利用k-means算法实现图像分割

## 一、kmeans函数用法介绍

K-means聚类算法采用的是将N\*P的矩阵X划分为K个类，使得类内对象之间的距离最大，而类之间的距离最小。

**使用方法：**

[┈]=Kmeans(X, K, ┈,‘Name1’, ‘Val1’, ‘Name2’, ‘Val2’,┈)

**各输入参数介绍：**

X--- 数值矩阵

K--- 集群的数量，为正整数

**Name-Value对参数**

Name是参数名称而Value是对应的值。我们可以指定Name,Value变量，用符号分割。Name必须出现单引号（’’）。我们可以按照任何顺序指定多个名称和值对的参数Name1, Value1, ……, NameN, ValueN.这里只选取几个常用的参数-值对进行介绍。

### ‘Distance’-测量距离

**‘sqeuclidean’(默认)|‘cityblock’|‘cosine’|‘correlation’|‘hamming’**

基于不同的测量距离，kmeans函数计算出不同的聚类中心

‘sqeuclidean’---欧氏距离

‘cityblock’---绝对误差和，又称L1

‘cosine’---针对向量

‘correlation’---针对有时序关系的值

‘hamming’---只针对二进制数据

### ‘Replicates’- 利用新的初始聚类中心位置，重复聚类的次数

**1（默认）|正整数**

### 'Start'- 选择初始聚类重心位置的方法

**'plus'（默认值）| 'cluster'| 'sample'| 'uniform'| 数字矩阵 | 数值数组**

‘sample’---从X中随机选取K个质心点

‘uniform’---根据X的分布范围均匀的随机生成K个质心

‘cluster’---初始聚类阶段随机选取10%的X的子样本（此方法初始使用’sample’方法）

参考网址：https://cn.mathworks.com/help/stats/kmeans.html

## 二、利用matlab的实现结果

*Matlab的代码如下：*

k **=** 4**;** %K表示聚类中心的个数

p\_rgb **=** imread**(**'samplePicture.bmp'**);** %读入图像

figure**(),**imshow**(**p\_rgb**),**title**(**'samplePicture'**);** %显示原图像

p\_gray **=** rgb2gray**(**p\_rgb**);** %灰度图

%figure(),imshow(p\_gray),title('sampleGrayPicture'); %显示原灰度图像

%获取图像的长宽

**[**m**,** n**]** **=** size**(**p\_gray**);**

%将图像进行RGB-3通道分解

R **=** reshape**(**p\_rgb**(:,:,**1**),** m**\***n**,** 1**);** %得到一个m\*n行1列的矩阵，存储red值

G **=** reshape**(**p\_rgb**(:,:,**2**),** m**\***n**,** 1**);** %得到一个m\*n行1列的矩阵，存储green值

B **=** reshape**(**p\_rgb**(:,:,**3**),** m**\***n**,** 1**);** %得到一个m\*n行1列的矩阵，存储blue值

dat **=** **[**R G B**];**

RGB **=** kmeans**(**double**(**dat**),** k**,** 'Distance'**,** 'sqeuclidean'**,**'start'**,**'sample'**,** 'Replicates'**,** 4**);**

%重复聚类4次

result **=** reshape**(**RGB**,** m**,** n**);**

result **=** label2rgb**(**result**);** %转换为RGB图像

figure**(),** imshow**(**result**),**title**(**'samplePictureResult'**);** %显示分割结果

imwrite**(**result**,** 'seg\_pic.bmp'**);**

图片的处理效果如下：

原图：



分割结果（k = 2）



分割结果（k = 4）



## 三、疑问

在上述程序中，直接指定了k值。而论文中给出了k值的选取准则——将图像转化到HSV色彩空间，通过计算色调直方图来判定彩色图像中主色调的个数，进而自动地确定K值的大小。