## 基于的工具包

想想，我能为下一届学习系统工程的学生做些什么呢？

为了深入学习层次分析法的方法，方便后来人继续研究。

基于制作工具包，得到层次分析法的计算机自动求解算法包，包括正规精准特征值算法，和积法，方根法。

留给下一届学弟学妹们计算矩阵的时候不需要那么费劲巴拉地摁计算器吧！超级好用的！开源版!

注释已经写得很明白了，无需废话，请看：

clear all;

Data;

%肖P76表3-13

disp("EigenVector\_EigenValue\_standard");

eig\_st(estimate\_matrix);

disp("power\_root");

power\_root(estimate\_matrix);

disp("sum\_product");

sum\_product(estimate\_matrix);

%剩下的根据不同题目的不同具体情况作出最终总排序表格，得到总排序

%计算机标准特征根特征值eig(Matrix)法——应该是用的计算方法里面的幂法，或者经过并行优化

function [Max\_Eigenvector,Max\_Eigenvalue]=eig\_st(matrix)

%先编辑数据保存在Data中，运行Data，显示到命令行，有矩阵数据:A\_Bi,B1\_Ci,B2\_Ci,B3\_Ci,B4\_Ci

n=size(matrix,2);%[m,n]=size(Matrix)

[Eigenvector,Eigenvalue]=eig(matrix);

%特征根Eigenvalue特征向量Eigenvector

%按列归一化Eigenvector./repmat(sqrt(sum(Eigenvector.^2,1)),size(Eigenvector,1),1);

for i=1:n

Eigenvector(:,i)=Eigenvector(:,i)/norm(Eigenvector(:,i));

end

diag\_Eigenvalue=diag(Eigenvalue);%对角矩阵变成向量

Max\_Eigenvalue=max(diag\_Eigenvalue)%求解最大特征值

index=find(diag\_Eigenvalue==Max\_Eigenvalue);Max\_Eigenvector=Eigenvector(:,index);

ci=CI(Max\_Eigenvalue,n);RI\_List=[0,0,0.52,0.89,1.12,1.26,1.36,1.41,1.46,1.49,1.52,1.54,1.56,1.58];

if n>2

RI=RI\_List(n);

else

disp("n<3,1/RI=1/0,let RI=eps(Very small)");RI=eps;

end

CR=ci/RI

if CR<0.1

disp("Pass,C.R<0.1")

else

disp("Fail,C.R>0.1")

end

%方根法,行乘开方根归一

function [W0,LambdaMax]=power\_root(matrix)

%先编辑数据保存在Data中，运行Data，显示到命令行，有矩阵数据:A\_Bi,B1\_Ci,B2\_Ci,B3\_Ci,B4\_Ci

n=size(matrix,2);%[m,n]=size(Matrix)%特征根Eigenvalue特征向量Eigenvector

format short

Wi=(prod(matrix,2)).^(1/n);W0=Wi./sum(Wi)%行乘根归一

LambdaMi=matrix\*W0./W0

LambdaMax=sum(matrix\*W0./W0)/n%根据特征向量求特征值

cI=CI(LambdaMax,n)

RI\_List=[0,0,0.52,0.89,1.12,1.26,1.36,1.41,1.46,1.49,1.52,1.54,1.56,1.58];%汪书P125随机数1000次跑出来的结果

if n>2

RI=RI\_List(n);

else

disp("n<3,1/RI=1/0,let RI=eps(Very small)");RI=eps;

end

CR=cI/RI

if CR<0.1

disp("Pass,C.R<0.1")

else

disp("Fail,C.R>0.1")

end

%和积法——列归行加再归一

function[W0,LambdaMax]=sum\_product(matrixOld)

%先编辑数据保存在Data中，运行Data，显示到命令行，有矩阵数据:A\_Bi,B1\_Ci,B2\_Ci,B3\_Ci,B4\_Ci

n=size(matrixOld,2);%[m,n]=size(Matrix)%特征根Eigenvalue特征向量Eigenvector

format short

%按列归一化

for i=1:n

matrixNew(:,i)=matrixOld(:,i)/norm(matrixOld(:,i));

end%列归一

Wi=sum(matrixNew,2);W0=Wi./sum(Wi)%行加再归一

LambdaMax=sum(matrixOld\*W0./W0)/n%根据特征向量求特征值

ci=CI(LambdaMax,n);

RI\_List=[0,0,0.52,0.89,1.12,1.26,1.36,1.41,1.46,1.49,1.52,1.54,1.56,1.58];;%汪书P125随机数1000次跑出来的结果

if n>2

RI=RI\_List(n);

else

disp("n<3,1/RI=1/0,let RI=eps(Very small)");RI=eps;

end

CR=ci/RI

if CR<0.1

disp("Pass,C.R<0.1")

else

disp("Fail,C.R>0.1")

end

## 稳态平衡之我看

假设,,公司现阶段市场占比30%，40%，30%，假设满足马尔可夫状态转移方程，状态转移情况如下：

1. 本月购买者下月购买,,的概率0.75 0.2 0.05
2. 者下月购买,,的概率0.1 0.7 0.2
3. 者下月购买,,的概率0.1 0.1 0.8

即

动态平衡（稳态）就是求

而根据代数常识告诉我们必定趋向极限或者周期循环,求解PX=X就是稳态。

>> p^32=

0.28572 0.32143 0.39286

0.28571 0.32143 0.39286

0.28571 0.32143 0.39286

>> p^33=

0.28571 0.32143 0.39286

0.28571 0.32143 0.39286

0.28571 0.32143 0.39286

就是稳态市场占比。

也可以理解为稳态下X占比不变求解方程组：

且归一化条件：，

求解：就是稳态市场占比

变换一种新的理解方式（类基尔霍夫定律，出=入）:

且归一化条件：

同样可以求解答案就是稳态市场占比。