**模式识别——Kmeans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Kmeans的全部功能，总用时：4小时，其中2小时编写代码，1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。

本实习报告主要从以下X方面讲解本次实习中遇到的问题以及算法相关：

**2.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. **Kmeans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；**

**一、Kmeans****算法代码化过程以及相关问题**

**（一）算法原理：**

k-means 算法接受参数 k ；然后将事先输入的n个数据对象划分为 k个聚类以便使得所获得的聚类满足：同一聚类中的对象相似度较高；而不同聚类中的对象相似度较小。聚类相似度是利用各聚类中对象的均值所获得一个“中心对象”（引力中心）来进行计算的。

K-means算法是最为经典的基于划分的聚类方法，是十大经典数据挖掘算法之一。K-means算法的基本思想是：以空间中k个点为中心进行聚类，对最靠近他们的对象归类。通过迭代的方法，逐次更新各聚类中心的值，直至得到最好的聚类结果。

假设要把样本集分为c个类别，算法描述如下：

（1）适当选择c个类的初始中心；

（2）在第k次迭代中，对任意一个样本，求其到c个中心的距离，将该样本归到距离最短的中心所在的类；

（3）利用均值等方法更新该类的中心值；

（4）对于所有的c个聚类中心，如果利用（2）（3）的迭代法更新后，值保持不变，则迭代结束，否则继续迭代。

该算法的最大优势在于简洁和快速。算法的关键在于初始中心的选择和距离公式。

**（二）编程思想：**

1、随机选取k个聚类质心点（cluster centroids）为clip\_image008[6]。

2、 重复下面过程直到收敛 （聚类中心不再发生改变）{

对于每一个样例i，计算其应该属于的类

clip\_image009

对于每一个类j，重新计算该类的质心

clip\_image010[6] }

**编写代码时的问题及解决方法：**

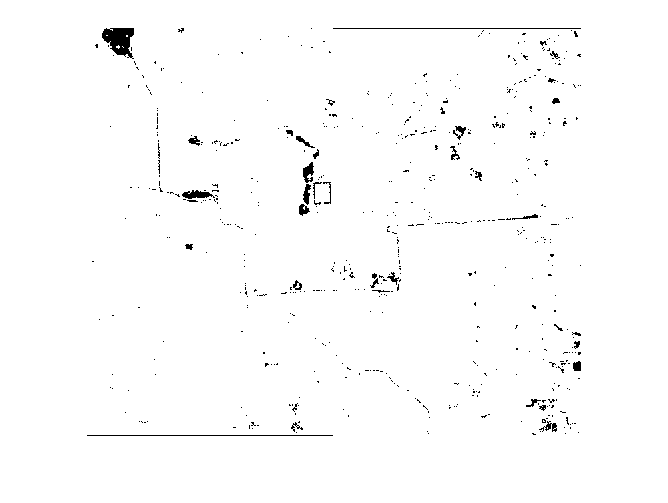
1. **txt和遥感图像中数据存储格式不统一的问题：**

由于txt格式是按：每一行代表一个样本，每列代表其相应的特征，也就是每行是代表一个特征向量（一个样本），matlab读取出来是二维的矩阵，编写程序时先是基于这种数据存储格式进行的数据处理，但是读取出tif图像文件后，matlab是使用3维矩阵存储的，第三维空间代表的是特征，于是需要将三维空间转换到二维空间，之后的数据处理才能够正常进行。

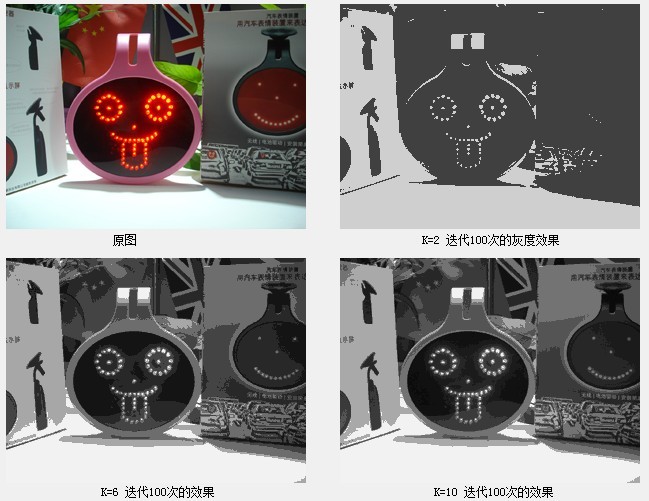
由于算法比较简单，其他没什么大问题。

1. **程序运行结果**

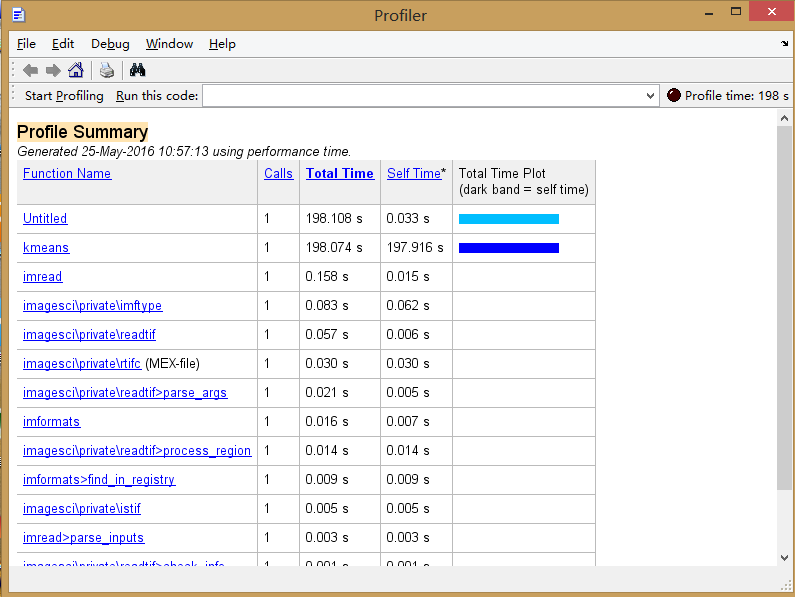
**对beijingcut.tif的分类结果：**

****

**可以看出，可以基本地将特别清晰的道路分类出来，水系也可以部分分类出来。我认为这与这张图本来的特征不是非常明显有关。**

**对普通图像的处理结果（rgb分别作为一种特征）：**

**三、程序性能及算法评价**



**四、附录：matlab源代码**

%K-means 算法，输入参数t为自定义类别数目(中心数目)，取前t个样本作为初始中心。n为特征数目

% function y=kmeans(t,n,datafile\_name)

% file=fopen(datafile\_name,'r');

function kmeans(t)

%data中样本每行为一个样本，列为特征，n为特征数量。最后C1是个m\*n的矩阵，m为样本数目。

%具体的textscan见help

%Delimiter后面的是指文件中每个数据用什么符号间隔开

choice=input('Choose your data file type: \n1:txt \n2:picture \n');

%C\_text=textscan(file,'%s',n,'Delimiter','')

if choice==1

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

C\_data=importdata('data.txt');

%C\_data=cell2mat(C\_data);

[Num\_samples,n]=size(C\_data);

%fclose(file);

end

if choice==2

% img=input('input your picture name');

img='beijingcut.tif';

C\_data\_temp=imread(img);

%将三维的数组格式转化一下，成为我写的数据格式。

[rows,lines,deep]=size(C\_data\_temp);

new\_img=zeros(rows,lines,3);

Num\_samples=rows\*lines;

n=deep;

for i=1:rows

for j=1:lines

for k=1:deep

C\_data((i-1)\*lines+j,k)=C\_data\_temp(i,j,k);

end

end

end

C\_data=double(C\_data);

end

if choice~=1&&choice~=2

disp('your input number is wrong, please reinput it!\n');

return;

end

%关于创建矩阵：http://blog.csdn.net/perfumekristy/article/details/8119861

Center=zeros(t,n);

%中心赋初值，不知道毛线 C\_data(1:t,1:n)直接取为毛错了,因为C\_data是cell类型的

for i=1:t

for j=1:n

Center(i,j)=C\_data(i,j);

end

end

temp\_Center=ones(t,n);%用来做判断的，前后中心一致性问题。

%Num\_samples=size(C\_data,1);

D=zeros(Num\_samples,t);

%创建分类数组,多维数组参考http://jingyan.baidu.com/article/5225f26b0a6650e6fa0908ea.html

class(:,:,:)=zeros(Num\_samples,n,t);%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

%迭代开始了，先求距离，再分类，再求新类别的中心

while temp\_Center~=Center%这里temp初值是ones.

%1.开始求距离,生成i j的矩阵，每行是一个点到t个中心各距离。

for i=1:Num\_samples%求C\_data的第1维大小

for j=1:t

a=C\_data(i,:);

b=Center(j,:);

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

end

end

%2.求距离并且分类

temp=ones(t,1);%每一类都需要一个temp来给他逐个样本写进去。

for i=1:Num\_samples

%备注：这里错了的话可以尝试D整体去求最小值，行列对应得转化一下。

[M,I]=min(D(i,:)');%min求最小值是按列求的，故转置。

%第i个样本， I就是其应属于的类别了

if choice==1

class(temp(I,1),:,I)=C\_data(i,:);%给样本根据最小距离分类.

temp(I,1)=temp(I,1)+1;

end

if choice==2

new\_rows=uint64(i/lines);

new\_lines=rem(i,lines);

new\_img(new\_rows,new\_lines,:)=255/(t-1)\*(I-1);

end

end

%3.求类中心均值了,变换中心了。

for i=1:t%i是类别号码

temp\_Center(i,:)=Center(i,:);%将旧的中心保存下来.

Center(i,:)=mean(class(:,:,i));%mean是按列求的均值，生成了n个均值（各个特征的）1\*n，这就是第i类的中心了

end

end

imshow(new\_img);

end