**模式识别——Isodata编程实习报告**

**姚毅2014302590137**

**武汉大学-模式识别课程**

本程序使用matlab编写，程序实现了isodata的全部功能，总用时：7小时，其中1小时回顾理解算法，2小时编写代码，3小时调试。1小时撰写编程实习报告。

经过测试程序可以运行，并且程序得到的分类结果与书上例子、网上基本数据例子和相应结果相符，程序可靠性很高。另外，程序的亮点有：

**1.数据自由无限制**，读取txt文件中的数据，自动识别特征数目以及样本数目。使用者可自由更换数据，支持多特征多类别。

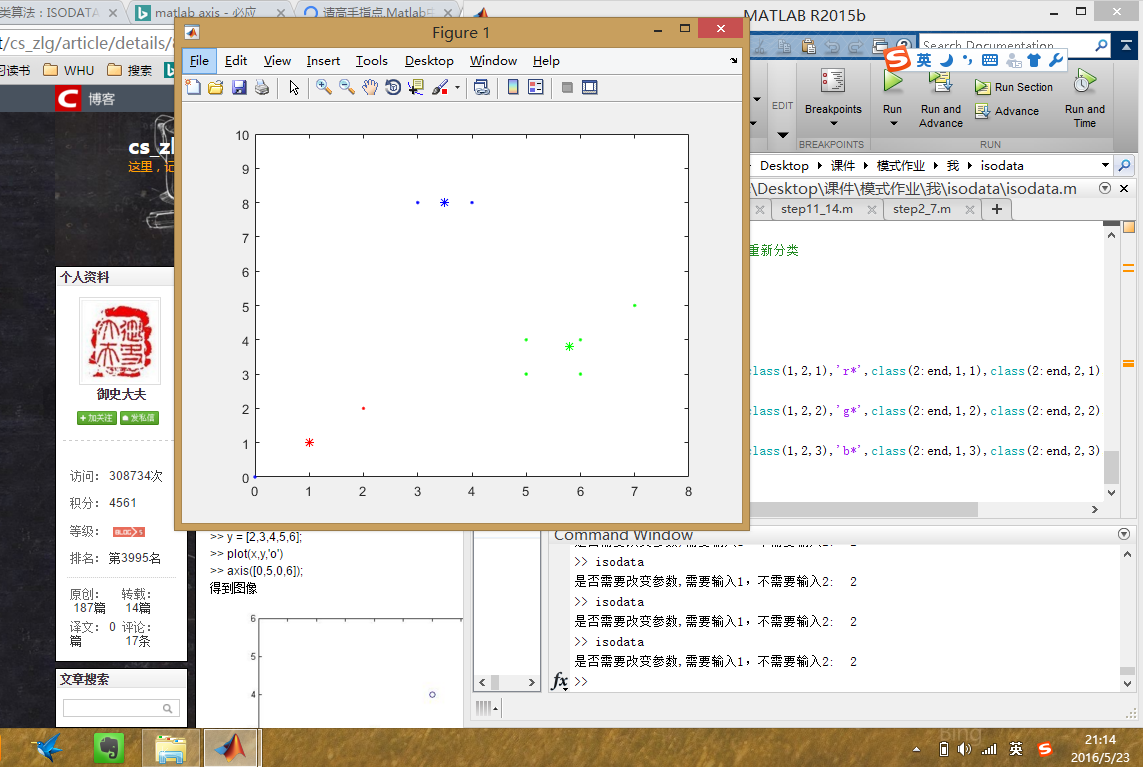
**2.程序参数课选择**，采用命令输入的形式，用户可自由选择是否改变参数。

本实习报告主要从以下4方面讲解本次实习中遇到的问题以及算法相关：

3.**实现了对遥感图像的处理。**采用命令输入的形式，可以选择数据来源，可以是txt的数据文件，也可以是遥感图像

1. **Isodata算法代码化过程，程序中遇到的问题以及解决方法**
2. **程序可行性测试以及结果输出**
3. **程序性能以及算法评价**
4. **附录：matlab源代码；**

程序分类结果：星号代表聚类中心。



**一、Isodata算法代码化过程以及相关问题**

**1）初始参数的输入：**

建立一系列单double空间存储参数。

**2）读取输入数据：**

使用textscan函数读取txt中的数据，并且读取其样本数目以及特征数目。

1. **初始分类：**

创建一个class的三维空间，其中一二三维分别代表：样本行、特征列、类别平面；class的一二维大小由txt中读取出的数据量决定。第三维类别数目由初始参数决定，在之后大小会改变（因为中心分裂有可能数据超出原大小，后文提及具体变化方法。）

各类初始样本中心为自由随机选择的样本。

计算样本与中心距离并按最小欧式距离分类。各个样本写进class空间。

**问题：**创建一个Center空间，存储各类的中心，由于中心会不断改变，并且会有新的类别中心加入进来，同时也有的类别不做改变，编程时如何让计算机识别出哪个中心应该改变哪个不改变，并且做到不将原类别中心不覆盖呢？

**解决方法：**

%给中心初始值，并且第一列标定是否这个空间为空。1有类，0为空，-1为删除了

Center=zeros(temp,n+1);

Center(1:Nc,2:n+1)=C\_data(1:Nc,:);

Center(1:Nc,1)=1;%有,

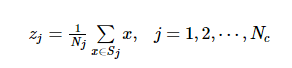
即多加一列，第一行存储这个类的信息，是否被删除（可能是样本数太少或是被分裂），或者此处空间为可用，或者此处空间有类别占领不可用。

**Tips：**为了读取处理方便，class第一行为各类的样本中心。

1. **判断样本数目与初始参数的关系，过小时取消该样本类。**

这里使用 Center(i,1)=-1;%delete标志。 作为删除标志，之后重新分类时将第i类的样本全部重新分类。

1. **修正聚类中心，各类别求平均中心。**



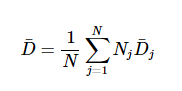
这里对于matlab来说是一个sum函数，之后求平均值即可。

**5）计算各聚类域Sj中模式样本与各聚类中心间的平均距离：**

这里由于有多个类别，需要遍历各个类内的样本，这是一层循环，另外需要新建一个空间来存储各个样本与中心的距离之后再统一求和。需要一个临时的变量。



**6）计算全部模式样本的对应聚类中的总平均距离：**



**7）判断分裂、合并及迭代运算：**

条件一：若迭代运算次数已达到I次，即最后一次迭代，则置θc =0，转至第十一步。

若Nc≤K2Nc≤K2，即聚类中心的数目小于或等于规定值的一半，则转至第八步，对已有聚类进行分裂处理。

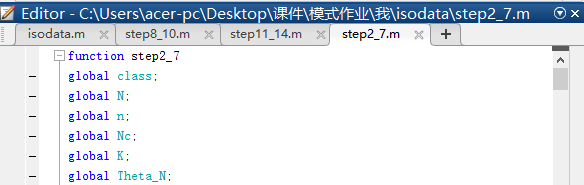
条件二：若迭代运算的次数是偶数次，或Nc≥2KNc≥2K，不进行分裂处理，转至第十一步；

条件三：否则（即既不是偶数次迭代，又不满足Nc≥2KNc≥2K），转至第八步，进行分裂处理。

这部是整个算法中比较困难的一部分，更是核心的一部分，这里针对几个问题作出解释：

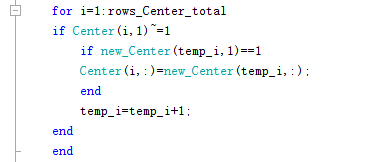
1. **如何实现“跳到第x步”：**

使用函数，将整个算法分为四大块：起始、重新分类（2-7步）、分裂（8-10步）、合并（11-14步），每一步对应一个函数，每次判断条件时决定了是否调用函数，而至于共有的变量和空间，需要定义全局变量：



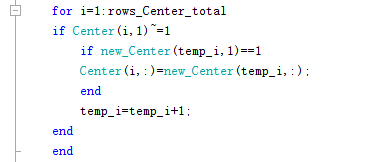
**2）如何实现类分裂：**

我的解决方法是定义了一个new\_Center空间，用来存储新的空间，需要注意的是，分裂后原类别的类不再存在，需要将其删除。我采用的方法是每次重新分类之前，将所有类空间全部擦除： class(:,:,:)=0; 之后再根据新的类中心分类，原始数据不再保留以免出错，如出现某样本重复在多个类中，这一切是根据聚类中心决定的，而聚类中心的第一列存储着删除、保留、空白的信息，分裂时将此值赋为-1，意为删除，之后用新的中心覆盖。



**3）如果分裂后类数目大于了K，也大于了初始定义的空间大小怎么办：**

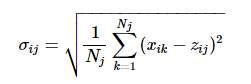
计算现在的大小，与原来的空间大小做比较，如果超过，重新建立一个足够大的空间



**8） 计算每个聚类中样本距离的标准差向量**



其中向量的各个分量为：



式中，i = 1, 2, …, n为样本特征向量的维数，j = 1, 2, …, Nc为聚类数，Nj为Sj中的样本个数。

1. **求每一标准差向量**

{σj, j = 1, 2, …, Nc}中的最大分量，以{σjmax, j = 1, 2, …, Nc}代表。

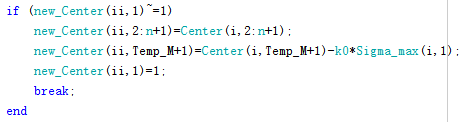
**10）判断分裂：**

在任一最大分量集{σjmax, j = 1, 2, …, Nc}中，若有σjmax>θS ，同时又满足如下两个条件之一：

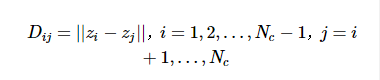
1. D¯j>D¯ 和Nj > 2(θN + 1)，即Sj中样本总数超过规定值一倍以上，
2. Nc≤K2Nc≤K2

     则将zj 分裂为两个新的聚类中心和，且Nc加1。 中对应于σjmax的分量加上kσjmax，其中；中对应于σjmax的分量减去kσjmax。

如果分裂，转至第二步。



1. **计算全部聚类中心的距离**

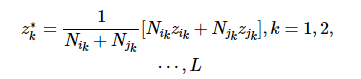


1. **比较类间距离与初始参数的关系，判断是否发生合并：**

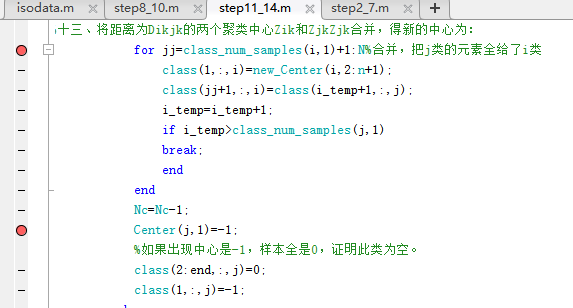
如果距离小于初始参数，发生合并操作。

1. **合并操作：**

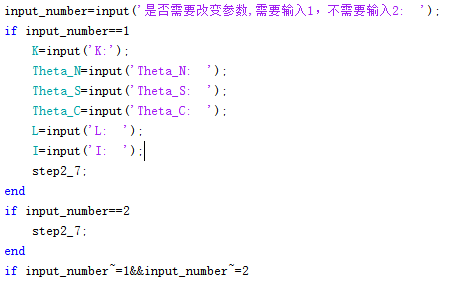
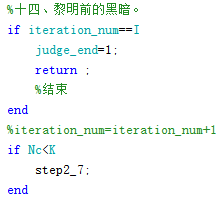
得到的新的中心为：



难点：需要将两类的所有样本分类至新的中心代表的类别中，解决方法如下：



1. **最后一步，判断是****否重新输入参数，以及迭代次数是否增加。**

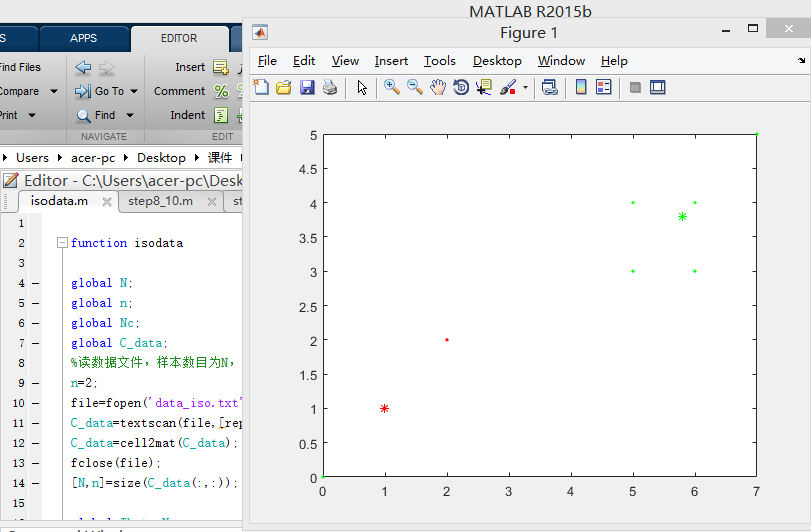


1. **程序可行性测试以及结果输出**

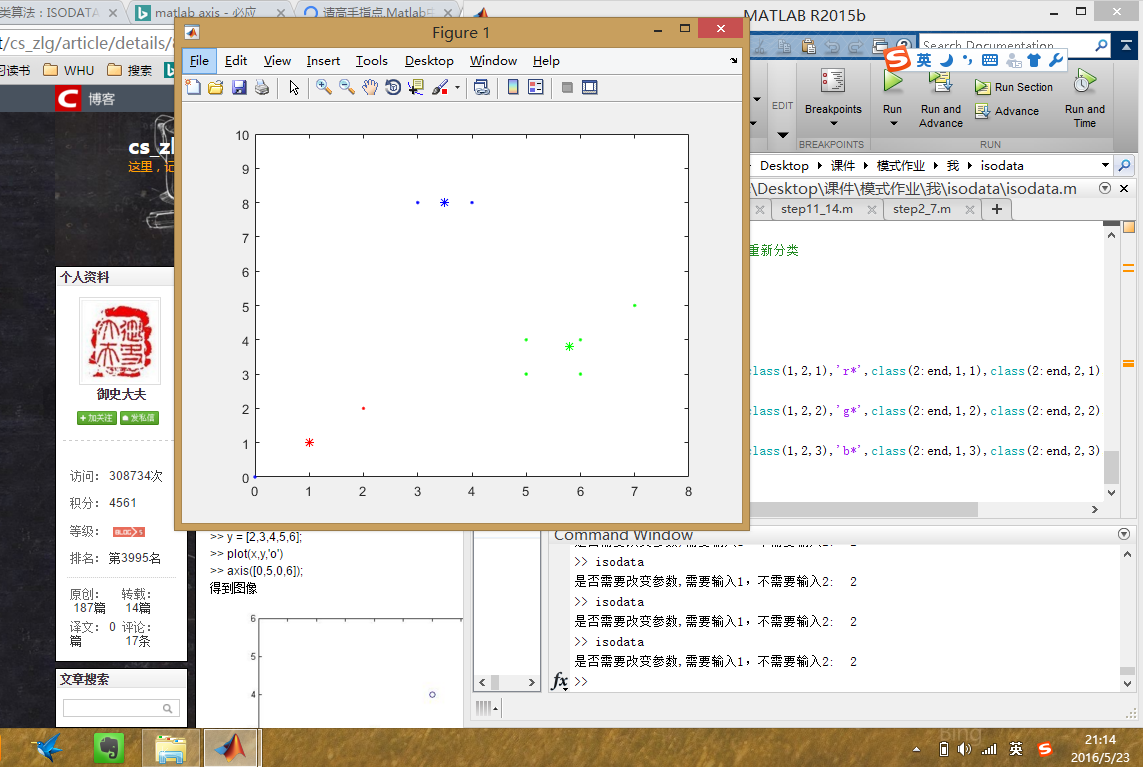
**共测试数据：5组，程序运行结果：良好。**

**其中网上提供的数据与结果1组，书上的例子1组，其余三组自我生成并且分类：**

**(结果在下一页)**

****

**书上例子分类结果**



**其他测试结果**

**三、附录：matlab源代码；**

**1、isodata主程序：**

function isodata

global N;

global n;

global Nc;

global C\_data;

%读数据文件，样本数目为N，特征数为n

n=2;

file=fopen('data\_iso.txt','r');

C\_data=textscan(file,[repmat('%f',[1,n])],'CollectOutput',1);

C\_data=cell2mat(C\_data);

fclose(file);

[N,n]=size(C\_data(:,:));

global Theta\_N;

global Theta\_S;

global Theta\_C;

global L;

global I;

global K;

%function parameter

%设置控制参数(一)

Nc=1;

K=2;Theta\_N=1;Theta\_S=1;Theta\_C=4;L=1;I=4;

if K>=Nc

temp=K;

else

temp=Nc;

end

% parameter;

%华丽的分割线，一堆空间-----------------------------------------------------------------

global class;

global class\_num\_samples;

global Center;

global D\_mean;

global delete;

global D\_temp;

global D\_mean\_total\_temp;

global Sigma;

global Sigma\_temp;

global Sigma\_max;

global D\_total;

global D\_total\_compare;

global new\_Center;

global D\_mean\_total;

global iteration\_num;

global D\_temp\_num;

global k0;

global Num\_Center;

global Num\_new\_Center;

global temp\_judge\_fenlie;

global judge\_end;

class\_num\_samples=zeros(temp,1);%很重要的参数，每个类别中有多少样本

%给中心初始值，并且第一列标定是否这个空间为空。1为是，0为空，-1为删除了

Center=zeros(temp,n+1);

Center(1:Nc,2:n+1)=C\_data(1:Nc,:);

Center(1:Nc,1)=1;%有,

delete=zeros(Nc\*999,1);

D\_mean=zeros(Nc\*999,1);%设置存储平均距离的空间，最多有Nc类。每类最多有N个样本

D\_temp=zeros(N,1);

D\_mean\_total\_temp=zeros(N,1);

Sigma=zeros(temp,n);

Sigma\_temp=zeros(N,n);

Sigma\_max=zeros(temp,1);

D\_total=zeros(Nc\*999,Nc\*999);

D\_total\_compare=zeros(Nc\*Nc,3);

new\_Center=zeros(999\*Nc,n+1);

new\_Center(:,1)=-1;%第一列标志一下是否被占用。-1为空，1为占了

D\_mean\_total=0;

iteration\_num=0;

D\_temp\_num=1;

k0=0.5;%分裂系数

Num\_Center=0; %中心数目

class=zeros(N+1,n,K);%每个类别中样本数最多是Num\_samples,故肯定不会t个类别全占满的

class(1,:,:)=-1;

%注意，class中第一行是中心。

for i=1:Nc

class(1,:,i)=Center(i,2:n+1);%随便选C\_data,Class中的每类的第一行是中心.

end

step2\_7;%基本运算与重新分类

%step8\_10;%分裂

%step11\_14;%合并

figure;

plot(class(1,1,1),class(1,2,1),'r\*',class(2:end,1,1),class(2:end,2,1),'r.');

hold on;

plot(class(1,1,2),class(1,2,2),'g\*',class(2:end,1,2),class(2:end,2,2),'g.');

hold on;

plot(class(1,1,3),class(1,2,3),'b\*',class(2:end,1,3),class(2:end,2,3),'b.');

axis([0 8 0 10]);

end

1. **step2\_7**

function step2\_7

global class;

global N;

global n;

global Nc;

global K;

global Theta\_N;

global Theta\_S;

global Theta\_C;

global L;

global I;

global class\_num\_samples;

global Center;

global D\_mean;

global delete;

global D\_temp;

global D\_mean\_total\_temp;

global Sigma;

global Sigma\_temp;

global Sigma\_max;

global D\_total;

global D\_total\_compare;

global new\_Center;

global D\_mean\_total;

global iteration\_num;

global D\_temp\_num;

global k0;

global Num\_Center;

global C\_data;

global Num\_new\_Center;

global temp\_judge\_fenlie;

iteration\_num=iteration\_num+1;

Num\_Center=0;

Num\_new\_Center=0;

temp\_Center=Center;

if temp\_judge\_fenlie==0

new\_Center(:,:)=0;

end

[rows\_Center\_total,aslkfjkslafjkl]=size(Center);

for i=1:rows\_Center\_total

if Center(i,1)==1

Num\_Center=Num\_Center+1;

end

if new\_Center(i,1)==1

Num\_new\_Center=Num\_new\_Center+1;

end

end

if (Num\_new\_Center+Num\_Center)>rows\_Center\_total

Center=zeros(Num\_new\_Center/2+Num\_Center,n+1);

for i=1:Num\_new\_Center+Num\_Center %给新创建的Center赋值。

if i<=Num\_Center

Center(i,:)=temp\_Center(1,:);

else

Center(i,:)=new\_Center(i,:);

end

end

else

temp\_i=1;

for i=1:rows\_Center\_total

if Center(i,1)~=1

if new\_Center(temp\_i,1)==1

Center(i,:)=new\_Center(temp\_i,:);

end

temp\_i=temp\_i+1;

end

end

end%结束新类别的赋值。

new\_Center(:,:)=0;

class\_num\_samples(:,1)=0;%重新数

class(:,:,:)=0;

D=zeros(N,Nc);

%（二）分给最近的聚类

for i=1:N

for j=1:Nc %！！！！！！！！！！！！！ 这里可能有错，因为Nc和K不等

if Center(j,1)==1

a=C\_data(i,:);

b=Center(j,2:n+1);

D(i,j)=sqrt((a-b)\*(a-b)');

end

end

[temp\_M,temp\_I]=min(D(i,:)');%属于第temp\_I类

class\_num\_samples(temp\_I,1)=class\_num\_samples(temp\_I,1)+1;

class(class\_num\_samples(temp\_I,1)+1,:,temp\_I)=C\_data(i,:);%给样本根据最小距离分类.之所以加一是因为第一行是中心

end

%由于第一行存了类别中心，所以+1

%（三）取消样本集与否,并且修正中心（一起搞才避免再计算的麻烦）。

for i=1:Nc

if class\_num\_samples(i,1)<Theta\_N%第i类中的样本个数

%Nc=Nc-1;!!!!!!!!!!!!!!这里不减一，以后仍然Nc次循环，遇到删除了的类别跳过去.

delete(i,1)=1;%应该delete第i个类

Center(i,1)=-1;%delete标志。

else

%修正聚类中心

Center(i,2:n+1)=sum(class(2:end,:,i))/class\_num\_samples(i,1);%新的聚类中心

class(1,:,i)=Center(i,2:n+1);

end

end

%（五）计算各聚类中模式样本与各聚类中心间的平均距离

D\_temp(:,:)=0;

D\_mean(:,:)=0;

for i=1:Nc

if delete(i,1)~=1

for j=1:class\_num\_samples(i,1)

a=class(j+1,:,i);

b=Center(i,2:n+1);

D\_temp(j,1)=sqrt((a-b)\*(a-b)');

end

D\_mean(i,1)=sum(D\_temp(1:class\_num\_samples(i,1),1))/class\_num\_samples(i,1);

end

end

%（六）计算全部模式样本和对应聚类中心的总平均值。

%D\_mean\_total=0;

for i=1:Nc

if delete(i,1)~=1%恩，自以为很吊，注意到了求的是非删除的全部总平均距离.

D\_mean\_total\_temp(i,1)=class\_num\_samples(i,1)\*D\_mean(i,1);

end

end

D\_mean\_total=sum(D\_mean\_total\_temp(:,1))/N;

%(七)判别分裂、合并和迭代运算。

if iteration\_num==I

Theta\_C=0;

step11\_14;%跳到十一步。

end

if Nc<=K/2

step8\_10;

%跳到第八步

end

if Nc>=2\*K %求余数的函数rem

step11\_14;

%十一步

end

if (Nc>K/2)&&(Nc<2\*K)

if rem(iteration\_num,2)==0

step11\_14;

end

if rem(iteration\_num,2)~=0

step8\_10;

end

end

end

1. **step8\_10**

function step8\_10

global N;

global n;

global Nc;

global Theta\_N;

global Theta\_S;

global Theta\_C;

global L;

global I;

global K;

global class\_num\_samples;

global Center;

global D\_mean;

global delete;

global D\_temp;

global D\_mean\_total\_temp;

global Sigma;

global class;

global Sigma\_temp;

global Sigma\_max;

global D\_total;

global D\_total\_compare;

global new\_Center;

global D\_mean\_total;

global iteration\_num;

global D\_temp\_num;

global k0;

global Num\_Center;

global temp\_judge\_fenlie;

%第(八)步 计算聚类中样本距离的标准差向量 每个类有特征数个标准差。

Sigma(:,:)=0;

Sigma\_temp(:,:)=0;

for i=1:Nc

if delete(i,1)~=1

for ii=1:n %循环特征

for j=1:class\_num\_samples(i)

% a=class(:,n,i);

a=class(j+1,ii,i);%第i类第j个样本的ii特征

b=Center(i,ii+1);%i类的中心的第ii特征

Sigma\_temp(j,ii)=(a-b)\*(a-b);

end

Sigma(i,ii)=sqrt(sum(Sigma\_temp(:,ii))/class\_num\_samples(i,1));

end

end

end

%(九)求每一个标准差向量中的最大分量，用存起来。

for i=1:Nc

if delete(i,1)~=1

[Temp\_I,Temp\_M]=max(Sigma(i,:)');

Sigma\_max(i,1)=Temp\_I;

end

%(十)在任一最大分良机中，若有Sigma\_max>Theta\_S,又同时满足两个条件之一，则分类成两个新的聚类中心。

%这里有问题，Nc的值和K的值，到底类别数目最大有多少？

if Sigma\_max(i,1)>Theta\_S

if class\_num\_samples(i,1)>2\*(Theta\_N+1)&&D\_mean(i,1)>D\_mean\_total

temp\_judge\_fenlie=1;

Center(i,1)=-1;%本身分裂的话，这个类也就没了

%分裂的第一类。

for ii=1:999\*Nc

if (new\_Center(ii,1)~=1)

new\_Center(ii,2:n+1)=Center(i,2:n+1);

new\_Center(ii,Temp\_M+1)=Center(i,Temp\_M+1)-k0\*Sigma\_max(i,1);

new\_Center(ii,1)=1;

break;

end

end

%分裂的第二类。

for jj=ii+1:999\*Nc

if new\_Center(jj,1)~=1

new\_Center(jj,2:n+1)=Center(i,2:n+1);

new\_Center(jj,Temp\_M+1)=Center(i,Temp\_M+1)+k0\*Sigma\_max(i,1);

new\_Center(jj,1)=1;

Nc=Nc+1;

break;

end

end

end

if Nc<=K/2

temp\_judge\_fenlie=1;

Center(i,1)=-1;%本身分裂的话，这个类也就没了,-1是删除

%分裂的第一类。

for ii=1:999\*Nc

if (new\_Center(ii,1)~=1)

new\_Center(ii,2:n+1)=Center(i,2:n+1);

new\_Center(ii,Temp\_M+1)=Center(i,Temp\_M+1)-k0\*Sigma\_max(i,1);

new\_Center(ii,1)=1;

break;

end

end

%分裂的第二类。

for jj=ii+1:999\*Nc

if new\_Center(jj,1)~=1

new\_Center(jj,2:n+1)=Center(i,2:n+1);

new\_Center(jj,Temp\_M+1)=Center(i,Temp\_M+1)+k0\*Sigma\_max(i,1);

new\_Center(jj,1)=1;

Nc=Nc+1;

break;

end

end

end

if temp\_judge\_fenlie==1

step2\_7;

end

end

end

end

**4、step11\_14**

function step11\_14

global class;

global N;

global K;

global n;

global Nc;

global Theta\_N;

global Theta\_S;

global Theta\_C;

global L;

global I;

global class\_num\_samples;

global Center;

global D\_mean;

global delete;

global D\_temp;

global D\_mean\_total\_temp;

global Sigma;

global Sigma\_temp;

global Sigma\_max;

global D\_total;

global D\_total\_compare;

global new\_Center;

global D\_mean\_total;

global iteration\_num;

global D\_temp\_num;

global k0;

global Num\_Center;

global temp\_judge\_fenlie;

global judge\_end;

temp\_judge\_fenlie=0;

%(十一)计算全部聚类中心的距离

for i=1:Nc-1

if delete(i,1)~=1;

for j=i+1:Nc

D\_total(i,j)=sqrt((Center(i,2:n+1)-Center(j,2:n+1))\*(Center(i,2:n+1)-Center(j,2:n+1))');

%十二、比较Dij和Theta\_c的值，将Dij <θc 的值按最小距离次序递增排列，即

%sort(D\_total\_compare);排序卵子用？

%合并i,j类,j类删除了，i类留着呢

if D\_total(i,j)<Theta\_C

for ii=1:999\*Nc %遍历new\_Center找到空间

if new\_Center(ii,1)~=1

new\_Center(i,2:n+1)=(class\_num\_samples(a)\*Center(a,2:n+1)+class\_num\_samples(b)\*Center(b,2:n+1))/(class\_num\_samples(a)+class\_num\_samples(b));

end

end

i\_temp=1;

%十三、将距离为Dikjk的两个聚类中心Zik和ZjkZjk合并，得新的中心为：

for jj=class\_num\_samples(i,1)+1:N%合并，把j类的元素全给了i类

class(1,:,i)=new\_Center(i,2:n+1);

class(jj+1,:,i)=class(i\_temp+1,:,j);

i\_temp=i\_temp+1;

if i\_temp>class\_num\_samples(j,1)

break;

end

end

Nc=Nc-1;

Center(j,1)=-1;

%如果出现中心是-1，样本全是0，证明此类为空。

class(2:end,:,j)=0;

class(1,:,j)=-1;

end

end

end

end

%十四、黎明前的黑暗。

if iteration\_num==I

judge\_end=1;

return ;

%结束

end

%iteration\_num=iteration\_num+1;

if Nc<K

step2\_7;

end

%这一块应该放在最上面的，每次迭代到11-14都会问一次-----------------------------------------------

input\_number=input('是否需要改变参数,需要输入1，不需要输入2: ');

if input\_number==1

K=input('K:');

Theta\_N=input('Theta\_N: ');

Theta\_S=input('Theta\_S: ');

Theta\_C=input('Theta\_C: ');

L=input('L: ');

I=input('I: ');

step2\_7;

end

if input\_number==2

step2\_7;

end

if input\_number~=1&&input\_number~=2

disp('error!Input again.\n!');

return;

end

%-----------------------------------------------

%跳与不跳

end