的每个旧y与当前y，以及每个当前x与当前y做了求和，进而通过配分函数Z算出了**整个序列**的条件概率！注意对比逻辑回归的假设函数，逻辑回归的各个时间点是相互独立的，而这个链状模型则是统一考虑所有时间点，因此**是基于整个序列去做每个单词的词性预测**。

这个看似复杂，实则至简的链状模型就是“**线性链条件随机场（CRF）**”。实际上，线性链的条件随机场也是使用最广泛的条件随机场，几乎成了条件随机场的代名词。

这个模型的训练方法与隐马尔可夫模型是一样的，都是基于最大化似然函数的方法，方法已经在《HMM（下）》中讲解啦，在此不再赘述。当然啦，小夕只讲了最理想的情况，也就是训练集中既有X（观测序列），也有Y（隐状态序列）的情况。对于无法得到隐状态序列的情况，可以使用《EM算法》来迭代训练，在这里叫做BaumWelch算法，有兴趣的同学自行了解，这里不再展开啦。

诶？还有一个问题！虽然CRF的假设函数可以直接得到当前序列的每种可能的词性标注序列的概率，但是如果要枚举出所有可能的词性序列再找最大概率的那个词性序列的话，显然是指数爆炸的。对此有**维特比算法**进行优化，也已在《HMM（下）》中详细讲解过啦。在此不再赘述。值得一提，维特比算法的本质即**动态规划**。

看，是不是感觉一切都是一通百通了呢？这么看来条件随机场真的是没有新奇的东西，仅仅是用特征函数的老办法来将人人都能想到的前后两个时刻的y连起来，就结束了，结束了，束了，了。。