<http://www.voidcn.com/blog/theonegis/article/p-5716999.html>

function [ IDX, C ] = kmeans( X, k )

% KMEANS K-Means聚类算法

% Author: 谭振宇

% Data: 2016.03.24

% Input:

% X: n\*m的矩阵，n表示点的个数，m表示点的维数

% k: 聚类的个数

% Output:

% IDX: n\*1的向量，指示每个点所在聚类中心的索引

% C: n\*k的矩阵，聚类中心

n = size(X, 1); % 点的个数

m = size(X, 2); % 点的维数

% 1.从nums个向量中选择k个向量作为质心

M = X(1:k, :); % 选取前k行为初始聚类中心

loop = 0;

while true

% 2. 对剩余的每个向量测量其到每个质心的距离，并把它归到最近的质心的类

DIST = zeros(n, k); % DIST为每个点到聚类中心的距离

for i = 1:n

for j = 1:k

DIST(i, j) = norm(X(i, :) - M(j, :)); % 计算每个点到聚类中心的聚类

end

end

[~, IDX] = min(DIST, [], 2); % IDX为每个点到聚类中心最小距离的索引，表征的是每个点隶属哪一类

% 3. 重新计算已经得到的各个类的质心

C = zeros(k, m); % C为重新计算以后的聚类中心

count = zeros(k, 1); % count统计各个类别中点的个数

for i = 1:n

idx = IDX(i); % idx为隶属的类

count(idx) = count(idx) + 1; % 统计每个类别的个数

C(idx, :) = C(idx, :) + X(i, :);

end

for i = 1:k

C(i, :) = C(i, :) / count(i); % 这里C计算出来是新的质心

end

loop = loop + 1;

disp(['第' , num2str(loop) , '次迭代'])

% 4. 迭代2~3步直至新的质心与原质心相等或小于指定阈值，算法结束

residual = norm(M - C); % 计算新质心和原始质心的距离

disp(['新的质心与原质心距离为：' , num2str(residual)])

if residual < 1e-5

break;

end

M = C; % 如果迭代继续，则将新质心赋值给原来的质心，继续循环

end

end

测试主函数：

clear

clc

% 生成测试数据

rng('default');

X = [gallery('uniformdata',[10 3],12);

gallery('uniformdata',[10 3],13)+1.2;

gallery('uniformdata',[10 3],14)+2.5];

% 原始数据绘图显示

figure

scatter3(X(:,1), X(:,2), X(:,3))

title('原始数据')

k = 3; % 聚类的个数设置为3

[idx, C] = kmeans(X, k);

num = size(X, 1); % 点的个数

COLOR = zeros(num, 3); % 根据类别设置的颜色矩阵

for i = 1:num

COLOR(i, idx(i)) = 1;

end

% 进行聚类结果的显示，不同的类别用不同的颜色显示

figure

scatter3(X(:,1), X(:,2), X(:,3), [], COLOR)

hold on

% 绘制聚类中心

scatter3(C(:,1), C(:,2), C(:,3), 'm', '\*')

title('聚类结果')