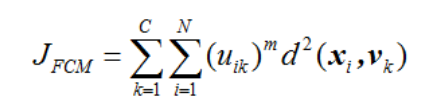
实验9 模糊聚类实验

姓名：唐川淇 学号：1131190111

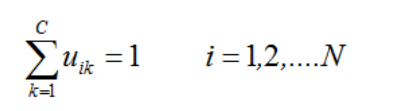
# 问题重述

两数据集合iris和wine，任选其一编程实现FCM算法，并与matlab工具箱中fcm函数的分类效果进行比较。

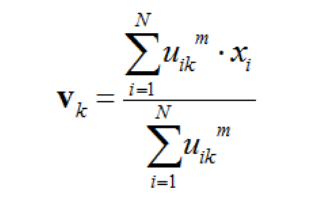
其中FCM算法的目标函数定义如下：



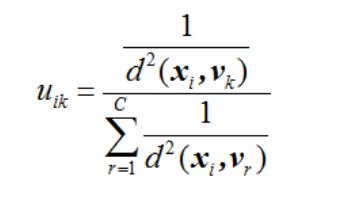
其中隶属度的约束条件为：



聚类中心和的更新公式：



隶属度的更新公式：



# 模型的建立与求解

## 参数设定

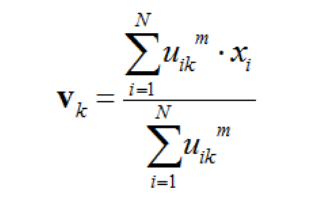
首先根据题目要求设定好聚类要使用的参数，首先从txt文件中读取数据，这里选择iris数据集作为实验对象。预先读取数据，存入变量data，其中行数代表数据的个数，列数代表每一变量的维度。设定聚类最终的分类个数K=4，最大的迭代次数maxgen=100，其中公式中的m设定为2，以及初始的隶属度函数矩阵，这里使用随机函数rand()，随机的生成一个隶属度矩阵，其中每一个数据的隶属度之和为1。设定一个阈值threshold，当迭代中两次的目标函数的差小于阈值时自动跳出迭代。参数设定部分的代码如下：

|  |
| --- |
| 代码 |
| 介绍：参数设定 |
| [a1,a2,a3,a4]=textread('iris.txt','%f%f%f%f');  data=[a1,a2,a3,a4];  K = 4; % 分类个数  maxgen = 100; % 最大迭代次数  m = 2; % 指数的次幂  threshold = 1e-6; % 阈值  [data\_n, in\_n] = size(data); % 行数，即样本个数/列数，即样本维数  % 初始化隶属度矩阵  U = rand(K, data\_n);  col\_sum = sum(U);  U = U./col\_sum(ones(K, 1), :); |

## 迭代聚类

### 更新聚类中心

首先更新聚类中心，根据题目，聚类中心的更新公式如下：

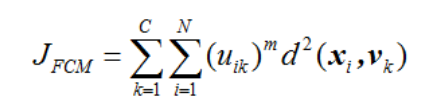


Vk代表第k个中心的坐标，公式分数线上方为隶属度的m次方与数据的乘积之和，分数线下方为隶属度的的m次方之和，隶属度下相当于一个系数，如果隶属度越大说明重要程度越大，该中心也越靠近该点。更新聚类中的函数如下：

|  |
| --- |
| 代码 |
| 介绍：更新聚类中心 |
| mf = U.^m;  center = mf\*data./((ones(in\_n, 1)\*sum(mf'))'); |

### 更新目标函数值

根据公式更新目标函数值，具体公式如下：

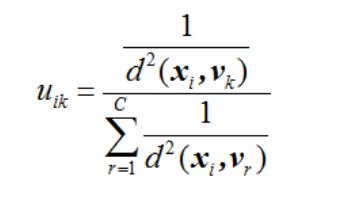


目标函数为每个样本到中心的隶属度的m次方与距离的平方的乘积之和，目标函数反映了整体的聚类效果，如果聚类效果越好，那么整体的函数值也就越小，所以迭代也是在不断减小目标函数的数值。更新目标函数的代码如下：

|  |
| --- |
| 代码 |
| 介绍：更新目标函数 |
| dist = zeros(size(center, 1), data\_n);  for k = 1:size(center, 1)  dist(k, :) = sqrt(sum(((data-ones(data\_n, 1)\*center(k, :)).^2)', 1));  end  J(i) = sum(sum((dist.^2).\*mf)); |

### 更新隶属度函数

接下来更新各数据的隶属度，隶属度更新的函数如下：



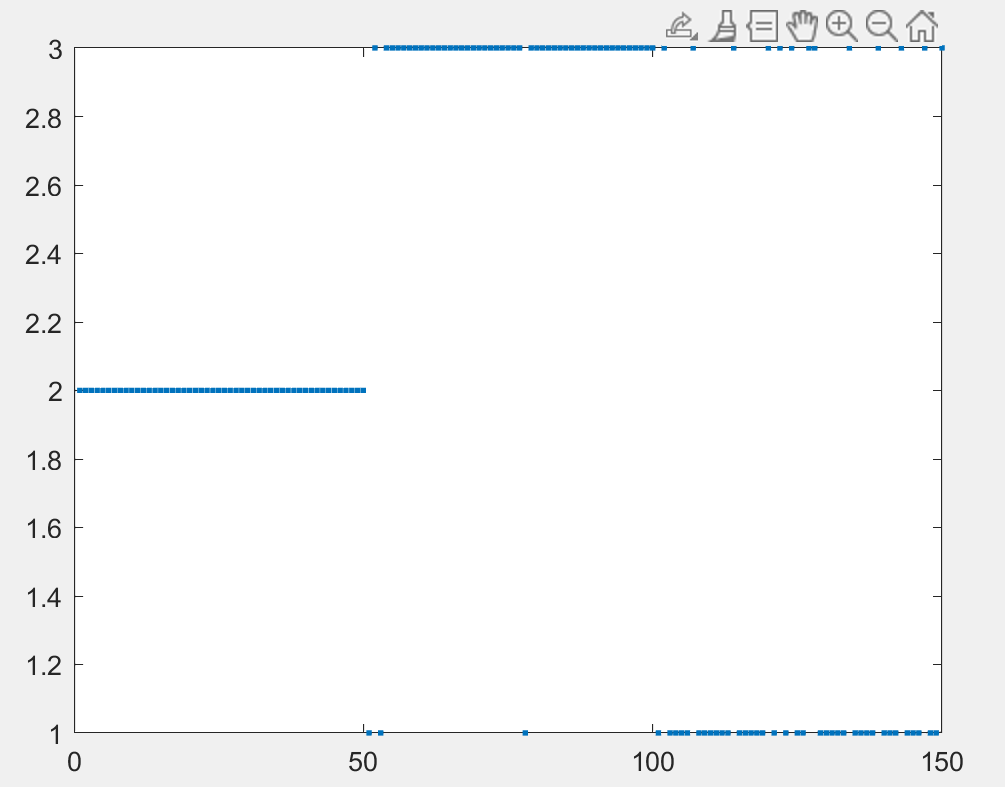
简单来看，分子是到某一中心的距离平方的倒数，分母是到左右中心的距离平方的倒数之和。整体来看，如果到某一中心的距离越小，那么就属于该中心的隶属度越高，同时该隶属度函数保证了同一样本到不同中心的隶属度之和为1。

更新隶属度的代码如下：

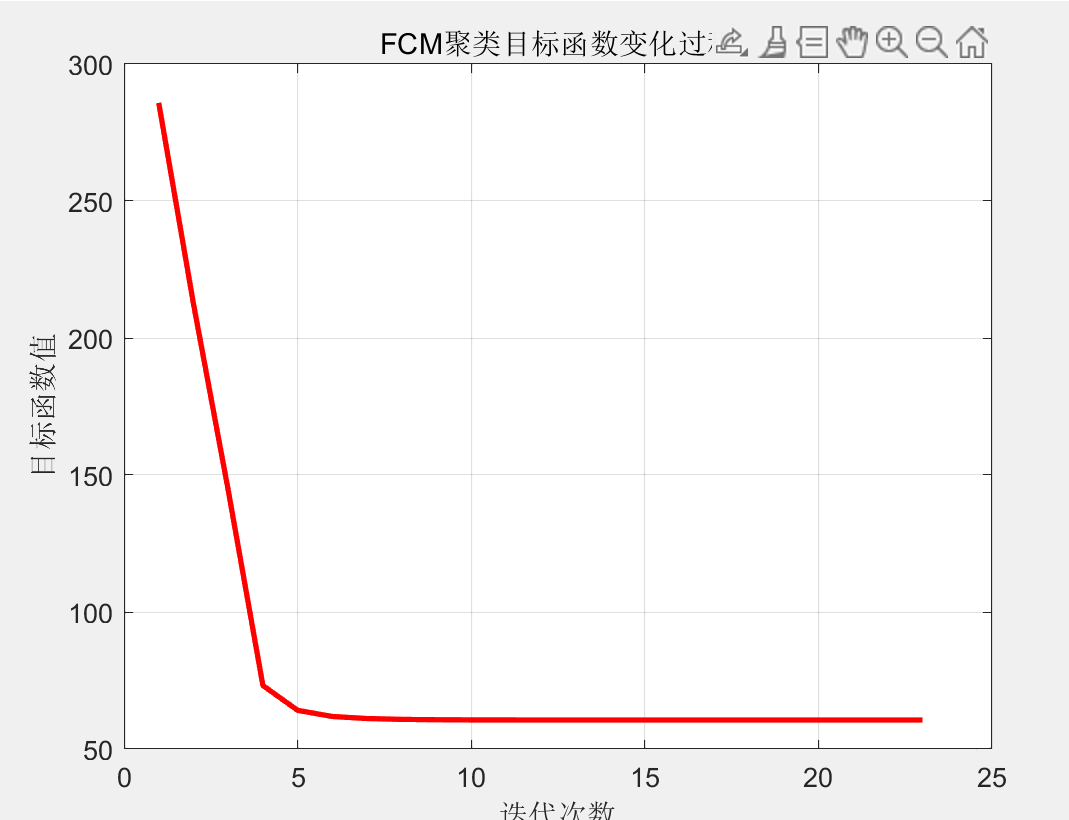
|  |
| --- |
| 代码 |
| 介绍：更新隶属度 |
| tmp = dist.^(-2);  U = tmp./(ones(K, 1)\*sum(tmp)); |

## 结果展示

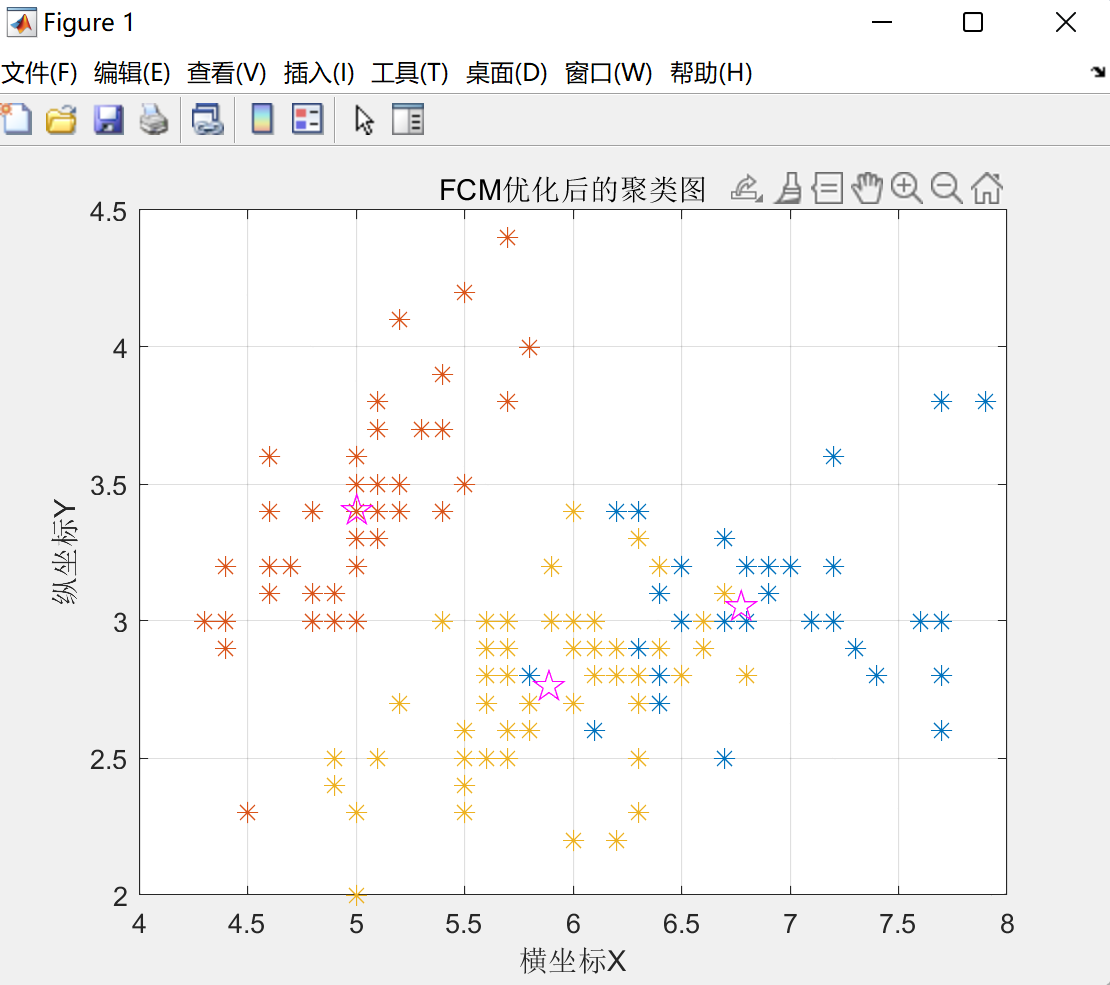
鸢尾花数据集设定类别为3，每一个鸢尾花最终属于的类别展示图如下：



其中目标函数的变化曲线如下：



由于数据的维数大于2，因此并不能直接在二维平面上展示结果，所以选择数据的前两维，画出最终的聚类结果如下：



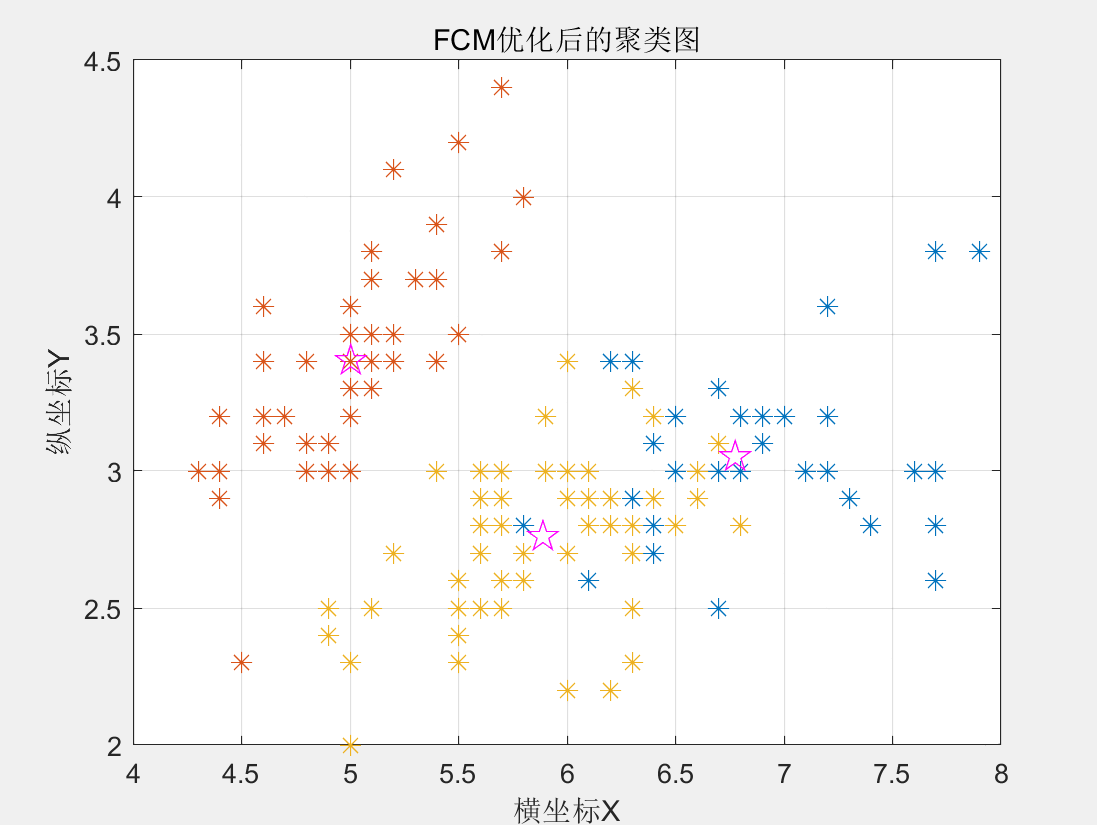
从上图可以看出，从前两个维度的角度看，数据分类的效果较好，同一颜色的数据大概都分在了同一类别中。

# 模型与MATLAB自带工具箱的比较

使用工具箱进行聚类分析，其中代码如下：

|  |
| --- |
| 代码 |
| 介绍：fcm |
| [a1,a2,a3,a4]=textread('iris.txt','%f%f%f%f');  data=[a1,a2,a3,a4];  % 初始化隶属度矩阵  [centers,U] = fcm(data,3); |

和之前相同可以选取前两个维度，画出数据属于的类别，同一类别颜色相同，五角星为每一类别的中心，画出的图像如下：



与之前得到的聚类结果做比较，发现两者聚类结果完全相同。

附录

|  |
| --- |
| 代码 |
| 介绍：FCM聚类完整代码 |
| clear;  close all;  clc;  %% 初始化参数  [a1,a2,a3,a4]=textread('iris.txt','%f%f%f%f');  data=[a1,a2,a3,a4];  K = 3; % 分类个数  maxgen = 100; % 最大迭代次数  m = 2; % 指数的次幂  threshold = 1e-6; % 阈值  [data\_n, in\_n] = size(data); % 行数，即样本个数/列数，即样本维数  % 初始化隶属度矩阵  U = rand(K, data\_n);  col\_sum = sum(U);  U = U./col\_sum(ones(K, 1), :);    %% 迭代  for i = 1:maxgen  % 更新聚类中心  mf = U.^m;  center = mf\*data./((ones(in\_n, 1)\*sum(mf'))');  % 更新目标函数值  dist = zeros(size(center, 1), data\_n);  for k = 1:size(center, 1)  dist(k, :) = sqrt(sum(((data-ones(data\_n, 1)\*center(k, :)).^2)', 1));  end  J(i) = sum(sum((dist.^2).\*mf));  % 更新隶属度矩阵  tmp = dist.^(-2);  U = tmp./(ones(K, 1)\*sum(tmp));  % 终止条件判断  if i > 1  if abs(J(i) - J(i-1)) < threshold  break;  end  end  end    %% 绘图  [max\_vluae, index] = max(U);  index = index';  figure;  for i = 1:K  col = find(index == i); % max(U)返回隶属度列最大值所在行一致的分为一类  plot(data(col, 1), data(col, 2), '\*', 'MarkerSize', 8);  hold on  end  grid on  % 画出聚类中心  plot(center(:, 1), center(:, 2), 'p', 'color', 'm', 'MarkerSize', 12);  xlabel '横坐标X'; ylabel '纵坐标Y';  title 'FCM优化后的聚类图';  % 目标函数变化过程  figure;  plot(J, 'r', 'linewidth', 2);  xlabel '迭代次数'; ylabel '目标函数值';  title 'FCM聚类目标函数变化过程';  grid on |