

MATLAB程序：用FCM分割脑图像

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

脑图像基础知识请看：[脑图像](#)；FCM算法介绍请看：[聚类——FCM](#)；数据来源：[BrainWeb: Simulated Brain Database](#)，只选取脑图像中的0、1、2、3类，其余类别设为0。本文用到的数据：[Simulated Brain Database](#)

1. MATLAB程序

FCM_image_main.m

```
function [accuracy, iter_FCM, run_time]=FCM_image_main(filename, num, K)
%num: 第几层, K: 聚类数
%[accuracy, iter_FCM, run_time]=FCM_image_main('t1_icbm_normal_1mm_pn0_rf0.rawb', 100, 4)
[data_load, label_load]=main(filename, num); %原图像
[m, n]=size(data_load);
X=reshape(data_load, m*n, 1); % (m*n)*1
real_label=reshape(label_load, m*n, 1)+ones(m*n, 1);
Ground_truth(num, K); %标准分割结果, 进行渲染
t0=cputime;
[label_1, ~, iter_FCM]=My_FCM(X, K);
[label_new, accuracy]=succeed(real_label, K, label_1);
run_time=cputime-t0;
label_2=reshape(label_new, m, n);
rendering_image(label_2, K); %聚类结果
```

main.m

```
function [read_new, mark]=main(filename, num)
%将真实脑图像中的0、1、2、3拿出来, 其余像素为0.
%函数main(filename, num)中的第一个参数filename是欲读取的rawb文件的文件名, 第二个参数num就是第多少张。
%例如: main('t1_icbm_normal_1mm_pn0_rf0.rawb', 100)
mark=Mark('phantom_1.0mm_normal_crisp.rawb', num);
read=readrawb(filename, num);
[row, col]=size(read);
read_new=zeros(row, col);
for i=1:row %行
    for j=1:col %列
        if mark(i, j)==0
```

```

        read_new(i, j)=0;
    else
        read_new(i, j)=read(i, j);    %将第0、1、2、3类拿出来，其余类为0
    end
end
end
%旋转90° 并显示出来
figure(1);
init_image=imrotate(read_new, 90);
imshow(uint8(init_image));
title('原图像');

```

Mark.m

```

function mark=Mark(filename,num)
%将标签为1、2、3类分出来，其余为0，mark取值：0、1、2、3
%[mark_new,mark]=Mark('phantom_1.0mm_normal_crisp.rawb',90);
fp=fopen(filename);
temp=fread(fp, 181 * 217 * 181);
image=reshape(temp, 181 * 217, 181);
images=image(:, num);
images=reshape(images, 181, 217);
mark_data=images;
fclose(fp);
mark=zeros(181,217);
%将第0、1、2、3类标签所在的坐标点拿出来，其余置0
for i=1:181
    for j=1:217
        if (mark_data(i, j)==1) || (mark_data(i, j)==2) || (mark_data(i, j)==3)
            mark(i, j)=mark_data(i, j);
        else
            mark(i, j)=0;
        end
    end
end
end

```

readrawb.m

```

function g = readrawb(filename, num)
% 函数readrawb(filename, num)中的第一个参数filename是欲读取的rawb文件的文件名，第二个参数num就是第多少张。
fid = fopen(filename);
% 连续读取181*217*181个数据，这时候temp是一个长度为181*217*181的向量。
% 先将rawb中的所有数据传递给temp数组，然后将tempreshape成图片集。
temp = fread(fid, 181 * 217 * 181);
% 所以把它变成了一个181*217行，181列的数组，按照它的代码，这就是181张图片的数据，每一列对应一张图。

```

```
% 生成图片集数组。图片集images数组中每一列表示一张图片。
images = reshape(temp, 181 * 217, 181);
% 读取数组中的第num行，得到数组再reshape成图片原来的行数和列数：181*217。
image = images(:, num);
image = reshape(image, 181, 217);
g = image;
fclose(fid);
end
```

Ground_truth.m

```
function Ground_truth(num, K)
%标准分割结果
%Ground_truth(100, 4)
mark=Mark('phantom_1.0mm_normal_crisp.rawb', num); %0、1、2、3
m=181;
n=217;
read_new=zeros(m,n);
mark=mark+ones(m, n); %标签：1、2、3、4
for i=1:m %行
    for j=1:n %列
        for k=1:K
            if mark(i, j)==k
                read_new(i, j)=floor(255/K)*(k-1);
            end
        end
    end
end
end
% 旋转90° 并显示出来
figure(2)
truth_image=imrotate(read_new, 90);
imshow(uint8(truth_image));
title('标准分割结果');
```

My_FCM.m

```
function [label_l, para_miu_new, iter]=My_FCM(data, K)
%输入K：聚类数
%输出：label_l:聚的类， para_miu_new:模糊聚类中心  $\mu$ ，responsivity:模糊隶属度
format long
eps=1e-8; %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
fitness=zeros(T, 1);
[data_num, ~]=size(data);
```

```

count=zeros(data_num,1); %统计distant中每一行为0的个数
responsivity=zeros(data_num,K);
R_up=zeros(data_num,K);
%-----
%对data做最大-最小归一化处理
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
% -----
% FCM算法
for t=1:T
    %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';
    %更新隶属度矩阵X_num*K
    for i=1:X_num
        count(i)=sum(distant(i,:)==0);
        if count(i)>0
            for k=1:K
                if distant(i,k)==0
                    responsivity(i,k)=1./count(i);
                else
                    responsivity(i,k)=0;
                end
            end
        else
            R_up(i,:)=distant(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)=sum(sum(distant.*(responsivity.^(alpha))));
    %更新聚类中心K*X_dim
    miu_up=(responsivity'.^(alpha))*X; %μ 的分子部分
    para_miu=miu_up./((sum(responsivity.^(alpha)))'*ones(1,X_dim));
    if t>1
        if abs(fitness(t)-fitness(t-1))<eps
            break;
        end
    end
end
para_miu_new=para_miu;
iter=t; %实际迭代次数
[~,label_1]=max(responsivity,[],2);

```

succeed.m

```
function [label_new, accuracy]=succeed(real_label, 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
%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); %iris数据库, 1 2 3
            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
[M, I]=max(num);
accuracy=M/N;
label_new=new_label(:, I);
```

rendering_image.m

```
function rendering_image(label, K)
%对分割结果进行渲染, 4类, label:1、2、3、4
[m, n]=size(label);
read_new=zeros(m, n);
for i=1:m %行
    for j=1:n %列
        for k=1:K
            if label(i, j)==k
                read_new(i, j)=floor(255/K)*(k-1);
            end
        end
    end
end
```

```
end
% 旋转90° 并显示出来
figure(3);
cluster_image=imrotate(read_new, 90);
imshow(uint8(cluster_image));
title('分割后');
```

2. 实验及结果

对T1模态、icmb协议下，切片厚度为1mm，噪声水平为7%，灰度不均匀水平为40%的第90层脑图像进行分割。因为FCM随机初始化，所以聚类结果会有偏差，结果受初始化影响比较大。

```
>> [accuracy, iter_FCM, run_time]=FCM_image_main('t1_icbm_normal_1mm_pn7_rf40.rawb', 90, 4)
```

```
accuracy =
```

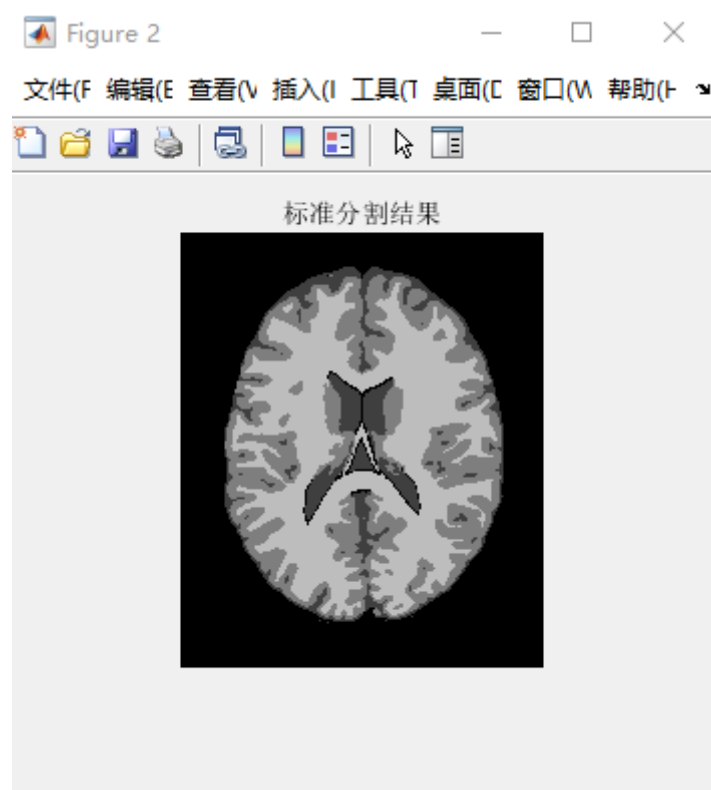
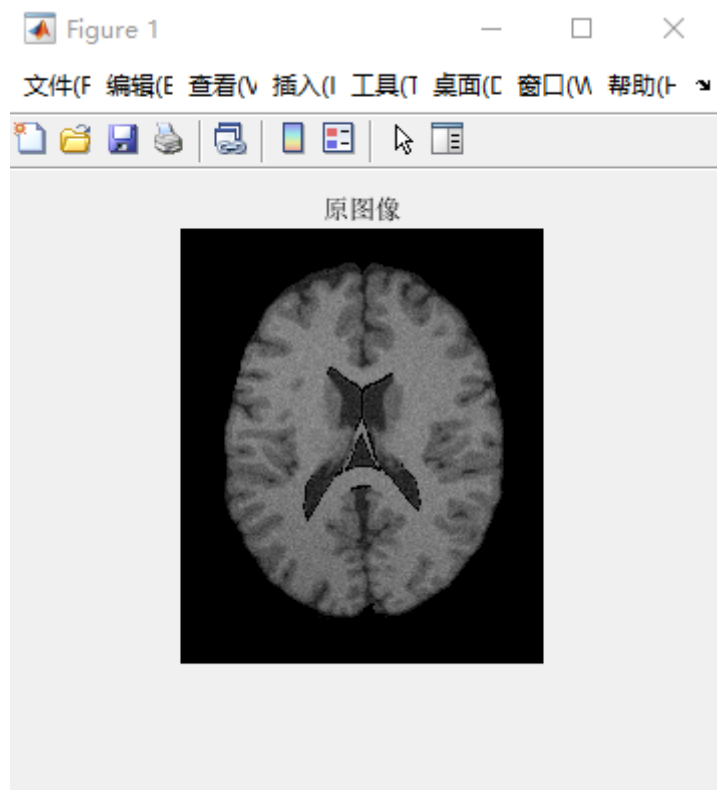
```
0.943783893881916
```

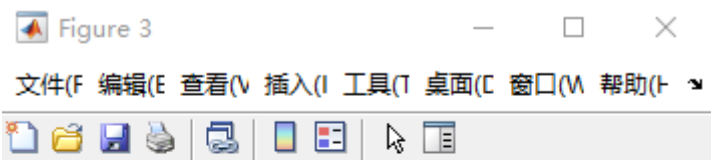
```
iter_FCM =
```

```
25
```

```
run_time =
```

```
1.9375000000000000
```





分割后

