```
clear;clc;
load X.mat;
%获取行数列数
r = size(X,1);
c = size(X,2);
%首先，把我们的原始指标矩阵正向化
%第六列中间型--->极大型
middle = input("请输入最佳的中间值：");
%中间值为 1
M = max(abs(X(:,6)-middle));
for i=1:r
      X(i,6) = 1-abs(X(i,6)-middle)/M;
end
disp("正向化后的矩阵为：");
```

正向化后的矩阵为：

```
disp(X);
```

```
   1.0e+04 *

    0.0021    0.0002    0.0030    0.0105    0.0001    0.0001    0.0399
    0.0015    0.0001    0.0012    0.0552    0.0001         0    0.0343
    0.0049    0.0001    0.0050    7.7868    0.0004    0.0001    0.0510
    0.0049    0.0001    0.0055    0.2595    0.0003    0.0001    0.0517
    0.0064    0.0002    0.0041    1.1640    0.0003    0.0000    0.0534
    0.0070    0.0001    0.0051    0.0627    0.0003    0.0001    0.0537
    0.0029    0.0001    0.0042    0.3599    0.0004    0.0001    0.0493
```

```
%把正向化后的矩阵进行预处理，消除量纲的影响
avg = repmat(mean(X),r,1);
new_X = X./avg;
%将预处理后的矩阵每一行的最大值取出，当成母序列
Y = max(new_X,[],2);
%计算各个指标和母序列的灰色关联度
%先把 new_X 矩阵所有元素都减去母序列中同行的元素，并取绝对值
Y2 = repmat(Y,1,c);
new_X = abs(new_X-Y2);
a = min(min(new_X)); %全矩阵最小值
b = max(max(new_X)); %全矩阵最大值
ro = 0.5;
new_X = (a+ro*b)./(new_X+ro*b);
disp("各个指标对于母序列的灰色关联度为：");
```

各个指标对于母序列的灰色关联度为：

```
gamma = mean(new_X)
```

```
gamma = 1×7
    0.7847    0.7869    0.7734    0.7138    0.7721    0.7836    0.7757
```

```
%计算各个指标的权重
disp("各个指标的权重为：");
```

各个指标的权重为：

```
weight = gamma./(sum(gamma,2))
```

```
weight = 1×7
    0.1456    0.1460    0.1435    0.1324    0.1432    0.1454    0.1439
```

```
%-------------------------------------------------------------------------------------------------------
%继续 TOPSIS 的步骤：对正向化后的矩阵 X 进行标准化（原矩阵除以每一列元素平方之和的开方）
temp1 = X.*X;                %先让每每一个元素平方
temp2 = sum(temp1);          %再对每一列求和
temp3 = temp2.^0.5;          %再把结果开方
temp4 = repmat(temp3,r,1);   %把开方后的结果按行复制 r 行
disp("******标准化后的矩阵为：");
```

******标准化后的矩阵为：

```
Z = X./temp4                 %原矩阵除以每一列元素平方之和的开方
```

```
Z = 7×7
    0.1659    0.4876    0.2691    0.0013    0.1393    0.4464    0.3134
    0.1202    0.3716    0.1087    0.0070    0.0988         0    0.2692
    0.4001    0.4121    0.4479    0.9874    0.4844    0.4035    0.4008
    0.3924    0.1879    0.4840    0.0329    0.3924    0.4001    0.4059
    0.5209    0.5033    0.3655    0.1476    0.3799    0.2457    0.4194
    0.5648    0.2942    0.4568    0.0080    0.4208    0.4450    0.4220
    0.2316    0.2812    0.3724    0.0456    0.5108    0.4686    0.3872
```

```
Z_max = max(Z)               %获得 Z 每一列中最大的元素
```

```
Z_max = 1×7
    0.5648    0.5033    0.4840    0.9874    0.5108    0.4686    0.4220
```

```
Z_min = min(Z)               %获得 Z 每一列中最小的元素
```

```
Z_min = 1×7
    0.1202    0.1879    0.1087    0.0013    0.0988         0    0.2692
```

```
D_max = sum(weight.*(Z-repmat(Z_max,r,1)).^2,2).^0.5
```

```
D_max = 7×1
    0.4244
    0.4882
    0.0783
    0.3771
    0.3246
    0.3671
    0.3778
```

```
D_min = sum(weight.*(Z-repmat(Z_min,r,1)).^2,2).^0.5
```

```
D_min = 7×1
    0.2159
    0.0702
    0.4598
    0.2635
    0.2712
    0.3079
    0.2675
```

```
disp("该矩阵得分为：")
```

该矩阵得分为：

```
S = D_min./(D_max+D_min)
```

```
S = 7×1
    0.3371
    0.1258
    0.8545
    0.4113
    0.4552
    0.4561
    0.4145
```

```
disp("矩阵归一化后得分为：");
```

矩阵归一化后得分为：

```
S = S./(repmat(sum(S),r,1))
```

```
S = 7×1
    0.1104
    0.0412
    0.2797
    0.1347
    0.1490
    0.1493
    0.1357
```

```

```