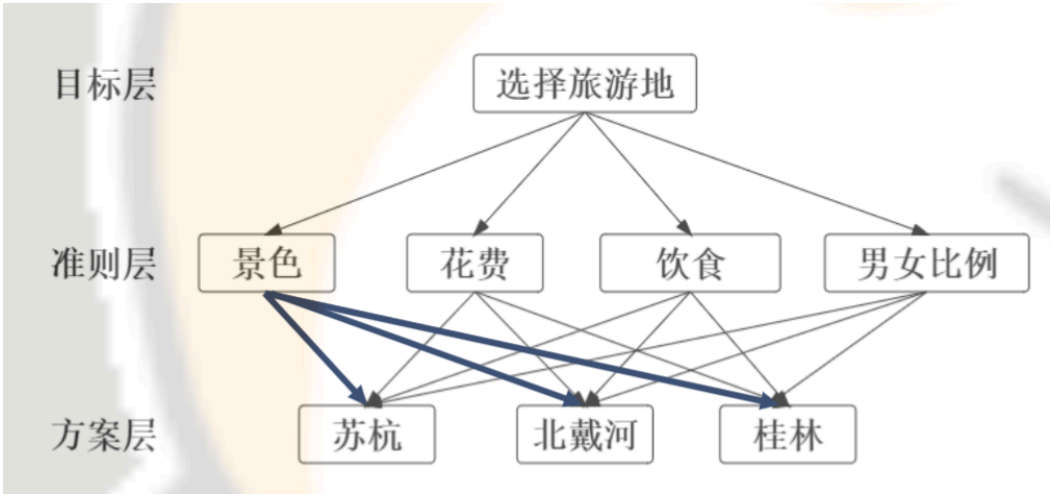
层次分析法

In [ ]:
```
import numpy as np
```

解题步骤

1. 解决评价类问题

2. 画出层级结构图（目标层、准则层、方案层）（出现在论文里）



3. 构造判断矩阵（确定评价指标孰重孰轻）

> 有多少个评价标准，就有多少个判断矩阵和一个总的判断矩阵

4. 依照评价指标对各个方案进行打分

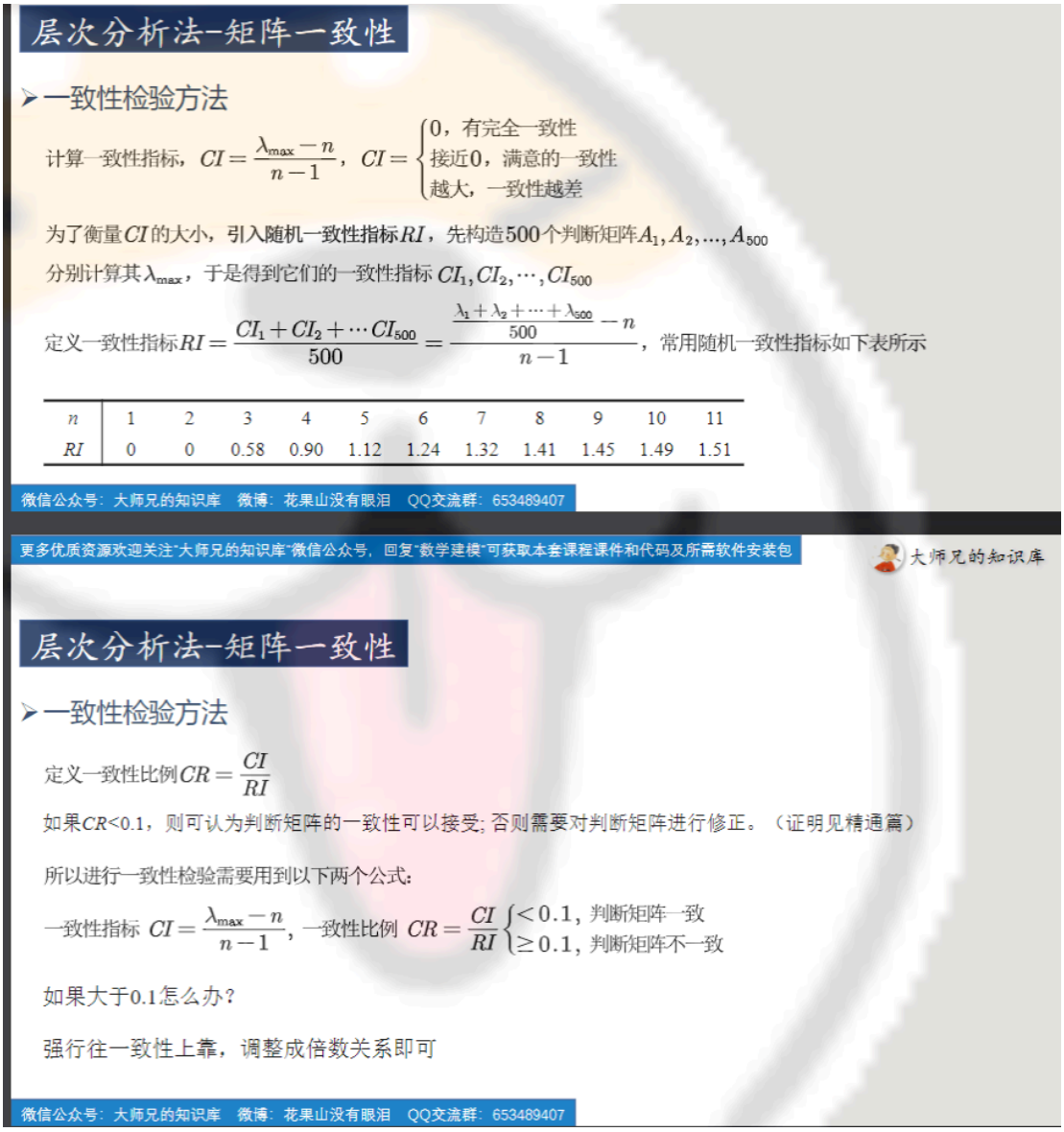
景色	苏杭	北戴河	桂林
苏杭	1	2	5
北戴河	1/2	1	2
桂林	1/5	1/2	1

5. 判断一致性

6. 求出权重，填表，求得最后得分

7. 层次总排序一致性检验

计算一致性比例



- 以下为 Matlab 代码：

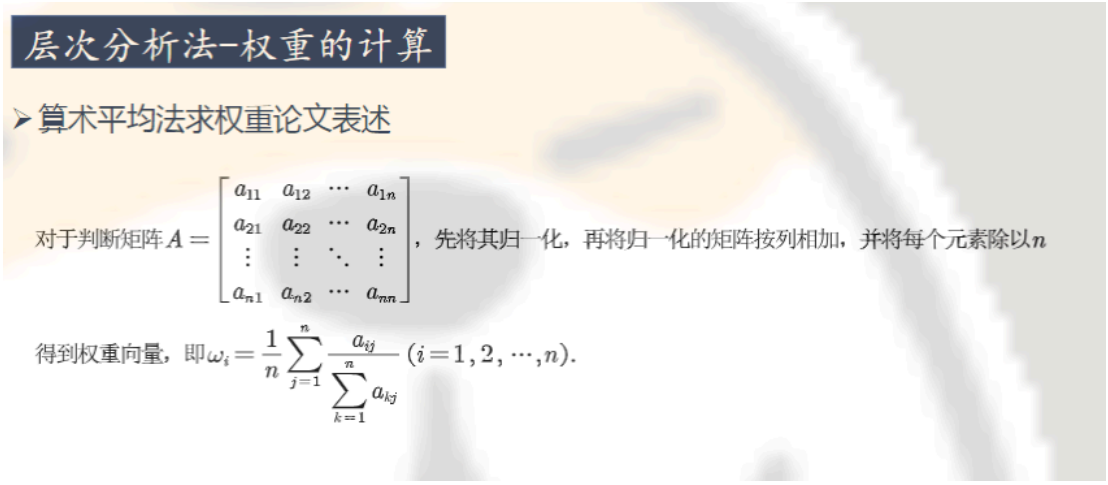
```
CI = (Max_eig - n) / (n-1);
% Max_eig 为最大特征值，n 为判断矩阵的维度。
% 最大特征值的计算在特征值法求权重时会介绍。

RI=[0 0.0001 0.52 0.89 1.12 1.26 1.36 1.41 1.46 1.49 1.52 1.54 1.56
1.58 1.59];
% 这里的 RI 需要查表。
% 这里 n=2 时，RI=0，我们为了避免分母为0，将这里的第二个元素改为了很接近0
的正数，即 0.0001。

CR=CI/RI(n);
disp('最大特征值为:');
disp(Max_eig);
disp('一致性指标CI=');disp(CI);
disp('一致性比例CR=');disp(CR);
if CR<0.10
    disp('CR<0.10，该判断矩阵A的一致性可以接受!');
else
    disp('注意：CR >= 0.10，该判断矩阵需要进行修改!');
end
```

权重的计算

算数平均法



以下图判断矩阵为例：

景色	苏杭	北戴河	桂林
苏杭	1	2	5
北戴河	1/2	1	2
桂林	1/5	1/2	1

1. 输入矩阵 A。

In [ ]:
```
A = [[1, 2, 5], [1 / 2, 1, 2], [1 / 5, 1 / 2, 1]]
A = np.array(A)
print(A)
```

```
[[1.  2.  5. ]
 [0.5 1.  2. ]
 [0.2 0.5 1. ]]
```

2. 先将矩阵按列求和。

In [ ]:
```
Sum_A = np.sum(A, axis=0)
print(Sum_A)
```

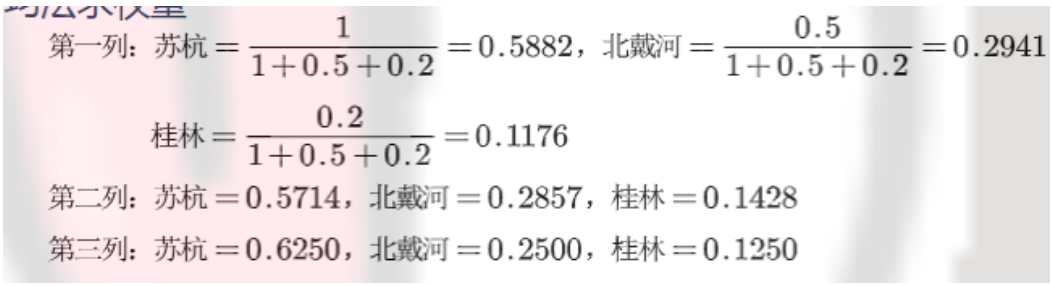
```
[1.7 3.5 8. ]
```

3. 再将矩阵扩大到三行。

In [ ]:
```
size = len(A)
SUM_A = np.tile(Sum_A, (size, 1))
print(SUM_A)
```

```
[[1.7 3.5 8. ]
 [1.7 3.5 8. ]
 [1.7 3.5 8. ]]
```

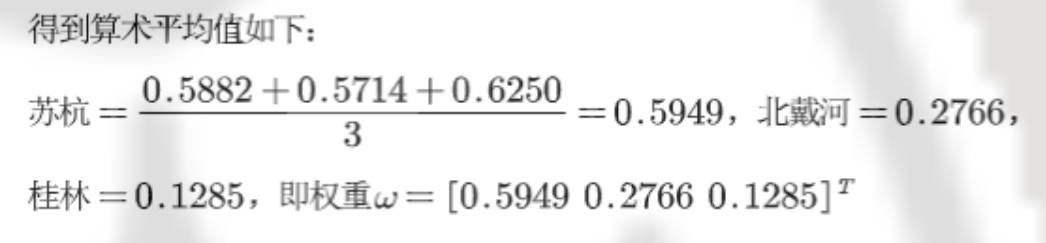
4. 将A中的每一个元素与 SUM_A 中的对应元素相除，得到标准化矩阵。



In [ ]:
```
Stand_A = A / SUM_A
print(Stand_A)
```

```
[[0.58823529 0.57142857 0.625     ]
 [0.29411765 0.28571429 0.25      ]
 [0.11764706 0.14285714 0.125     ]]
```

5. 把三列的结果按行相加，再做一个算数平均值



In [ ]:
```
w1 = np.sum(Stand_A, axis=1) / np.size(Stand_A, axis=1)
print(w1)
```

```
[0.59488796 0.27661064 0.1285014 ]
```

特征值法

1. 先求特征值和特征向量

```
[V,D] = eig(A); %% D是特征值，V是特征向量
Max_eig = max(max(D));% 先按列求最大值，得到行向量，再从这个向量里面求最大值
[r,c]=find(D == Max_eig , 1);
```

In [ ]:
```
V, D = np.linalg.eig(A)

print("特征向量:")
print(D)
print("\n特征值:")
print(V)
```

```
特征向量:
[[-0.89021421+0.j         -0.89021421+0.j         -0.89021421-0.j        ]
 [-0.41320083+0.j          0.20660042+0.35784242j  0.20660042-0.35784242j]
 [-0.19179084+0.j          0.09589542-0.16609574j  0.09589542+0.16609574j]]

特征值:
[ 3.00553511e+00+0.j         -2.76755587e-03+0.12895082j
 -2.76755587e-03-0.12895082j]
```

2. 求出最大特征值对应的特征向量，并确认最大特征值的位置

matlab 代码:

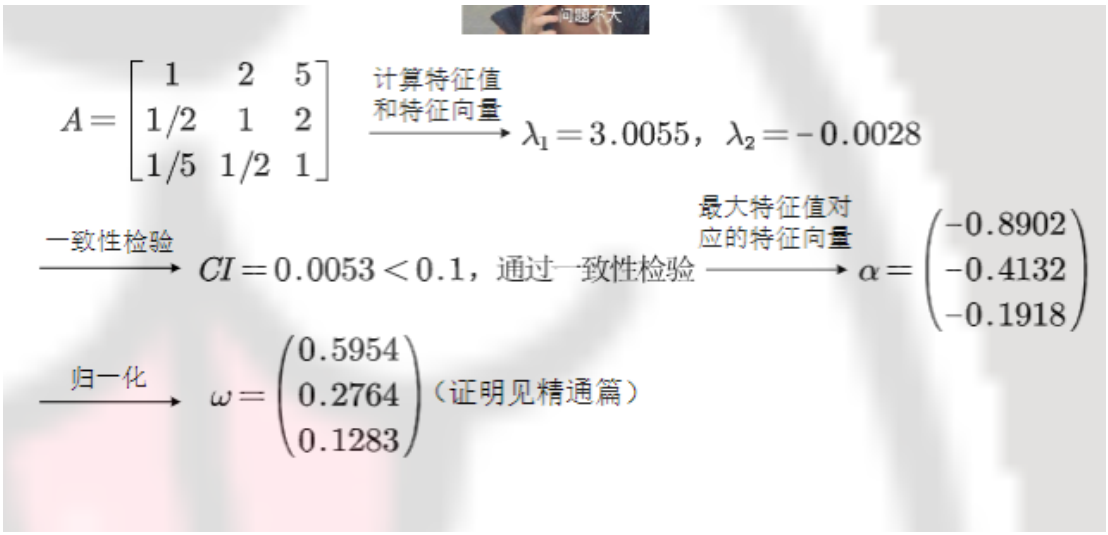
```
Max_eig = max(max(D));
 % 先按列求最大值，得到行向量，再从这个向量里面求最大值
 % 结果是 3.0055

[r,c]=find(D == Max_eig , 1);
% 确认最大特征值的位置
% r = 1
% c = 1
% 在第一行第一列
```

3. 把第 c 列中的元素除以第 c 列所有元素的和

```
w2 = V(:,c) ./ sum(V(:,c));
disp(w2)
```

总过程：



In [ ]:
```
# 求特征值中的最大值
Max_eig = np.max(V)
Max_eig
# 找到特征值中最大值对应的行和列
r = np.where(V == Max_eig)

print("\n特征值中的最大值:", Max_eig)
```

特征值中的最大值: (3.0055351117384976+0j)

In [ ]:
```
w2 = D[:, r] / np.sum(D[:, r])
w2
```

Out[ ]:
```
array([[[0.59537902-0.j]],

       [[0.27635046-0.j]],

       [[0.12827052-0.j]]])
```