信息网络专题研究课应用层实验报告

一、资料信息：

1.作者：trekhleb

2.资料题目：MatLab/Octave examples of popular machine learning algorithms with code examples and mathematics being explained

3.资料来源：https://github.com/trekhleb/machine-learning-octave

二、问题意义：

1.研究问题：在 MatLab 中的机器学习，主要研究了无监督学习中聚类问题的 K-means 算法。

2.意义：可应用于市场细分，社会网络分析，组织计算集群，天文数据分析，图像压缩等。

三、思路方法：

1.研究原理：无监督学习是机器学习的一个分支，它从没有标记、分类或分类的测试数据中学习。非监督学习不是对反馈做出响应，而是识别数据中的共性，并根据每个新数据中是否存在这些共性做出反应。在聚类问题中，用未知的特征来分割训练样本。K-means 聚类的目的是将 n 个观测值划分为 K 个聚类，每个观测值都属于最接近均值的聚类，作为聚类的原型。

2.研究步骤：

（1）给定一组训练数据集，中心点个数和迭代次数；

（2）基于训练集随机选取一定量中心点；

（3）根据到中心点的距离，将训练样本分割成多个聚类；

（4）基于前一部分中得到的最近的中心，计算每个簇的下一个平均中心点；

（5）完成要求的迭代次数得到最优中心点位置并绘制聚类图像。

四、实验结论：

1.实验代码：

Demo.m %演示程序

```
clear; close all; clc;
fprintf('Loading the data set #1...\n');
load('set1.mat');
fprintf('Plotting the data set #1...\n');
subplot(2, 2, 1);
plot(X(:, 1), X(:, 2), 'k+','LineWidth', 1, 'MarkerSize', 7);
title('Training Set #1'); %加载并绘制训练数据集 1，黑色+表示
fprintf('Training K-Means for data set #1...\n');
```

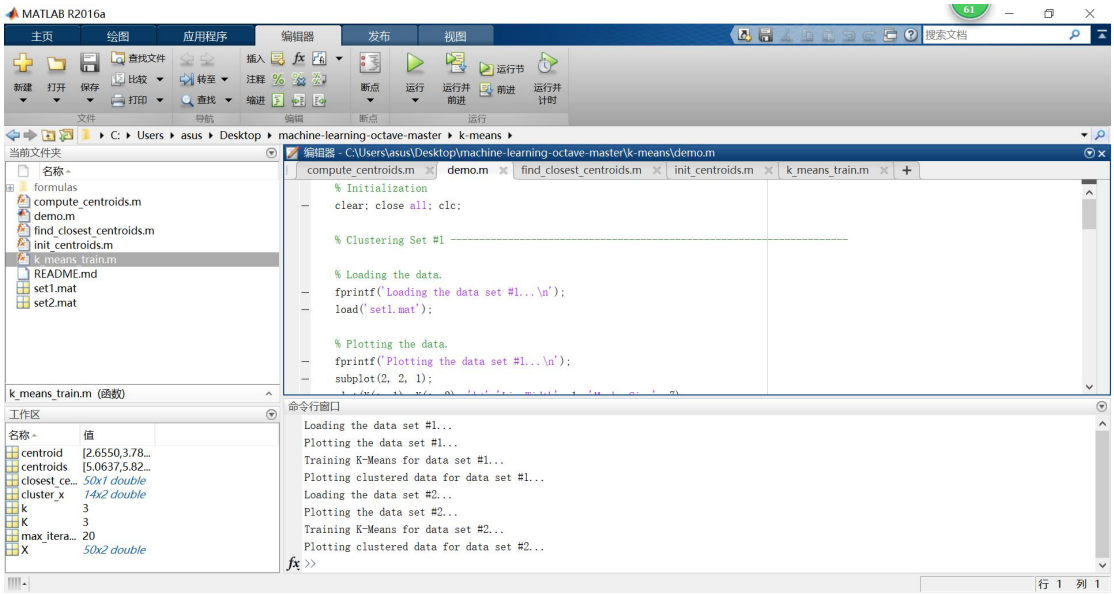
```
K = 3; %中心点数量
max_iterations = 20; %20 次迭代以找到最优的中心点位置
[centroids, closest_centroids_ids] = k_means_train(X, K, max_iterations); %运行 K-means 算法
fprintf('Plotting clustered data for data set #1...\n');
subplot(2, 2, 2);
for k=1:K
    cluster_x = X(closest_centroids_ids == k, :);
    plot(cluster_x(:, 1), cluster_x(:, 2), '+'); %分别绘制 3 个集群，用+表示
    hold on;
    centroid = centroids(k, :);
    plot(centroid(:, 1), centroid(:, 2), 'ko', 'MarkerFaceColor', 'r', 'MarkerSize', 8);
    hold on; %分别绘制 3 个中心点，用黑色圈，红色填充表示
end
title('Clustered Set #1');
hold off;

fprintf('Loading the data set #2...\n');
load('set2.mat');
fprintf('Plotting the data set #2...\n');
subplot(2, 2, 3);
plot(X(:, 1), X(:, 2), 'k+','LineWidth', 1, 'MarkerSize', 7);
title('Training Set #2');
fprintf('Training K-Means for data set #2...\n');
K = 3;
max_iterations = 20;
[centroids, closest_centroids_ids] = k_means_train(X, K, max_iterations);
fprintf('Plotting clustered data for data set #2...\n');
subplot(2, 2, 4);
for k=1:K
    cluster_x = X(closest_centroids_ids == k, :);
    plot(cluster_x(:, 1), cluster_x(:, 2), '+');
    hold on;
    centroid = centroids(k, :);
    plot(centroid(:, 1), centroid(:, 2), 'ko', 'MarkerFaceColor', 'r', 'MarkerSize', 8);
    hold on;
```

```
end
title('Clustered Set #2'); %绘制训练数据集 2 和聚类图像 2 同 1 方法
hold off;

init_centroids.m %通过随机的训练数据随机初始化中心点
function centroids = init_centroids(X, K)
    random_ids = randperm(size(X, 1)); %随机重新排序 X 中数据的序号
    centroids = X(random_ids(1:K), :); %先以前 K 个数据为中心点
end

find_closest_centroids.m %根据到中心点的距离，将训练样本分割成多个聚类
function closest_centroids_ids = find_closest_centroids(X, centroids)
    m = size(X, 1); %设置 m 为 X 矩阵的行数
    K = size(centroids, 1); %设置 K 为 centroids 矩阵的行数
    closest_centroids_ids = zeros(m, 1); %初始化 closest_centroids_ids 为 m 行 1 列 0 矩阵
    for i = 1:m
        distances = zeros(K, 1); %初始化 distances 为 K 行 1 列 0 矩阵
        for j = 1:K
            distances(j) = sum((X(i, :) - centroids(j, :)) .^ 2);
        end %distances 第 j 行为 X 第 i 行数据分别到 3 个中心点的平方和
        [min_distance, centroid_id] = min(distances);
        closest_centroids_ids(i) = centroid_id;
    end %遍历每个数据，找到它最近的中心点，并返回最小距离和中心点的序号
end %closest_centroids_ids(i)包含与第 i 个数据最接近的中心点的序号

compute_centroids.m %计算每个簇的下一个平均中心点
function centroids = compute_centroids(X, closest_centroids_ids, K)
    [m n] = size(X); %X 为 m 行 n 列矩阵
    centroids = zeros(K, n); %初始化 centroids 为 K 行 n 列 0 矩阵
    for centroid_id = 1:K
        centroids(centroid_id, :) = mean(X(closest_centroids_ids == centroid_id, :));
    end %遍历 3 个中心点，得到的行向量 centroids(i,:)包含分配给中心点 i 的数据点的均值
end %通过计算分配给每个中心点的数据点的方法返回新的中心点

k_means_train.m %使用 K-Means 算法进行数据聚类
```

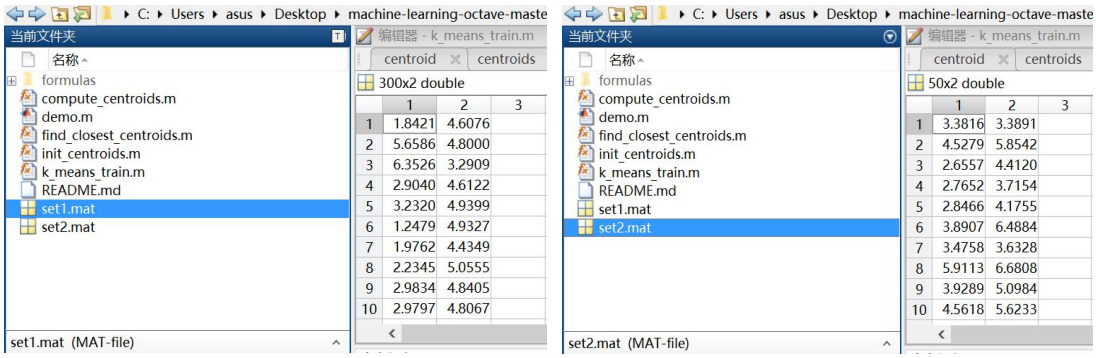
```
function [centroids, closest_centroids_ids] = k_means_train(X, K, max_iterations)
    [m n] = size(X);
    centroids = init_centroids(X, K);
    for i=1:max_iterations
        closest_centroids_ids = find_closest_centroids(X, centroids);
        centroids = compute_centroids(X, closest_centroids_ids, K);
    end
end
```

2.实验结果：

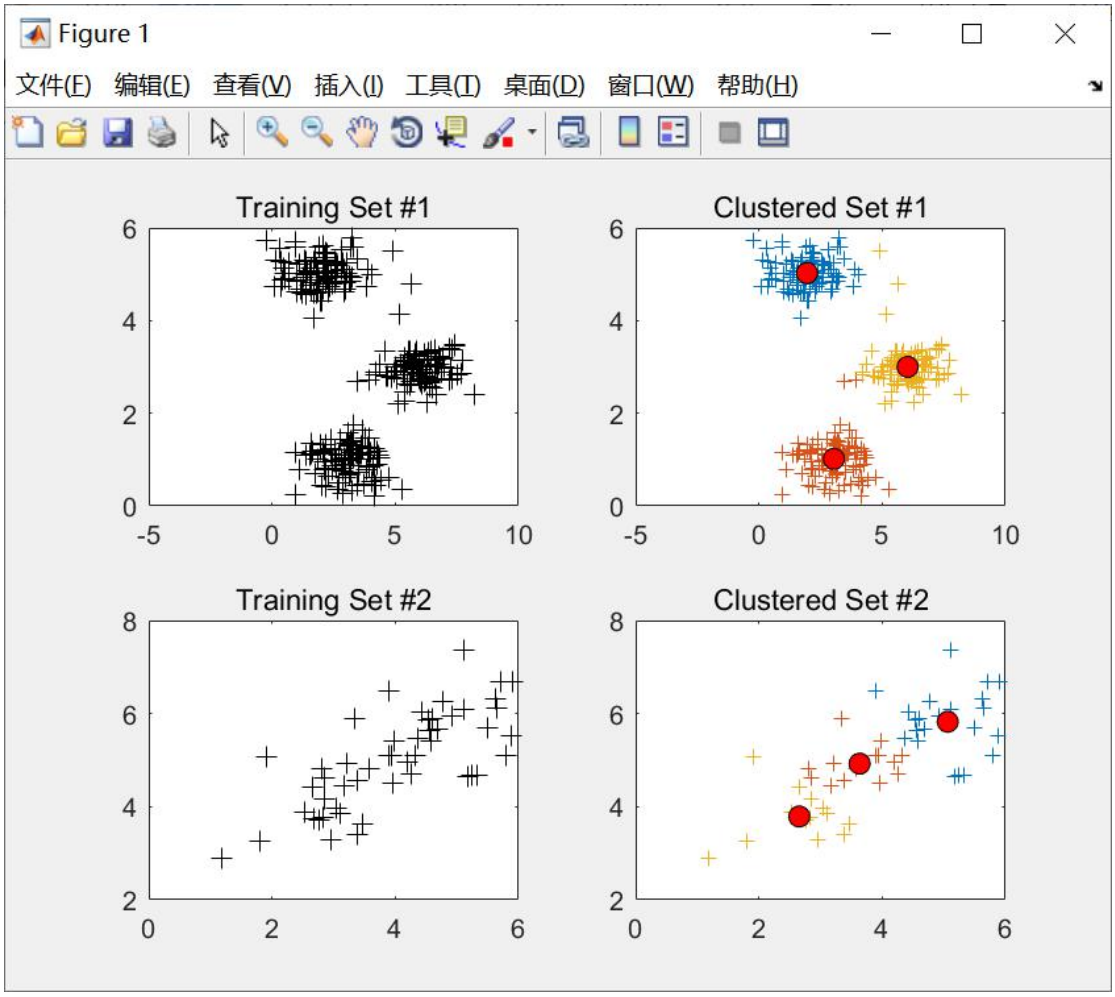
（1）实验运行界面如下图所示。



训练数据集 1 包含 300 个数据，训练数据集 2 包含 50 个数据。



（2）实验结果图，聚类分析分别将两个训练数据集各分为 3 个聚类。



以训练数据集 2 为例，centroids 矩阵中保存着 3 个中心点的坐标，即分配给该中心点的数据点的均值，closest_centroids_ids 矩阵中保存着与各个数据最接近的中心点的序号。