

K-means算法的matlab程序

在“[K-means算法的matlab程序 \(初步\)](#)”这篇文章中已经用matlab程序对iris数据库进行简单的实现，下面的程序最终的目的是求准确度。

作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/>

1.采用iris数据库

iris_data.txt



```
5.1    3.5    1.4    0.2
4.9    3     1.4    0.2
4.7    3.2    1.3    0.2
4.6    3.1    1.5    0.2
5      3.6    1.4    0.2
5.4    3.9    1.7    0.4
4.6    3.4    1.4    0.3
5      3.4    1.5    0.2
4.4    2.9    1.4    0.2
4.9    3.1    1.5    0.1
5.4    3.7    1.5    0.2
4.8    3.4    1.6    0.2
4.8    3     1.4    0.1
4.3    3     1.1    0.1
5.8    4     1.2    0.2
5.7    4.4    1.5    0.4
5.4    3.9    1.3    0.4
5.1    3.5    1.4    0.3
5.7    3.8    1.7    0.3
5.1    3.8    1.5    0.3
5.4    3.4    1.7    0.2
5.1    3.7    1.5    0.4
4.6    3.6    1     0.2
5.1    3.3    1.7    0.5
4.8    3.4    1.9    0.2
5      3     1.6    0.2
5      3.4    1.6    0.4
5.2    3.5    1.5    0.2
5.2    3.4    1.4    0.2
```

4.7	3.2	1.6	0.2
4.8	3.1	1.6	0.2
5.4	3.4	1.5	0.4
5.2	4.1	1.5	0.1
5.5	4.2	1.4	0.2
4.9	3.1	1.5	0.2
5	3.2	1.2	0.2
5.5	3.5	1.3	0.2
4.9	3.6	1.4	0.1
4.4	3	1.3	0.2
5.1	3.4	1.5	0.2
5	3.5	1.3	0.3
4.5	2.3	1.3	0.3
4.4	3.2	1.3	0.2
5	3.5	1.6	0.6
5.1	3.8	1.9	0.4
4.8	3	1.4	0.3
5.1	3.8	1.6	0.2
4.6	3.2	1.4	0.2
5.3	3.7	1.5	0.2
5	3.3	1.4	0.2
7	3.2	4.7	1.4
6.4	3.2	4.5	1.5
6.9	3.1	4.9	1.5
5.5	2.3	4	1.3
6.5	2.8	4.6	1.5
5.7	2.8	4.5	1.3
6.3	3.3	4.7	1.6
4.9	2.4	3.3	1
6.6	2.9	4.6	1.3
5.2	2.7	3.9	1.4
5	2	3.5	1
5.9	3	4.2	1.5
6	2.2	4	1
6.1	2.9	4.7	1.4
5.6	2.9	3.6	1.3
6.7	3.1	4.4	1.4
5.6	3	4.5	1.5
5.8	2.7	4.1	1
6.2	2.2	4.5	1.5
5.6	2.5	3.9	1.1
5.9	3.2	4.8	1.8
6.1	2.8	4	1.3
6.3	2.5	4.9	1.5
6.1	2.8	4.7	1.2
6.4	2.9	4.3	1.3
6.6	3	4.4	1.4

6.8	2.8	4.8	1.4
6.7	3	5	1.7
6	2.9	4.5	1.5
5.7	2.6	3.5	1
5.5	2.4	3.8	1.1
5.5	2.4	3.7	1
5.8	2.7	3.9	1.2
6	2.7	5.1	1.6
5.4	3	4.5	1.5
6	3.4	4.5	1.6
6.7	3.1	4.7	1.5
6.3	2.3	4.4	1.3
5.6	3	4.1	1.3
5.5	2.5	4	1.3
5.5	2.6	4.4	1.2
6.1	3	4.6	1.4
5.8	2.6	4	1.2
5	2.3	3.3	1
5.6	2.7	4.2	1.3
5.7	3	4.2	1.2
5.7	2.9	4.2	1.3
6.2	2.9	4.3	1.3
5.1	2.5	3	1.1
5.7	2.8	4.1	1.3
6.3	3.3	6	2.5
5.8	2.7	5.1	1.9
7.1	3	5.9	2.1
6.3	2.9	5.6	1.8
6.5	3	5.8	2.2
7.6	3	6.6	2.1
4.9	2.5	4.5	1.7
7.3	2.9	6.3	1.8
6.7	2.5	5.8	1.8
7.2	3.6	6.1	2.5
6.5	3.2	5.1	2
6.4	2.7	5.3	1.9
6.8	3	5.5	2.1
5.7	2.5	5	2
5.8	2.8	5.1	2.4
6.4	3.2	5.3	2.3
6.5	3	5.5	1.8
7.7	3.8	6.7	2.2
7.7	2.6	6.9	2.3
6	2.2	5	1.5
6.9	3.2	5.7	2.3
5.6	2.8	4.9	2
7.7	2.8	6.7	2

6.3	2.7	4.9	1.8
6.7	3.3	5.7	2.1
7.2	3.2	6	1.8
6.2	2.8	4.8	1.8
6.1	3	4.9	1.8
6.4	2.8	5.6	2.1
7.2	3	5.8	1.6
7.4	2.8	6.1	1.9
7.9	3.8	6.4	2
6.4	2.8	5.6	2.2
6.3	2.8	5.1	1.5
6.1	2.6	5.6	1.4
7.7	3	6.1	2.3
6.3	3.4	5.6	2.4
6.4	3.1	5.5	1.8
6	3	4.8	1.8
6.9	3.1	5.4	2.1
6.7	3.1	5.6	2.4
6.9	3.1	5.1	2.3
5.8	2.7	5.1	1.9
6.8	3.2	5.9	2.3
6.7	3.3	5.7	2.5
6.7	3	5.2	2.3
6.3	2.5	5	1.9
6.5	3	5.2	2
6.2	3.4	5.4	2.3
5.9	3	5.1	1.8

[View Code](#)

iris_id.txt

0
0
0
0
0
0
0
0
0
0
0
0
0

[illegible]

[illegible]

[illegible][View Code](#)

2.matlab源程序:

My_Kmeans.m

```
function label_1=My_Kmeans(K)
%输入K: 聚类数
%输出: label_1:聚的类, para_miu_new:聚类中心  $\mu$ 
format long
eps=1e-5; %定义迭代终止条件的eps
data=dlmread('E:\www.cnblogs.comkailugaji\data\iris\iris_data.txt');
%-----
%对data做最大-最小归一化处理
[data_num,~]=size(data);
X=(data-ones(data_num,1)*min(data))./(ones(data_num,1)*(max(data)-min(data)));
[X_num,~]=size(X);
%-----
%随机初始化K个聚类中心
rand_array=randperm(X_num); %产生1~X_num之间整数的随机排列
para_miu_new=X(rand_array(1:K),:); %随机排列取前K个数, 在X矩阵中取这K行作为初始聚类中心
responsivity=zeros(X_num,K);
%-----
%K-means算法
while true
    para_miu=para_miu_new; %上一步的聚类中心
    %欧氏距离, 计算  $(X-\text{para\_miu})^2=X^2+\text{para\_miu}^2-2*X*\text{para\_miu}'$ , 矩阵大小为X_num*K
    distant=repmat(sum(X.*X,2),1,K)+repmat(sum(para_miu.*para_miu,2)',X_num,1)-2*X*para_miu';
    %返回distant每行最小值所在的下标
    [~,label_1]=min(distant,[],2);
    %构建隶属度矩阵X_num*K
    for i=1:X_num
        for j=1:K
            responsivity(i,j)=isequal(j,label_1(i));
        end
    end
    R_k=sum(responsivity,1); %分母, 第k类的个数, 1*k的矩阵
    para_miu_new=diag(1./R_k)*responsivity'*X; %更新参数miu(聚类中心)
    if norm(para_miu_new-para_miu)<=eps
        break;
    end
end
end
```

Eg_Kmeans.m


```
function ave_acc_kmeans=Eg_Kmeans(K,max_iter)
%输入K:聚的类，max_iter是最大迭代次数
%输出ave_acc_kmeans: 迭代max_iter次之后的平均准确度
s=0;
for i=1:max_iter
    label_1=My_Kmeans(K);
    accuracy=succeed(K,label_1);
    s=s+accuracy;
end
ave_acc_kmeans=s/max_iter;
```

3.结果

```
>> ave_acc_kmeans=Eg_Kmeans(3,50)
ave_acc_kmeans =
    0.842533333333333
```