function [Dd,D,E]=TOPSIS(A,B)

**%参数介绍：**

**%参数矩阵A是原始数据组成的矩阵。**

**%参数矩阵B是一个行矩阵，每一个元素由1或0组成（1代表该变量为高优指标；0代表该变量为底优指标）。**

**%返回值介绍：**

**%矩阵Dd是数据同趋势化后的结果。**

**%矩阵D是数据归一化后的结果。**

**%矩阵E是最后数据的优距、劣距、与最优接近程度、排序结果。**

[m,n]=size(A);

I=[];

D=[];

Dd=[];

Aup=[];

Adown=[];

Diup=[];

Didown=[];

for k=1:n **%计算出归一化的结果并存入矩阵D**

if B(k)==1

Ak=A(:,k);

Dd=[Dd,Ak];

elseif B(k)==0

Ak1=1./(A(:,k));

Dd=[Dd,Ak1];

end

end

for k=1:n

Dk=Dd(:,k)/sqrt(sum(Dd(:,k).^2));

D=[D,Dk];

end

for k=1:n

Aup=[Aup,max(D(:,k))]; **%计算出最优方案并存入矩阵Aup**

Adown=[Adown,min(D(:,k))]; **%计算出最劣方案并存入矩阵**Adown

end

for k=1:m

Diup=[Diup;sqrt(sum((Aup-D(k,:)).^2))];

**%计算出所有指标离最优方案的距离，并存入矩阵Diup**

Didown=[Didown;sqrt(sum((Adown-D(k,:)).^2))];

**%计算出所有指标离最劣方案的距离，并存入矩阵Didown**

end

C=Didown./(Didown+Diup); **%计算评价对象与最优方案的接近程度**

C1=sort(C,'descend');

for k=1:m

for w=1:m

if C(k)==C1(w)

I=[I;w];

end

end

end

E=[Diup,Didown,C,I];