

# KFCM算法的matlab程序

在“[聚类——KFCM](#)”这篇文章中已经介绍了KFCM算法，现在用matlab程序对iris数据库进行简单的实现，并求其准确度。

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



[illegible]

[illegible]

[illegible]

## 2.matlab程序

**My\_KFCM.m**

```
function label_1=My_KFCM(K,sigma)
%输入K: 聚类数, sigma: 高斯核函数的参数
```

```

%输出: label_1:聚的类, para_miu_new:模糊聚类中心  $\mu$ , responsivity:模糊隶属度
format long
eps=1e-4; %定义迭代终止条件的eps
alpha=2; %模糊加权指数, [1,+无穷)
max_iter=100; %最大迭代次数
data=dlmread('E:\www.cnblogs.com\kailugaji\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,X_dim]=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=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);
R_up=zeros(X_num,K);
% -----
% KFCM算法
for t=1:max_iter
    responsivity_new=responsivity; %上一步的隶属度矩阵
    %欧氏距离, 计算  $(X-\text{para\_miu})^2=X^2+\text{para\_miu}^2-2*\text{para\_miu}*X'$ , 矩阵大小为X_num*K
    distant=(sum(X.*X,2))*ones(1,K)+ones(X_num,1)*(sum(para_miu.*para_miu,2))'-2*X*para_miu';
    %高斯核函数, X_num*K的矩阵
    kernel_fun=exp((-distant)/(2*sigma*sigma));
    %更新隶属度矩阵X_num*K
    for i=1:X_num
        for j=1:K
            if kernel_fun(i,j)==1
                responsivity_new(i,j)=1./sum(responsivity_new(i,:)==0);
            else
                R_up(i,j)=(1-kernel_fun(i,j)).^(-1/(alpha-1)); %隶属度矩阵的分子部分
                responsivity_new(i,j)= R_up(i,j)./sum(R_up(i,:),2);
            end
        end
    end
    %目标函数值
    fitness(t)=2*sum(sum((1-kernel_fun).*(responsivity.^alpha)));
    %更新聚类中心K*X_dim
    miu_up=((kernel_fun.*responsivity_new)'.^alpha)*X; % $\mu$  的分子部分
    para_miu=miu_up./((sum((kernel_fun.*responsivity_new).^alpha))'*ones(1,X_dim));
    if t>1
        %if abs(fitness(t)-fitness(t-1))<eps
        if norm(responsivity_new-responsivity)<=eps
            break;
        end
    end
end
end

```



```

end
%iter=t; %实际迭代次数
[~,label_1]=max(responsivity_new,[],2);

```

## succeed.m

```

function accuracy=succeed(K,id)
%输入K: 聚的类, id: 训练后的聚类结果, N*1的矩阵
N=size(id,1); %样本个数
p=perms(1:K); %全排列矩阵
p_col=size(p,1); %全排列的行数
new_label=zeros(N,p_col); %聚类结果的所有可能取值, N*p_col
num=zeros(1,p_col); %与真实聚类结果一样的个数
real_label=dlmread('E:\www.cnblogs.com\kailugaji\data\iris\iris_id.txt');
%将训练结果全排列为N*p_col的矩阵, 每一列为一种可能性
for i=1:N
    for j=1:p_col
        for k=1:K
            if id(i)==k
                new_label(i,j)=p(j,k)-1; %iris数据库, 0 1 2
            end
        end
    end
end
%与真实结果比对, 计算精确度
for j=1:p_col
    for i=1:N
        if new_label(i,j)==real_label(i)
            num(j)=num(j)+1;
        end
    end
end
accuracy=max(num)/N;

```

## Eg\_KFCM.m

```

function ave_acc_KFCM=Eg_KFCM(K,sigma,max_iter)
%输入K:聚的类, max_iter是最大迭代次数, sigma: 高斯核函数的参数
%输出ave_acc_KFCM: 迭代max_iter次之后的平均准确度
s=0;
for i=1:max_iter
    label_1=My_KFCM(K,sigma);
    accuracy=succeed(K,label_1);
    s=s+accuracy;
end

```

```
end  
ave_acc_KFCM=s/max_iter;
```

### 3.结果

```
>> ave_acc_KFCM=Eg_KFCM(3,150,50)  
  
ave_acc_KFCM =  
  
    0.8933333333333333
```