K-均值（K-Means）

K-均值聚类算法是著名的划分聚类算法。划分的基本思想是：给定一个有 N 个元组或者记录的数据，分裂法将构造 K 个分组，每个分组就代表一个聚类，且 K<N。而且这 K 个分组满足下列条件：①每一个分组至少包含一个数据记录②每一个数据记录属于且仅属于一个分组。

K-均值原理

K-均值的工作原理：首先随机从数据集中选取 K 个点，每个点初始地代表每个簇的聚类中心，然后计算剩余各个样本到聚类中心的距离，将他赋给最近的簇，接着重新计算每一簇的平均值，整个过程不断重复，如果相邻两次调整没有明显变化，说明数据聚类形成的簇已经收敛。

本算法的一个特点就是在每次迭代中都要考察每个样本的分类是否正确。若不正确就要调整，在全部样本调整完后，再修改聚类中心，进入下一次迭代。这个过程将不断重复直到满足终止条件：（任选其一）

①没有对象被重新分配给不同的聚类
②聚类中心不再发生变化
③误差平方和局部最小

K-均值算法步骤

1、 从 N 个数据对象任意选择 K 个对象作为初始聚类中心
2、 循环 3 4 直到每个聚类不再发生变化为止
3、 根据每个聚类对象的均值（中心对象），计算每个对象与这些中心对象的距离，并根据最小距离重新对相应对象进行划分。
4、 重新计算每个聚类的均值（中心对象），直到聚类中心不再变化。这种划分使得下式最小

$$E = \sum_{j=1}^{k} \sum_{x_i \in \omega_j} \| x_i - m_j \|^2$$

K-Means 算法特点

1、 在 K-均值算法中 K 是事先给定的，而这个 K 值的选定是非常难以估计的
2、 在 K-均值算法中，首先需要根据初始聚类中心来确定一个初始化分，然后对初始划分进行优化
3、 K-均值算法需要不断地进行样本分类调整，不断地计算调整后的新的聚类中心，因此当数据量非常大时，算法的时间开销也是非常大的。
4、 K-均值算法对一些离散点和初始 K 值敏感，不同的距离初始值对同样的数据样本可能得到不同的结果。

优化的 K-均值算法

由于 K-均值算法结果与初始点的选定有较大联系，因此，我们需要对该算法进行改进。改进后的选取流程如下：

①在数据集中随机选取一个样本点作为第一个簇中心 C1

②计算剩余样本点与所有簇中心的最短距离，则某样本点被选为下一个簇中心的概率为：

$$\frac{(\min\{D(x_i)\})^2}{\sum D(x_j)\text{^}2}$$

这是因为，初始点的选择距离越大越好。

（注意：分母为所有元素点到当前簇点的距离的平方和）

K-均值实际应用与 Matlab 程序设计

已知有 20 个样本，每个样本有 2 个特征，数据分布如表 3－5 所列，试对这些数据进行分类。

表 3－5　数据

特　征	样　本									
X1	0	1	0	1	2	1	2	3	6	7
X2	0	0	1	1	1	2	2	2	6	6
X1	8	6	7	8	9	7	8	9	8	9
X2	6	7	7	7	7	8	8	8	9	9

Matlab 程序：

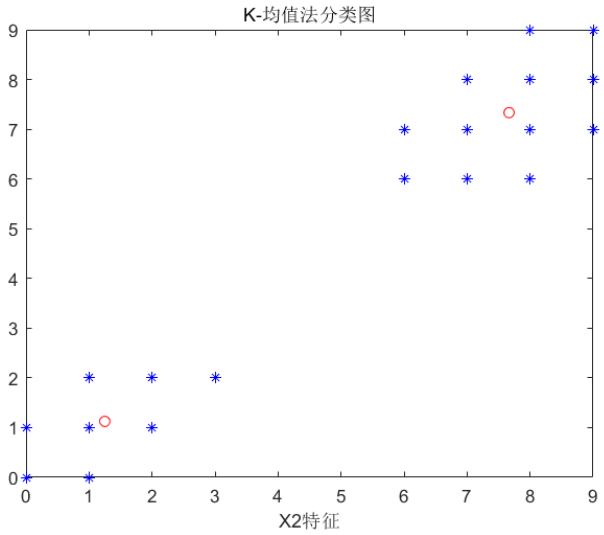
```
clc;
clear all;
%读取数据


X=xlsread('E:\顾子涵专用文件夹\学习\matlab 学习\matlab 与数学模型\K-均值算法数据.xlsx',1,'B2:U3');
X=X';
[a,b]=size(X);%计算矩阵的 size
K=2;%分类的个数
%选出两个簇
num(1)=ceil(unifrnd(0.5,a-0.5));
for j=2:K
    for i=1:a
        %dis(i,2)=distance(X(num(j-1),:),X(i,:));
        for m=1:j-1
            dis(i,2)=distance(X(num(m),:),X(i,:));
```

```
        end
        dis(i,1)=i;
        %说明:dis 的第一列为点的序号，第二列为点到
        %     第一个随机选定的簇的距离
    end

    for i=1:a
        dis(i,3)=dis(i,2)/sum(dis(:,2));
        %第三列为被选定为簇点的概率
        dis(i,4)=sum(dis(:,3));
        %第四列为累计概率
    end
    %轮盘法进行选点
    rnd=unifrnd(0,1);
    if rnd<dis(1,4)&&rnd>=0
        num(2)=1;
    else
        for i=2:a
            if rnd>=dis(i-1,4)&&rnd<dis(i,4)
                num(j)=i;
            end
        end
    end

    %dis
    num=num';
end
z=[X(num(1),:);X(num(2),:)];%选出的簇点
z1=zeros(2,2);%临时簇点
%下面开始分类操作
%计算所有样本点与簇中心的距离
while 1
    count=zeros(2,1);
    allsum=zeros(2,2);
    for i=1:a
        temp1=distance(X(i,:),X(num(1),:));
        temp2=distance(X(i,:),X(num(2),:));
        if temp1<temp2
            %划入最近的点中
            count(1)=count(1)+1;
            allsum(1,1)=allsum(1,1)+X(i,1);
            allsum(1,2)=allsum(1,2)+X(i,2);
        else
            count(2)=count(2)+1;
```

```
            allsum(2,1)=allsum(2,1)+X(i,1);
            allsum(2,2)=allsum(2,2)+X(i,2);
        end
    end
    %更新簇点：单行坐标均值
    z1(1,1)=allsum(1,1)/count(1);
    z1(1,2)=allsum(1,2)/count(1);
    z1(2,1)=allsum(2,1)/count(2);
    z1(2,2)=allsum(2,2)/count(2);
    if z==z1
        break;
    else
        z=z1;
    end
end


%显示结果
disp(z1);
plot(X(:,1),X(:,2),'b*');
hold on;
plot(z1(:,1),z1(:,2),'ro');
title('K-均值法分类图');
xlabel('X1 特征');
xlabel('X2 特征');
```

```
function d=distance(point1,point2)
    d=sqrt((point1(1)-point2(1))^2+(point1(2)-point2(2))^2);
end
```

分类结果:



簇点坐标:

```
    1.2500    1.1250
    7.6667    7.3333
```