# 聚类——KFCM的matlab程序

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

在聚类——KFCM文章中已介绍了KFCM-F算法的理论知识，现在用matlab进行实现,下面这个例子是用FCM初始化聚类中心，也可以随机初始化聚类中心。

## 1.matlab程序

### KFCM_main.m

```
%function [ave_acc_KFCM,max_acc_KFCM,min_acc_KFCM,ave_iter_KFCM,ave_run_time]=KFCM_main(X,real_label,K)
function [ave_acc_KFCM,max_acc_KFCM,min_acc_KFCM,ave_iter_FCM,ave_iter_KFCM,ave_run_time]=KFCM_main(X,real_label,K)
%输入K:聚的类，real_label：真实的标签，X：数据集
%输出ave_acc_KFCM：迭代max_iter次之后的平均准确度,iter:实际KFCM迭代次数
t0=cputime;
max_iter=20;
s_1=0;
s_2=0;
s_3=0;
accuracy=zeros(max_iter,1);
iter_KFCM_t=zeros(max_iter,1);
iter_FCM_t=zeros(max_iter,1);
%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)));
for i=1:max_iter
    %[label,iter_KFCM]=My_KFCM(X,K);
    [label,iter_KFCM,~,iter_FCM]=My_KFCM(X,K);
    iter_KFCM_t(i)=iter_KFCM;
    iter_FCM_t(i)=iter_FCM;
    accuracy(i)=succeed(real_label,K,label);
    s_1=s_1+accuracy(i);
    s_2=s_2+iter_KFCM_t(i);
    s_3=s_3+iter_FCM_t(i);
    %fprintf('第 %2d 次，KFCM的迭代次数为：%2d，准确度为：%.8f\n', i, iter_KFCM_t(i), accuracy(i));
    fprintf('第 %2d 次，FCM的迭代次数为：%2d，KFCM的迭代次数为：%2d，准确度为：%.8f\n', i,  iter_FCM_t(i),  iter_KFCM_t(i), accuracy(i));
end
ave_iter_FCM=s_3/max_iter;
ave_iter_KFCM=s_2/max_iter;
ave_acc_KFCM=s_1/max_iter;
```

```
max_acc_KFCM=max(accuracy);
min_acc_KFCM=min(accuracy);
run_time=cputime-t0;
ave_run_time=run_time/max_iter;
```

## My_KFCM.m

```
%function  [label, iter_KFCM, para_miu]=My_KFCM(X,K)
function  [label, iter_KFCM, para_miu,iter_FCM]=My_KFCM(X,K)
%输入K：聚类数，X：数据集
%输出：label:聚的类, para_miu:模糊聚类中心μ，iter_KFCM：KFCM迭代次数
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)
T=100;  %最大迭代次数
%sigma_2=2^(-4);  %高斯核函数的参数sigma^2
sigma_2=150;  %高斯核函数的参数sigma^2
[X_num,X_dim]=size(X);
fitness=zeros(X_num,1);  %目标函数
responsivity=zeros(X_num,K);  %隶属函数
R_up=zeros(X_num,K);  %隶属函数的分子部分
count=zeros(X_num,1);  %统计distant中每一行为0的个数
%随机初始化K个聚类中心
% [X_num,~]=size(X);
% 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
%用FCM初始聚类中心
[~,para_miu,iter_FCM]=My_FCM(X,K);
% KFCM算法
for t=1:T
    %欧氏距离，计算（X-para_miu）^2=X^2+para_miu^2-2*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)./(sigma_2));
    %更新隶属度矩阵X_num*K
    for i=1:X_num
        count(i)=sum(kernel_fun(i,:)==1);
        if count(i)>0
            for k=1:K
                if kernel_fun(i,k)==1
                    responsivity(i,k)=1./count(i);
                else
                    responsivity(i,k)=0;
                end
            end
        else
```

```
            R_up(i,:)=(1-kernel_fun(i,:)).^(-1/(alpha-1));  %隶属度矩阵的分子部分
            responsivity(i,:)= R_up(i,:)./sum( R_up(i,:),2);
        end
    end
    %目标函数值
    fitness(t)=2*sum(sum((ones(X_num,K)-kernel_fun).*(responsivity.^(alpha))));
     %更新聚类中心K*X_dim
    miu_up=(kernel_fun.*(responsivity.^(alpha)))'*X;  %μ的分子部分
    para_miu=miu_up./(sum(kernel_fun.*(responsivity.^(alpha)))'*ones(1,X_dim));
    if t>1
        if abs(fitness(t)-fitness(t-1))<eps
            break;
        end
    end
end
iter_KFCM=t;  %实际迭代次数
[~,label]=max(responsivity,[],2);
```

# 2.在UCI数据库的iris上的运行结果

```
>> [ave_acc_KFCM,max_acc_KFCM,min_acc_KFCM,ave_iter_FCM,ave_iter_KFCM,ave_run_time]=KFCM_main(data,real_label,3)
第  1 次，FCM的迭代次数为：12，KFCM的迭代次数为：  2，准确度为：0.89333333
第  2 次，FCM的迭代次数为：12，KFCM的迭代次数为：  2，准确度为：0.89333333
第  3 次，FCM的迭代次数为：18，KFCM的迭代次数为：  2，准确度为：0.89333333
第  4 次，FCM的迭代次数为：12，KFCM的迭代次数为：  2，准确度为：0.89333333
第  5 次，FCM的迭代次数为：14，KFCM的迭代次数为：  2，准确度为：0.89333333
第  6 次，FCM的迭代次数为：27，KFCM的迭代次数为：  2，准确度为：0.89333333
第  7 次，FCM的迭代次数为：15，KFCM的迭代次数为：  2，准确度为：0.89333333
第  8 次，FCM的迭代次数为：20，KFCM的迭代次数为：  2，准确度为：0.89333333
第  9 次，FCM的迭代次数为：13，KFCM的迭代次数为：  2，准确度为：0.89333333
第 10 次，FCM的迭代次数为：16，KFCM的迭代次数为：  2，准确度为：0.89333333
第 11 次，FCM的迭代次数为：15，KFCM的迭代次数为：  2，准确度为：0.89333333
第 12 次，FCM的迭代次数为：10，KFCM的迭代次数为：  2，准确度为：0.89333333
第 13 次，FCM的迭代次数为：24，KFCM的迭代次数为：  2，准确度为：0.89333333
第 14 次，FCM的迭代次数为：19，KFCM的迭代次数为：  2，准确度为：0.89333333
第 15 次，FCM的迭代次数为：10，KFCM的迭代次数为：  2，准确度为：0.89333333
第 16 次，FCM的迭代次数为：16，KFCM的迭代次数为：  2，准确度为：0.89333333
第 17 次，FCM的迭代次数为：15，KFCM的迭代次数为：  2，准确度为：0.89333333
第 18 次，FCM的迭代次数为：27，KFCM的迭代次数为：  2，准确度为：0.89333333
第 19 次，FCM的迭代次数为：15，KFCM的迭代次数为：  2，准确度为：0.89333333
第 20 次，FCM的迭代次数为：12，KFCM的迭代次数为：  2，准确度为：0.89333333

ave_acc_KFCM =
   0.893333333333333

max_acc_KFCM =
```

```
   0.893333333333333

min_acc_KFCM =
   0.893333333333333

ave_iter_FCM =
  16.100000000000001

ave_iter_KFCM =
     2

ave_run_time =
   0.028125000000000
```