

第11章 条件随机场

1. 概率无向图模型是由无向图表示的联合概率分布。无向图上的结点之间的连接关系表示了联合分布的随机变量集合之间的条件独立性，即马尔可夫性。因此，概率无向图模型也称为马尔可夫随机场。

概率无向图模型或马尔可夫随机场的联合概率分布可以分解为无向图最大团上的正值函数的乘积的形式。

2. 条件随机场是给定输入随机变量 X 条件下，输出随机变量 Y 的条件概率分布模型，其形式为参数化的对数线性模型。条件随机场的最大特点是假设输出变量之间的联合概率分布构成概率无向图模型，即马尔可夫随机场。条件随机场是判别模型。

3. 线性链条件随机场是定义在观测序列与标记序列上的条件随机场。线性链条件随机场一般表示为给定观测序列条件下的标记序列的条件概率分布，由参数化的对数线性模型表示。模型包含特征及相应的权值，特征是定义在线性链的边与结点上的。线性链条件随机场的数学表达式是

$$P(y|x) = \frac{1}{Z(x)} \exp \left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i) \right)$$

其中，

$$Z(x) = \sum_y \exp \left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i) \right)$$

4. 线性链条件随机场的概率计算通常利用前向-后向算法。

5. 条件随机场的学习方法通常是极大似然估计方法或正则化的极大似然估计，即在给定训练数据下，通过极大化训练数据的对数似然函数以估计模型参数。具体的算法有改进的迭代尺度算法、梯度下降法、拟牛顿法等。

6. 线性链条件随机场的一个重要应用是标注。维特比算法是给定观测序列求条件概率最大的标记序列的方法。

例11.1

```
from numpy import *
```

```
#这里定义T为转移矩阵列代表前一个y(i,j)代表由状态i转到状态j的概率,Tx矩阵x对应于时间序列
#这里将书上的转移特征转换为如下以时间轴为区分的三个多维列表，维度为输出的维度
T1 = [[0.6, 1], [1, 0]]
T2 = [[0, 1], [1, 0.2]]
#将书上的状态特征同样转换成列表,第一个是为y1的未规划概率，第二个为y2的未规划概率
S0 = [1, 0.5]
S1 = [0.8, 0.5]
S2 = [0.8, 0.5]
Y = [1, 2, 2] #即书上例一需要计算的非规划条件概率的标记序列
```

```

Y = array(Y) - 1 #这里为了将数与索引相对应即从零开始
P = exp(S0[Y[0]])
for i in range(1, len(Y)):
    P *= exp((eval('S%d' % i)[Y[i]]) + eval('T%d' % i)[Y[i - 1]][Y[i]])
print(P)
print(exp(3.2))

```

```

24.532530197109345
24.532530197109352

```

例11.2

```

#这里根据例11.2的启发整合为一个矩阵
F0 = S0
F1 = T1 + array(S1 * len(T1)).reshape(shape(T1))
F2 = T2 + array(S2 * len(T2)).reshape(shape(T2))
Y = [1, 2, 2] #即书上例一需要计算的非规划条件概率的标记序列
Y = array(Y) - 1

P = exp(F0[Y[0]])
Sum = P
for i in range(1, len(Y)):
    PIter = exp((eval('F%d' % i)[Y[i - 1]][Y[i]]))
    P *= PIter
    Sum += PIter
print('非规范化概率', P)

```

```
非规范化概率 24.532530197109345
```

第11章条件随机场-习题

习题11.1

写出图11.3中无向图描述的概率图模型的因子分解式。

图11.3 无向图的团和最大团

解答：

图11.3表示由4个结点组成的无向图。图中由2个结点组成的团有5个：

$\{Y_1, Y_2\}, \{Y_2, Y_3\}, \{Y_3, Y_4\}, \{Y_4, Y_2\}$ 和 $\{Y_1, Y_3\}$ ，有2个最大团： $\{Y_1, Y_2, Y_3\}$ 和 $\{Y_2, Y_3, Y_4\}$ ，而 $\{Y_1, Y_2, Y_3, Y_4\}$ 不是一个团，因为 Y_1 和 Y_4 没有边连接。

根据概率图模型的因子分解定义：将概率无向图模型的联合概率分布表示为其最大团上的随机变量的函数的乘积形式的操作。公式在书中(11.5)，(11.6)。

$$P(Y) = \frac{\Psi_{(1,2,3)}(Y_{(1,2,3)}) \cdot \Psi_{(2,3,4)}(Y_{(2,3,4)})}{\sum_Y [\Psi_{(1,2,3)}(Y_{(1,2,3)}) \cdot \Psi_{(2,3,4)}(Y_{(2,3,4)})]}$$

习题11.2

证明 $Z(x) = a_n^T(x) \cdot \mathbf{1} = \mathbf{1}^T \cdot \beta_1(x)$, 其中 $\mathbf{1}$ 是元素均为1的 m 维列向量。

解答:

第1步: 证明 $Z(x) = a_n^T(x) \cdot \mathbf{1}$

根据条件随机场的矩阵形式: $(M_{n+1}(x))_{i,j} = \begin{cases} 1, & j = \text{stop} \\ 0, & \text{otherwise} \end{cases}$ 根据前向向量的定义:

$$\alpha_0(y|x) = \begin{cases} 1, & y = \text{start} \\ 0, & \text{otherwise} \end{cases}$$

$$\begin{aligned} \therefore Z_n(x) &= (M_1(x)M_2(x) \cdots M_{n+1}(x))_{(\text{start}, \text{stop})} \\ &= \alpha_0(x)^T M_1(x)M_2(x) \cdots M_n(x) \cdot \mathbf{1} \\ &= \alpha_n(x)^T \cdot \mathbf{1} \end{aligned}$$

第2步: 证明 $Z(x) = \mathbf{1}^T \cdot \beta_1(x)$

根据条件随机场的矩阵形式: $(M_{n+1}(x))_{i,j} = \begin{cases} 1, & j = \text{stop} \\ 0, & \text{otherwise} \end{cases}$ 根据后向向量定义:

$$\begin{aligned} \beta_{n+1}(y_{n+1}|x) &= \begin{cases} 1, & y_{n+1} = \text{stop} \\ 0, & \text{otherwise} \end{cases} \\ \therefore Z_n(x) &= (M_1(x)M_2(x) \cdots M_{n+1}(x))_{(\text{start}, \text{stop})} \\ &= (M_1(x)M_2(x) \cdots M_n(x)\beta_{n+1}(x))_{\text{start}} \\ &= (\beta_1(x))_{\text{start}} \\ &= \mathbf{1}^T \cdot \beta_1(x) \end{aligned}$$

综上所述: $Z(x) = a_n^T(x) \cdot \mathbf{1} = \mathbf{1}^T \cdot \beta_1(x)$, 命题得证。

习题11.3

写出条件随机场模型学习的梯度下降法。

解答:

条件随机场的对数极大似然函数为: $L(w) = \sum_{j=1}^N \sum_{k=1}^K w_k f_k(y_j, x_j) - \sum_{j=1}^N \log Z_w(x_j)$ 梯度下降算法的目标函数是 $f(w) = -L(w)$

目标函数的梯度为: $g(w) = \nabla f(w^{(k)}) = \left(\frac{\partial f(w)}{\partial w_1}, \frac{\partial f(w)}{\partial w_2}, \dots, \frac{\partial f(w)}{\partial w_k} \right)$ 其中

$$\begin{aligned} \frac{\partial f(w)}{\partial w_i} &= - \sum_{j=1}^N w_i f_i(y_j, x_j) + \sum_{j=1}^N \frac{1}{Z_w(x_j)} \cdot \frac{\partial Z_w(x_j)}{\partial w_i} \\ &= - \sum_{j=1}^N w_i f_i(y_j, x_j) + \sum_{j=1}^N \frac{1}{Z_w(x_j)} \sum_y (\exp \sum_{k=1}^K w_k f_k(y, x_j)) w_i f_i(y, x_j) \end{aligned}$$

根据梯度下降算法:

1. 取初始值 $w^{(0)} \in \mathbf{R}^n$, 置 $k = 0$
2. 计算 $f(w^{(k)})$
3. 计算梯度 $g_k = g(w^{(k)})$, 当 $\|g_k\| < \epsilon$ 时, 停止迭代, 令 $w^* = w^{(k)}$; 否则令 $p_k = -g(w^{(k)})$, 求 λ_k , 使 $f(w^{(k)}) + \lambda_k p_k = \min_{\lambda \geq 0} f(w^{(k)}) + \lambda p_k$
4. 置 $w^{(k+1)} = w^{(k)} + \lambda_k p_k$, 计算 $f(w^{(k+1)})$
当 $\|f(w^{(k+1)}) - f(w^{(k)})\| < \epsilon$ 或 $\|w^{(k+1)} - w^{(k)}\| < \epsilon$ 时, 停止迭代, 令 $w^* = w^{(k+1)}$
5. 否则, 置 $k = k + 1$, 转(3).

习题11.4

参考图11.6的状态路径图, 假设随机矩阵 $M_1(x)$, $M_2(x)$, $M_3(x)$, $M_4(x)$ 分别是

$M_1(x) = \begin{bmatrix} 0 & 0 \\ 0.5 & 0.5 \end{bmatrix}$, $M_2(x) = \begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{bmatrix}$, $M_3(x) = \begin{bmatrix} 0.5 & 0.5 \\ 0.6 & 0.4 \end{bmatrix}$, $M_4(x) = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ 求以 $start = 2$ 为起点 $stop = 2$ 为终点的所有路径的状态序列 y 的概率及概率最大的状态序列。

解答:

```
import numpy as np
```

```
# 创建随机矩阵
```

```
M1 = [[0, 0], [0.5, 0.5]]
```

```
M2 = [[0.3, 0.7], [0.7, 0.3]]
```

```
M3 = [[0.5, 0.5], [0.6, 0.4]]
```

```
M4 = [[0, 1], [0, 1]]
```

```
M = [M1, M2, M3, M4]
```

```
print(M)
```

```
[[[0, 0], [0.5, 0.5]], [[0.3, 0.7], [0.7, 0.3]], [[0.5, 0.5], [0.6, 0.4]],  
 [[0, 1], [0, 1]]]
```

```
# 生成路径
```

```
path = [2]
```

```
for i in range(1, 4):
```

```
    paths = []
```

```
    for _, r in enumerate(path):
```

```
        temp = np.transpose(r)
```

```
        paths.append(np.append(temp, 1))
```

```
        paths.append(np.append(temp, 2))
```

```
    path = paths.copy()
```

```
path = [np.append(r, 2) for _, r in enumerate(path)]
```

```
print(path)
```

```
[array([2, 1, 1, 1, 2]), array([2, 1, 1, 2, 2]), array([2, 1, 2, 1, 2]),  
 array([2, 1, 2, 2, 2]), array([2, 2, 1, 1, 2]), array([2, 2, 1, 2, 2]),  
 array([2, 2, 2, 1, 2]), array([2, 2, 2, 2, 2])]
```

计算概率

```
pr = []
for _, row in enumerate(path):
    p = 1
    for i in range(len(row) - 1):
        a = row[i]
        b = row[i + 1]
        p *= M[i][a - 1][b - 1]
    pr.append((row.tolist(), p))
pr = sorted(pr, key=lambda x: x[1], reverse=True)
print(pr)
```

```
[([2, 1, 2, 1, 2], 0.21), ([2, 2, 1, 1, 2], 0.175), ([2, 2, 1, 2, 2], 0.175),
([2, 1, 2, 2, 2], 0.13999999999999999), ([2, 2, 2, 1, 2], 0.09), ([2, 1, 1, 1,
2], 0.075), ([2, 1, 1, 2, 2], 0.075), ([2, 2, 2, 2, 2], 0.06)]
```

打印结果

```
print("以start=2为起点stop=2为终点的所有路径的状态序列y的概率为：")
for path, p in pr:
    print("    路径为：" + "->".join([str(x) for x in path]), end=" ")
    print("概率为：" + str(p))
print("概率[" + str(pr[0][1]) + "]最大的状态序列为：",
      "->".join([str(x) for x in pr[0][0]]))
```

以start=2为起点stop=2为终点的所有路径的状态序列y的概率为：

```
    路径为：2->1->2->1->2  概率为：0.21
    路径为：2->2->1->1->2  概率为：0.175
    路径为：2->2->1->2->2  概率为：0.175
    路径为：2->1->2->2->2  概率为：0.13999999999999999
    路径为：2->2->2->1->2  概率为：0.09
    路径为：2->1->1->1->2  概率为：0.075
    路径为：2->1->1->2->2  概率为：0.075
    路径为：2->2->2->2->2  概率为：0.06
概率[0.21]最大的状态序列为：2->1->2->1->2
```

参考代码：<https://blog.csdn.net/GrinAndBearIt/article/details/79229803>

本文代码更新地址：<https://github.com/fengdu78/lihang-code>

习题解答：<https://github.com/datawhalechina/statistical-learning-method-solutions-manual>

中文注释制作：机器学习初学者公众号：ID:ai-start-com

配置环境：python 3.5+

代码全部测试通过。