第四章　足球比赛的预测问题

4.1　教学目标

让学生能够应用基于局部搜索的技术来实现一个问题求解智能体。这一类问题关注目标本身，而不是达到目标的路径。学生能够通过本次实验学会应用局部搜索技术解决优化问题。

4.2　实验任务

三支足球队A, B, C两两之间各赛一场，总共需要赛三场，分别是A对B, A对C, B对C。对一支队来说，结果可能是胜、平、负之一。每只球队实力为一个0-3之间的整数，但目前未知。假设每场比赛的结果以某种概率取决于实力差距。如果除最后一场比赛外，前两场比赛结果已知A战胜了B且和C战平，请预测最后一场比赛B对C的结果。

4.3　方法与原理

记三队的实力为X_A, X_B, X_C，其先验分别满足分布$P_A(X), P_B(X), P_C(X)$，其中X取值0，1，2，3。一场比赛结果与队伍实力的关系的表现为条件分布，如$P(S_{AB}|X_A, X_B)$，其中S_{AB} 是A队对战B队时A队的结果，假设胜、负、平分别用0，1，2表示。根据实力和比赛结果的关系，构建贝叶斯网络图4.1。

问题为求$P(S_{BC}|S_{AB}=0, S_{AC}=1)$。

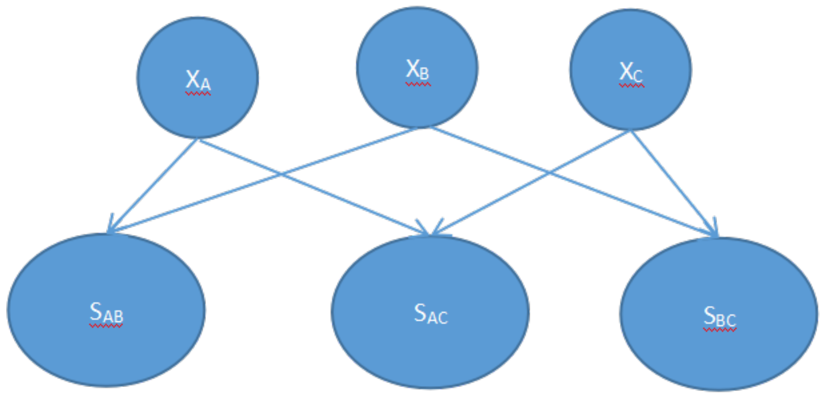

图 4.1: 比赛问题的贝叶斯网络

4.3.1 精确算法

贝叶斯网络的联合分布概率计算公式为

$$P(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} P(x_i|Parent(x_i)) \tag{4.1}$$

应用条件概率、边缘概率、以及联合分布率计算公式,

$$\begin{aligned}
& P(S_{BC}|S_{AB}=0, S_{AC}=1) \\
= \; & \alpha P(S_{BC}, S_{AB}=0, S_{AC}=1) && (4.2)\\
= \; & \alpha \sum_{X_A=0}^{3} \sum_{X_B=0}^{3} \sum_{X_C=0}^{3} P(S_{BC}, S_{AB}=0, S_{AC}=1, X_A, X_B, X_C) && (4.3)\\
= \; & \alpha \sum_{X_A=0}^{3} \sum_{X_B=0}^{3} \sum_{X_C=0}^{3} P(S_{BC}|X_B, X_C) P(S_{AB}=0|X_A, X_B) P(S_{AC}=1|X_A, X_C) \\
& P_A(X_A) P_B(X_B) P_B(X_C) && (4.4)
\end{aligned}$$

假定$P_A(X), P_B(X), P_C(X)$记录在1维数组PA,PB,PC中，条件概率如$P(S_{AB}|X_A, X_B)$记录在三维数组$PS[X_A][X_B][S_{AB}]$中，则精确计算方法为：

```
def direct_cal():
    res=[0,0,0]
    for XA in range (4):
```

```
        for XB in range (4):
            for XC in range (4):
                for sBC in range (3):
                    res[sBC] += PA[XA]*PB[XB]*PC[XC]*PS[XA][XB][0]*PS[XA][XC][1]*PS[XB][XC][sBC]
    return normal(res)
```

4.3.2 拒绝采样方法

4.3.1节给出的精确算法的计算式包括多层累加，当变量较多时，计算复杂性是指数级的。蒙特卡洛算法通过采样的方式，给出近似解，能降低算法的复杂性。拒绝采样方法按贝叶斯网络结点的顺序对所有变量进行采样，获得一个事件。经过N次采样后，对采样事件集进行统计，获得查询结果。

```
def reject_sampling():
    n=5000
    nBC=[0,0,0]
    for i in range(n):
        XA=np.random.choice(4,p=PA)
        XB=np.random.choice(4,p=PB)
        XC=np.random.choice(4,p=PC)
        SAB=np.random.choice(3,p=PS[XA][XB])
        SAC=np.random.choice(3,p=PS[XA][XC])
        SBC=np.random.choice(3,p=PS[XB][XC])
        if SAB==0 and SAC==1:
            nBC[SBC]+=1
    return normal(nBC)
```

4.3.3 似然加权采样方法

拒绝采样方法最后统计的是出现证据的样本点，其他的被拒绝，因此造成计算浪费。似然加权方法固定证据变量，只对非证据变量进行采样。然而每个事件与证据有不同的吻合程度，在计数时需考虑，因此对证据变量计算权值，最后算到结果中。

```
def likehood_weighting():
    n=5000
```

```
    nBC=[0,0,0]
    for i in range(n):
        w=1
        XA=np.random.choice(4,p=PA)
        XB=np.random.choice(4,p=PB)
        XC=np.random.choice(4,p=PC)
        #SAB
        w=w*PS[XA][XB][0]
        #SAC
        w=w*PS[XA][XC][1]
        SBC=np.random.choice(3,p=PS[XB][XC])
        nBC[SBC]+=w
    return normal(nBC)
```

4.3.4 Gibbs采样方法

Gibbs采样从一个初始样本出发，每次更改一个非证据变量形成一系列的采样点，然后对查询变量进行统计。采样一个非证据变量时，以其马尔可夫覆盖为条件。

```
def Gibs():
    n=4999
    nBC=[0,0,0]

    XA,XB,XC,SAB,SAC,SBC=0,0,1,0,1,1

    for k in range(n):

        _PA=normal([PA[i]*PS[i][B][SAB]*PS[i][C][SAC]\
                    for i in range(4)])
        XA=np.random.choice(4,p=_PA)

        _PB=normal([PB[i]*PS[XA][i][SAB]*PS[i][XC][SBC]\
                     for i in range(4)])
        XB=np.random.choice(4,p=_PB)

        _PC=normal([PC[i]*PS[XA][i][SAC]*PS[XB][i][SBC]\
                     for i in range(4)])
        XC=np.random.choice(4,p=_PC)
```

```

        SBC=np.random.choice(3,p=PS[XB][XC])
        nBC[SBC]+= 1
    return normal(nBC)
```

4.4　实验步骤

1. 设置贝叶斯网络的条件概率表

```
PA=[0.3,0.3,0.2,0.2]
PB=[0.4,0.4,0.1,0.1]
PC=[0.2,0.2,0.3,0.3]
PS=[[[0.2,0.6,0.2],[0.1,0.3,0.6],[0.05,0.2,0.75],[0.01,0.1,0.89]],
    [[0.6,0.3,0.1],[0.2,0.6,0.2],[0.1,0.3,0.6],[0.05,0.2,0.75]],
    [[0.75,0.2,0.05],[0.6,0.3,0.1],[0.2,0.6,0.2],[0.1,0.3,0.6]],
    [[0.89,0.1,0.01],[0.75,0.2,0.05],[0.6,0.3,0.1],[0.2,0.6,0.2]]]
```

2. 分别实现精确求解程序，拒绝采样方法，似然加权采样方法，Gibbs采样方法，获得BC比赛结果的后验分布；

3. 调整采样次数，观测几个近似方法的相对于精确解的差距。

4.5　小结

按实验步骤完成实验后，提供数据和代码，并完成对实验结果的分析和总结，撰写实验报告。