1、条件随机场CRF概述

将之前所有的观测作为未来预测的依据是**不现实的**，因为其复杂度会随着观测数量的增加而无限制地增长。因此，就有了马尔科夫模型，即假定**未来的预测仅与最近的观测有关，而独立于其他所有的观测。**

通过引入**隐变量**，解决Markov Model需要强独立性的问题，即隐马尔可夫模型 HMM。

隐马尔可夫模型HMM为生成式模型，计算联合概率分布$P(X,Z)$；

条件随机场CRF则是判别式模型，计算条件概率 $P(Y|X)$。由于 CRF 利用最大熵模型的思路建立条件概率模型，对于观测序列并没有做马尔科夫假设，可以得到全局最优，而HMM则是在马尔科夫假设下建立的联合分布，会出现局部最优的情况。（此处 Y,Z 均代表隐变量，X 为观测变量）

1.1、马尔可夫 Markov

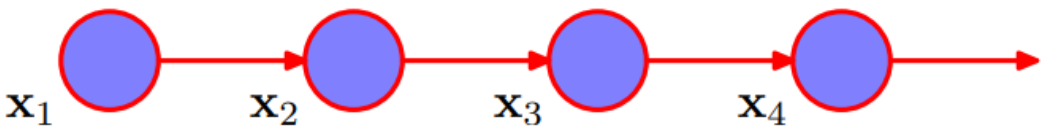
> 引例：假设我们观测一个二值变量，这个二值变量表示某一天是否下雨。给定这个变量的一系列观测，我们希望预测下一天是否会下雨。
>
> - 如果我们将所有的数据都看成独立同分布的， 那么我们能够从数据中得到的唯一的信息就是雨天的相对频率；
> - 然而，我们知道天气经常会呈现出持续若干天的趋势。因此，观测到今天是否下雨对于预测明天是否下雨会有极大的帮助。

我们可以使用概率的乘积规则来表示观测序列的联合概率分布，形式为：

$$P(x_1,x_2,\ldots,X_N)=p(x_1)\prod_{n=2}^{N}p(x_n|x1,x2,\ldots,x_{n-1})$$

利用马尔科夫性（一个马尔可夫过程可以表示为系统在状态转移过程中，第 n 次结果只受第 n-1 次结果的影响，即只与当前状态有关，而与过去状态，即与系统的初始状态和此次转移前的所有状态无关），可以将上式变为一阶马尔科夫链

$$P(x_1,x_2,\ldots,X_N)=p(x_1)\sum_{n=2}^{N}p(x_n|x_{n-1})$$



1.2、隐马尔可夫 HMM

> 隐变量：离散
>
> 观测变量：离散或连续
>
> **引例**：以句子和词性对应为例，观测变量 $X=\{x_1,x_2,\ldots,x_n\}$ 为显式的句子，隐变量 $Z=\{z_1,z_2,\ldots,z_n\}$ 为每一个单词对应的词性。

观测变量和隐变量上的联合概率分布为：

$$P(X,Z|\lambda)=P(z_1|\pi)[\prod_{n=2}^{N}P(z_n|z_{n-1},A)]\prod_{m=1}^{N}P(x_m|z_m,B)$$

$$P(X,Z|\lambda)=\alpha_t=\sum_{j=1}^{N}[\sum_{n=1}^{N}\alpha_{t-1}\times A_i]\times B_j(t-1)$$

定义HMM模型总用到的变量

给定模型 $\lambda=(A,B,\Pi)$ 和观测序列 $X=\{x_1,x_2,\ldots,x_n\}$，求观测序列 X 在模型 λ 下出现的条件概率 $P(X|\lambda)$：

- A = $[a_{ij}]$ 是隐藏状态转义概率矩阵
- B = $[b_j(k)]$ 就是观测状态生成概率矩阵，k表示时刻
- $\Pi=\pi_i$ 是隐藏状态的初始概率分布

举例说明：

我们的观察集是：

$$\mathrm{V}=\{红，白\}，\mathrm{M}=2$$

我们的状态集合是：

$$\mathrm{Q}=\{盒子一，盒子二，盒子三\}，\mathrm{N}=3$$

球的颜色观测序列

$$\mathrm{O}=\{红，白，红\}$$

观察序列和状态序列的长度都是3。

初始状态分布为：

$$\Pi=[0.2,0.4,0.4]^{\mathrm{T}}$$

状态转移概率分布矩阵为（三个盒子之间的状态转移、转变）：

$$\mathrm{A}=\begin{bmatrix}0.5 & 0.2 & 0.3\\ 0.3 & 0.5 & 0.2\\ 0.2 & 0.3 & 0.5\end{bmatrix}$$

观测状态矩阵

$$\mathrm{B}=\begin{bmatrix}0.5 & 0.5\\ 0.4 & 0.6\\ 0.7 & 0.3\end{bmatrix}$$

HMM可以解决的三个问题：

- 评估观察序列概率。
- 模型参数学习。
- 预测问题/解码问题。

1.3、马尔可夫随机场定义

> **什么叫随机场（MMF，Markov Random Field）？**
>
> 随机场是由若干个位置组成的整体，当给每一个位置中按照某种分布随机赋予一个值之后，其全体就叫做随机场。
>
> **例**：假如我们有一个十个词形成的句子需要做词性标注。这十个词每个词的词性可以在我们已知的词性集合（名词，动词...)中去选择。当我们为每个词选择完词性后，这就形成了一个随机场。

马尔可夫随机场是随机场的一个具有马尔可夫性得特例，它假设随机场中某一个位置的赋值仅仅与和它相邻的位置的赋值有关，和与其不相邻的位置的赋值无关。

例：如果我们假设所有词的词性只和它相邻的词的词性有关时，这个随机场就特化成一个马尔科夫随机场。比如第三个词的词性除了与自己本身的位置有关外，只与第二个词和第四个词的词性有关。

1.4、马尔可夫随机场拆解

马尔可夫随机场（Markov Random Field）包含两层意思。

马尔可夫性质：它指的是一个随机变量序列按时间先后关系依次排开的时候，第N+1时刻的分布特性，与N时刻以前的随机变量的取值无关。拿天气来打个比方。如果我们假定天气是马尔可夫的，其意思就是我们假设今天的天气仅仅与昨天的天气存在概率上的关联，而与前天及前天以前的天气没有关系。其它如传染病和谣言的传播规律，就是马尔可夫的。

随机场：当给每一个位置中按照某种分布随机赋予相空间的一个值之后，其全体就叫做随机场。我们不妨拿种地来打个比方。其中有两个概念：位置（site），相空间（phase space）。"位置"好比是一亩亩农田；"相空间"好比是种的各种庄稼。我们可以给不同的地种上不同的庄稼，这就好比给随机场的每个"位置"，赋予相空间里不同的值。所以，通俗的说，随机场就是在哪块地里种什么庄稼的事情。

马尔可夫随机场：马尔科夫随机场是具有马尔科夫特性的随机拿种地打比方，如果任何一块地里种的庄稼的种类仅仅与它邻近的地里种的庄稼的种类有关，与其它地方的庄稼的种类无关，那么这些地里种的庄稼的集合，就是一个马尔可夫随机场。

1.5、条件随机场 CRF

条件随机场 Conditional Random Field 是 **马尔可夫随机场 + 隐状态**的特例。

区别于**生成式**的隐马尔可夫模型HMM，CRF是**判别式**的。

假设我们有训练数据(X,Y)，X是属性集合，Y是类别标记。这时来了一个新的样本 x，我们想要预测它的类别 y。我们最终的目的是求得最大的条件概率 $P(y|x)$ 作为新样本的分类。

生成式模型：

一般会对每一个类建立一个模型，有多少个类别，就建立多少个模型。比如说类别标签有｛猫，狗，猪｝，那首先根据猫的特征学习出一个猫的模型，再根据狗的特征学习出狗的模型，之后分别计算新样本 x 跟三个类别的联合概率 $P(x,y)$，然后根据贝叶斯公式：

$$P(y|x) = \frac{P(x,y)}{P(x)}$$

分别计算$P(y|x)$，选择三类中最大的 $P(y|x)$ 作为样本的分类。

判别式模型：

根据训练数据得到分类函数和分界面，比如说根据 SVM 模型得到一个分界面，然后直接计算条件概率 $P(y|x)$，我们将最大的 $P(y|x)$ 作为新样本的分类。判别式模型是对条件概率建模，学习不同类别之间的最优边界，无法反映训练数据本身的特性，能力有限，其只能告诉我们分类的类别。

两模型区别：

不管是生成式模型还是判别式模型，它们最终的判断依据都是条件概率 $P(y|x)$，但是生成式模型先计算了联合概率 $P(y|x)$，再由贝叶斯公式计算得到条件概率。因此，生成式模型可以体现更多数据本身的分布信息，其普适性更广。

CRF 试图对多个随机变量（代表状态序列）在给定观测序列的值之后的条件概率进行建模：

给定观测序列，$X = \{x_1, x_2, \ldots, x_n\}$ 以及隐状态序列 $Y = \{y_1, y_2, \ldots, y_n\}$的情况下，构建条件概率模型 $P(Y|X)$。若随机变量 Y 构成的是一个马尔科夫随机场，则 $P(Y|X)$ 为条件随机场 CRF。

与**HMM**类似，**CRF**也关心如下三种问题：

- 评估观察序列概率。
- 模型参数学习。
- 预测问题/解码问题。

2、实体命名案例

2.1、三方库安装与导包

```
# pip install nltk
import nltk # Natural Language Toolkit 自然语言处理工具包

# pip install nltk
import sklearn_crfsuite # 条件随机场Python库
from sklearn_crfsuite import metrics
```

2.2、数据加载与配置

```
# 网络设置，跳过安全验证
import ssl
ssl._create_default_https_context = ssl._create_unverified_context
# 下载比较慢，不稳定
# 可以从百度网盘上下载，解压（解压到当前目录C:\Users\likai）到：C:\Users\likai\nltk_data
nltk.download('conll2002') # 下载数据，下载路径：C:\Users\likai\AppData\Roaming\nltk_data
nltk.corpus.conll2002.fileids() # 查看数据类别
```

加载训练和测试数据

```
%%time
train_sents = list(nltk.corpus.conll2002.iob_sents('esp.train'))
test_sents = list(nltk.corpus.conll2002.iob_sents('esp.testb'))
print('训练数据长度：',len(train_sents))
print('测试数据长度：',len(test_sents))
```

2.3、语料库介绍

- 通用语料库：CoNLL2002
- 语言：西班牙语
- 训练集：8323句
- 测试集：1517句
- 概念：一般来说，标注列表为['O' ,'B-MISC', 'I-MISC', 'B-ORG' ,'I-ORG', 'B-PER' ,'I-PER', 'B-LOC' ,'I-LOC']。其中，一般一共分为四大类：PER（人名），LOC（位置），ORG（组织）以及MISC（大杂烩），而且 B 表示开始，I 表示中间，O 表示单字词。
- 语料格式： 三列，分别表示词汇、词性、实体类型；使用Bakeoff-3评测中所采用的的BIO标注集即：
  - B-PER、I-PER 代表人名首字、人名非首字
  - B-LOC、I-LOC 代表地名首字、地名非首字
  - B-ORG、I-ORG 代表组织机构名首字、组织机构名非首字
  - O 代表该字不属于命名实体的一部分。
  - 如：('Australia', 'NP', 'B-LOC')
- 和其他语言一样，西班牙语也有很多词性，如：冠词、代词、动词、名词等等
  - 例如：NP表示专有名词，PP表示过去分词

```
display(train_sents[0],test_sents[-100])
'''
[('Melbourne', 'NP', 'B-LOC'),
 ('(', 'Fpa', 'O'),
 ('Australia', 'NP', 'B-LOC'),
 (')', 'Fpt', 'O'),
 (',', 'Fc', 'O'),
 ('25', 'Z', 'O'),
 ('may', 'NC', 'O'),
 ('(', 'Fpa', 'O'),
 ('EFE', 'NC', 'B-ORG'),
 (')', 'Fpt', 'O'),
 ('.', 'Fp', 'O')]
[('Valladolid', 'NC', 'B-LOC'),
 (',', 'Fc', 'O'),
 ('23', 'Z', 'O'),
 ('may', 'NC', 'O'),
 ('(', 'Fpa', 'O'),
 ('EFE', 'NC', 'B-ORG'),
 (')', 'Fpt', 'O'),
 ('.', 'Fp', 'O')]
'''
```

2.4、特征处理

主要选择处理了如下几个特征：

- 当前词的小写格式
- 当前词的后缀
- 当前词是否全大写 isupper

- 当前词的首字母大写，其他字母小写判断 istitle
- 当前词是否为数字 isdigit
- 当前词的词性
- 当前词的词性前缀

```
def word2features(sent, i):
    word = sent[i][0]
    postag = sent[i][1]
    # 当前词的特征信息
    features = {
        'word.lower()': word.lower(), # 小写形式
        'word[-3:]': word[-3:], # 词切片
        'word[-2:]': word[-2:],
        'word.isupper()': word.isupper(),# 判断是否大写
        'word.istitle()': word.istitle(), # 当前词的首字母大写，其他字母小写判断
        'word.isdigit()': word.isdigit(), # 判断是否是数字
        'postag': postag, # 词性
        'postag[:2]': postag[:2]} # 词性切片，去一部分

    # 前一个词的特征信息
    if i > 0:
        word1 = sent[i-1][0]
        postag1 = sent[i-1][1]
        features.update({
            '-1:word.lower()': word1.lower(),
            '-1:word.istitle()': word1.istitle(),
            '-1:word.isupper()': word1.isupper(),
            '-1:postag': postag1,
            '-1:postag[:2]': postag1[:2]})
    else:
        features['BOS'] = True # 表示开始

    # 后一个词的特征信息
    if i < len(sent)-1:
        word1 = sent[i+1][0]
        postag1 = sent[i+1][1]
        features.update({
            '+1:word.lower()': word1.lower(),
            '+1:word.istitle()': word1.istitle(),
            '+1:word.isupper()': word1.isupper(),
            '+1:postag': postag1,
            '+1:postag[:2]': postag1[:2]})
    else:
        features['EOS'] = True # 表示结束
    return features # 返回词特征信息

# 文本数据特征提取
def sent2features(sent):
    return [word2features(sent, i) for i in range(len(sent))]

# 文本数据对应词性
def sent2labels(sent):
    return [label for token, postag, label in sent]
```

2.4、文本数据特征提取

```
%%time
X_train = [sent2features(s) for s in train_sents]
y_train = [sent2labels(s) for s in train_sents]

X_test = [sent2features(s) for s in test_sents]
y_test = [sent2labels(s) for s in test_sents]

display(y_train[0],X_train[0])
display(y_test[0],X_test[0])
```

2.5、条件随机场CRF建模

```
%%time
crf = sklearn_crfsuite.CRF(
    c1=0.1,# L1正则化系数
    c2=0.1, # L2正则化系数
    max_iterations=100, # 梯度下降迭代次数
    all_possible_transitions=True)
# all_possible_transitions 参数说明：
# 指定条件随机场CRF是否生成训练数据中甚至没有出现的转移特征（即负转移特征）。
# 如果为True，CRF将生成关联所有可能标签对的转移特征
crf.fit(X_train, y_train)
```

可能出现问题：

2.6、测试数据预测

```
y_pred = crf.predict(X_test)
print('条件随机场实体命名准确率是：',crf.score(X_test,y_test))
display(y_pred[:2],y_test[:2])
'''
条件随机场实体命名准确率是： 0.971455184056818
[['B-LOC', 'I-LOC', 'O', 'O', 'O', 'O', 'B-ORG', 'O', 'O'], ['O']]
[['B-LOC', 'I-LOC', 'O', 'O', 'O', 'O', 'B-ORG', 'O', 'O'], ['O']]
'''
```