《 — 人工智能方向实习—》

实　习　报　告

专业：　计算机科学与技术

班级：　　12419013

学号：

姓名：

江苏科技大学计算机学院

2016 年 3 月

实验一 数据聚类分析

一、实验目的

编程实现数据聚类的算法。

二、实验内容

k-means 聚类算法。

三、实验原理方法和手段

k-means 算法接受参数 k ；然后将事先输入的 n 个数据对象划分为 k 个聚类以便使得所获得的聚类满足：同一聚类中的对象相似度较高.

四、实验条件

Matlab2014b

五、实验步骤

(1) 初始化 k 个聚类中心。

(2) 计算数据集各数据到中心的距离，选取到中心距离最短的为该数据所属类别。

(3) 计算(2)分类后，k 个类别的中心（即求聚类平均距离）

(4) 继续执行(2)(3)直到 k 个聚类中心不再变化(或者数据集所属类别不再变化)

六、实验代码

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% main.m
% k-means algorithm
% @author matcloud
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear;
close all;
load fisheriris;
X = [meas(:,3) meas(:,4)];
figure;
plot(X(:,1),X(:,2),'ko','MarkerSize',4);
title('fisheriris dataset','FontSize',18,'Color','red');

[idx,ctrs] = kmeans(X,3);
figure;
subplot(1,2,1);
```

```
plot(X(idx==1,1),X(idx==1,2),'ro','MarkerSize',4);
hold on;
plot(X(idx==2,1),X(idx==2,2),'go','MarkerSize',4);
hold on;
plot(X(idx==3,1),X(idx==3,2),'bo','MarkerSize',4);
hold on;
plot(ctrs(:,1),ctrs(:,2),'kx','MarkerSize',12);
title('official kmeans','FontSize',16,'Color','red');

[idx,ctrs] = my_kmeans(X,3);
subplot(1,2,2);
plot(X(idx==1,1),X(idx==1,2),'ro','MarkerSize',4);
hold on;
plot(X(idx==2,1),X(idx==2,2),'go','MarkerSize',4);
hold on;
plot(X(idx==3,1),X(idx==3,2),'bo','MarkerSize',4);
hold on;
plot(ctrs(:,1),ctrs(:,2),'kx','MarkerSize',12);
title('custom kmeans','FontSize',16,'Color','red');

function [idx,ctrs] = my_kmeans(m,k)
    [row col] = size(m);
    %init k centroids
    p = randperm(size(m,1));
    for i = 1 : k
        ctrs(i,:) = m(p(i),:);
    end

    idx = zeros(row,1);%idex is pointer of group
    while 1
        d = dist2matrix(m,ctrs);
        [z,g] = min(d,[],2);
        if(g == idx)
            break;
        else
            idx = g;
        end
        %update ctroids
        for i = 1 : k
            v = find(g == i);
            if v
                ctrs(i,:) = mean(m(v,:),1);
            end
        end
```

```
        end
    end

    function [idx,ctrs] = my_kmeans(m,k)
        [row col] = size(m);
        %init k centroids
        p = randperm(size(m,1));
        for i = 1 : k
            ctrs(i,:) = m(p(i),:);
        end

        idx = zeros(row,1);%idex is pointer of group
        while 1
            d = dist2matrix(m,ctrs);
            [z,g] = min(d,[],2);
            if(g == idx)
                break;
            else
                idx = g;
            end
            %update ctroids
            for i = 1 : k
                v = find(g == i);
                if v
                    ctrs(i,:) = mean(m(v,:),1);
                end
            end
        end
    end
```

七、实验结果

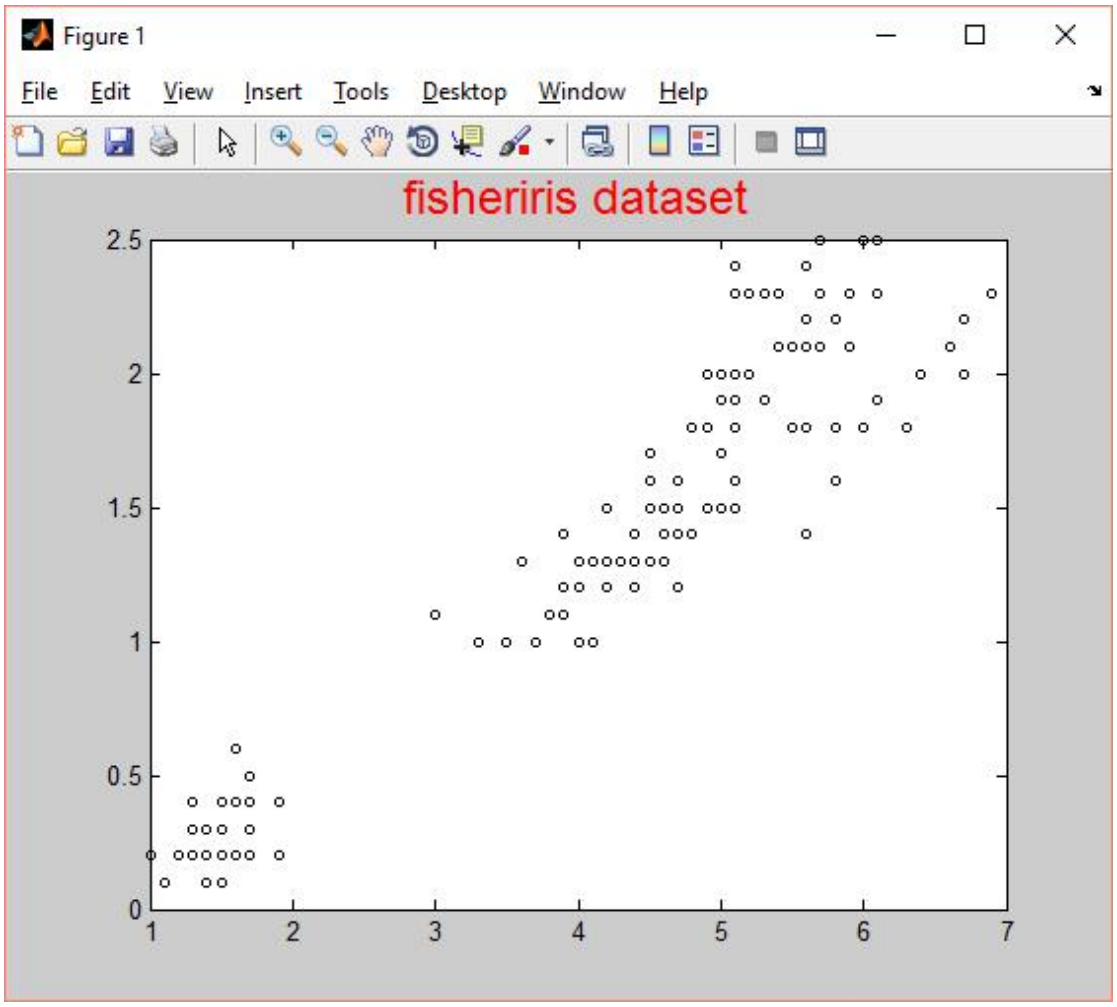

图 1-1 未聚类数据

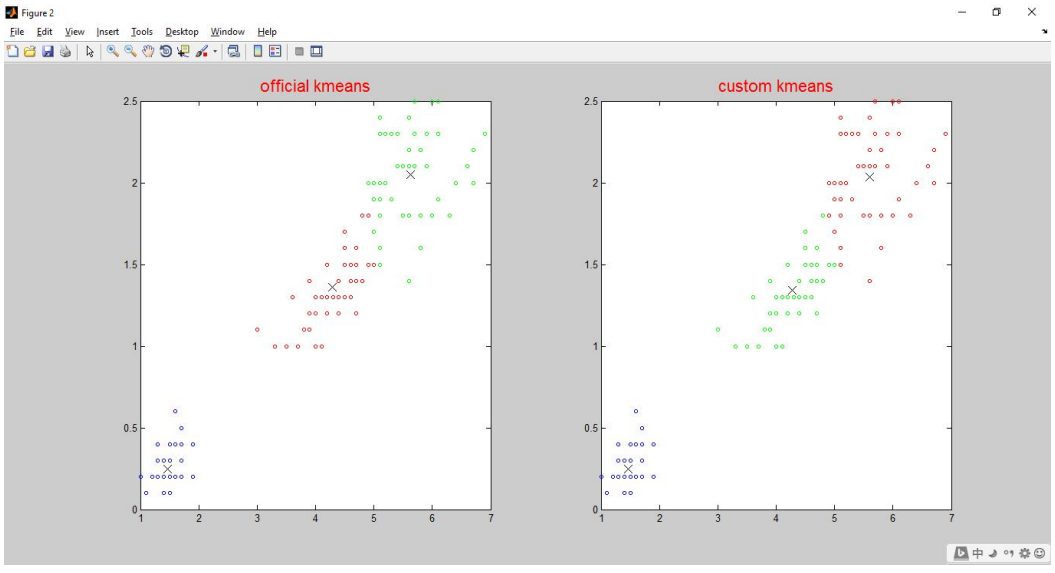

图 1-2 聚类后

八、实验分析

算法的时间复杂度上界为 O(n*k*t)，其中 t 是迭代次数。k-means 算法是一种基于

样本间相似性度量的间接聚类方法，属于非监督学习方法。此算法以 k 为参数，把 n 个对象分为 k 个簇，以使簇内具有较高的相似度，而且簇间的相似度较低。相似度的计算根据一个簇中对象的平均值（被看作簇的重心）来进行。此算法首先随机选择 k 个对象，每个对象代表一个聚类的质心。对于其余的每一个对象，根据该对象与各聚类质心之间的距离，把它分配到与之最相似的聚类中。然后，计算每个聚类的新质心。重复上述过程，直到准则函数收敛。k-means 算法是一种较典型的逐点修改迭代的动态聚类算法，其要点是以误差平方和为准则函数。逐点修改类中心：一个象元样本按某一原则，归属于某一组类后，就要重新计算这个组类的均值，并且以新的均值作为凝聚中心点进行下一次象元素聚类；逐批修改类中心：在全部象元样本按某一组的类中心分类之后，再计算修改各类的均值，作为下一次分类的凝聚中心点。

实验二 主成分分析

一、实验目的

编程实现主成分的算法。

二、实验内容

PCA 主成分分析算法。

三、实验原理方法和手段

PCA 的原理就是将原来的样本数据投影到一个新的空间中，相当于我们在矩阵分析里面学习的将一组矩阵映射到另外的坐标系下。通过一个转换坐标，也可以理解成把一组坐标转换到另外一组坐标系下，但是在新的坐标系下，表示原来的原本不需要那么多的变量，只需要原来样本的最大的一个线性无关组的特征值对应的空间的坐标即可。

四、实验条件

Matlab2014b

五、实验步骤

(1) 求 dataAdjust 矩阵

(2) 求 dataAdjust 的协方差矩阵

协方差公式

$$\operatorname{cov}(X,Y)=\frac{\sum_i^n(X_{\mathrm{i}}-\overline{X})(Y-\overline{Y})}{n-1}$$

协方差矩阵

$$C_{n\times n}=(c_{i,j},c_{i,j}=\operatorname{cov}(Dim_i,Dim_j))$$

(3) 求协方差矩阵的特征向量及特征值

(4) 取特征值最大的的特征向量 eigenVectors

(5) 降维矩阵 finalData = dataAdjust * eigenVectors

六、实验代码

```
data = [2.5 2.4;0.5 0.7;2.2 2.9;1.9 2.2;3.1 3.0;2.3 2.7;2 1.6;1 1.1;1.5
1.6;1.1 0.9];
  dim1_mean = mean(data(:,1));
  dim2_mean = mean(data(:,2));
  dataAdjust = [data(:,1)-dim1_mean,data(:,2)-dim2_mean];
```

```
c = cov(dataAdjust);
[vectors,values] = eig(c);
values = values*ones(2,1);
[max_v,max_idx] = max(values,[],1);
eigenVectors = vectors(:,max_idx);
finalData = dataAdjust * eigenVectors;
```

七、实验结果

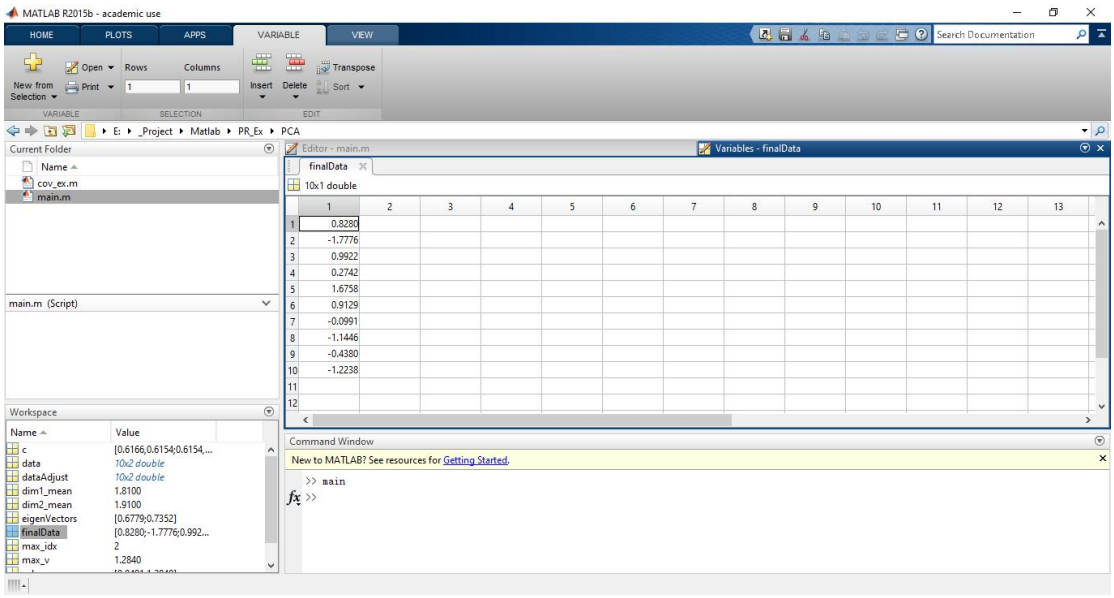

图 2-1 实验结果

八、实验分析

主成分分析，是考察多个变量间相关性一种多元统计方法，研究如何通过少数几个主成分来揭示多个变量间的内部结构，即从原始变量中导出少数几个主成分，使它们尽可能多地保留原始变量的信息，且彼此间互不相关.通常数学上的处理就是将原来 P 个指标作线性组合，作为新的综合指标。

实验三 最近邻分类器

一、实验目的

编程实现最近邻分类器算法。

二、实验内容

最近邻分类器算法，这里采用 k 近邻算法。

三、实验原理方法和手段

最近邻分类为监督学习方法，已知 n 个类别，判定给定样本属于哪个类别。

四、实验条件

Matlab2014b

五、实验步骤

(1) 计算样本到各数据集点的距离 D

① 欧式距离 $\mathrm{d}(x,y)=\|x-y\|=\left[\sum_{i=1}^{n}(x_i-y_i)^2\right]^{\frac{1}{2}}$ c

② 绝对值距离 $\mathrm{d}(x,y)=\|x-y\|=\sum_{i=1}^{n}|x_i-y_i|$

③ 明氏距离 $d(x,y)=\|x-y\|=\left[\sum_{i=1}^{n}\left|x_i-y_i^{\,p}\right|\right]^{\frac{1}{p}}$

④ 马氏距离 $d(x,y)=\|x-y\|=-\sum_{i=1}^{n}\frac{x_iy_i}{\sqrt{\lambda_i}}$（$\lambda_i$ 为对应的特征值）

⑤ 余弦距离 $d(x,y)=\|x-y\|=-\cos(x,y)=-\frac{\sum_{i=1}^{n}x_iy_i}{\sqrt{\sum_{i=1}^{n}x_i^{\,2}\sum_{i=1}^{n}y_i^{\,2}}}$

(2) 对 D 排序

(3) 给定 k 值(即邻居数)，从 D 中选取 k 个数据，统计 k 个数据中所属类别的个数 C。

(4) C 中值最大的便是该样本所属类别。

六、实验代码

```
close all;
clear;
clc;
red = randn(100,2)+ones(100,2);
red = [red ones(100,1)];
```

```
green = randn(100,2)-ones(100,2);
green = [green ones(100,1)*2];
data = [red;green];

figure;
plot(red(:,1),red(:,2),'ro','MarkerSize',4);
hold on;
plot(green(:,1),green(:,2),'go','MarkerSize',4);

blue_sample = randn(1,2);

hold on;
plot(blue_sample(:,1),blue_sample(:,2),'bo','MarkerSize',4);

%give a k value
k = input('input neighbors count');
[row,col] = size(data);
for i = 1 : row
    d(:,i) = norm(data(i,1:2) - blue_sample(1,:));
end
[d,idx] = sort(d);
for i = 1 : k
    k_vector(:,i) = idx(:,i);
end

%caculate category
redCount = 0;
greenCount = 0;
for i = 1 : k
    tag = data(k_vector(1,i),3);
    if (tag == 1)
        redCount = redCount+1;
    else if(tag == 2)
            greenCount = greenCount+1;
        end
    end
end

if(redCount > greenCount)
    blue_sample = [blue_sample 1];
    disp('sample is red');
else
    blue_sample = [blue_sample 2];
    disp('sample is green');
```

```
end
```

七、实验结果

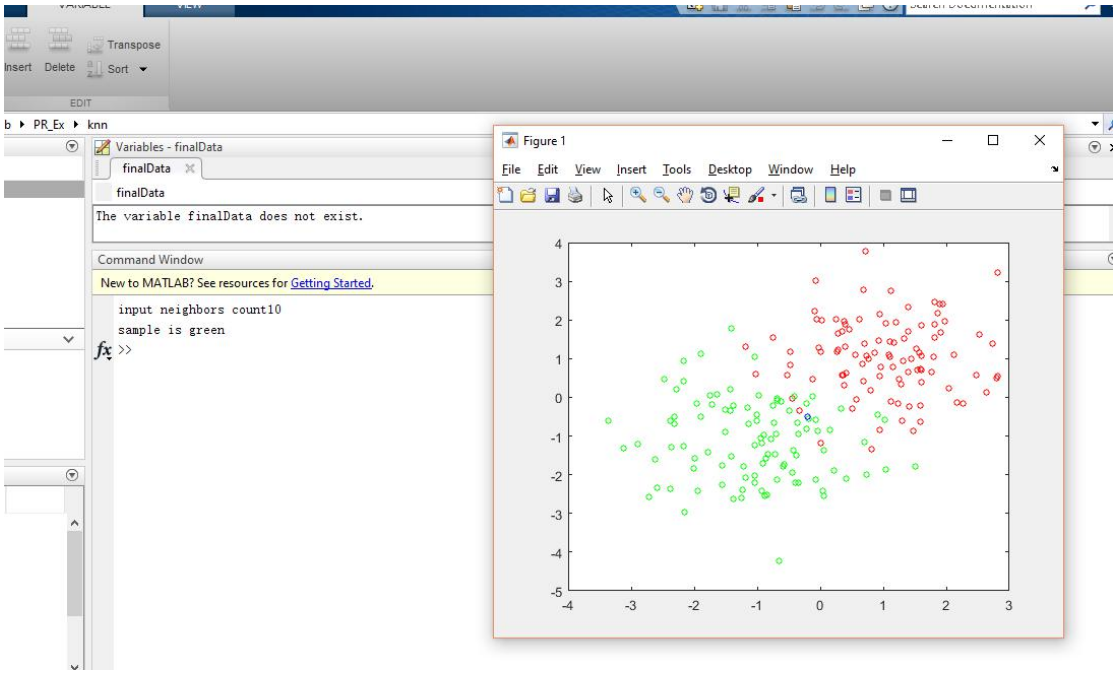

图 3-1 实验结果

八、实验分析

KNN 算法本身简单有效，它是一种 lazy-learning 算法，分类器不需要使用训练集进行训练，训练时间复杂度为 0。KNN 分类的计算复杂度和训练集中的文档数目成正比，也就是说，如果训练集中文档总数为 n，那么 KNN 的分类时间复杂度为 O(n)。

实验四 贝叶斯分类器

一、实验目的

编程实现贝叶斯分类器算法。

二、实验内容

贝叶斯分类器算法。

三、实验原理方法和手段

已知类别，给定一样本判定样本所属类别。

四、实验条件

Matlab2014b

五、实验步骤

(1) 已知 k 个类别

(2) 计算 k 个类别所占全体的比重 Pr(k)

(3) 给定值 radius，在二维空间，以样本点为圆心以 radiu 为半径作圆。

(4) 统计圆内 k 个类别的分布情况(在圆内包含该类多少个数据点)记为 C(k)

(5) 计算圆内分布比重 Pr_c(k)

(6) 根据贝叶斯公式 $\mathrm{P}(A/B)=P(A)\times\frac{P(B/A)}{P(B)}$ 计算各类最终比重，取值大的作为样本类别。

六、实验代码

```
clear;
close all;
%init two cluters
rH = randi([80,100]);
gH = randi([80,100]);
red = randn(rH,2)+ones(rH,2);
green = randn(gH,2)-ones(gH,2);

red = [red ones(rH,1)];
green = [green ones(gH,1)*2];
data = [red;green];

total = rH+gH;
```

```
pr_red = rH/(total);
pr_green = gH/(total);

%init a sample
sample_blue = randn(1,2);

plot(red(:,1),red(:,2),'ro','MarkerSize',4);
hold on;
plot(green(:,1),green(:,2),'go','MarkerSize',4);
hold on;
plot(sample_blue(:,1),sample_blue(:,2),'b*','MarkerSize',6);

for i = 1 : total
    p = data(i,1:2);
    tmp = sample_blue - p;
    d(:,i) = sqrt(dot(tmp,tmp));
end
%select an circle(center is sample_blue)
radius = 5;
redCount = 0;
greenCount = 0;
for i = 1 : total
    if(d(:,i) <= radius)
        if(data(i,3) == 1)%red cluster
            redCount = redCount + 1;
        else if(data(i,3) == 2)%green cluster
                greenCount = greenCount+1;
            end
        end
    end
end

pr_redInCircle = redCount/rH;
pr_greenInCircle = greenCount/gH;

pr_redFinal = pr_red * pr_redInCircle;
pr_greenFinal = pr_green * pr_greenInCircle;

fprintf('final red pr = %f\n',pr_redFinal);
fprintf('final green pr = %f\n',pr_greenFinal);
if(pr_redFinal >= pr_greenFinal)
    disp('sample is red cluster');
else
    disp('sample is green cluster');
```

```
end
```

七、实验结果

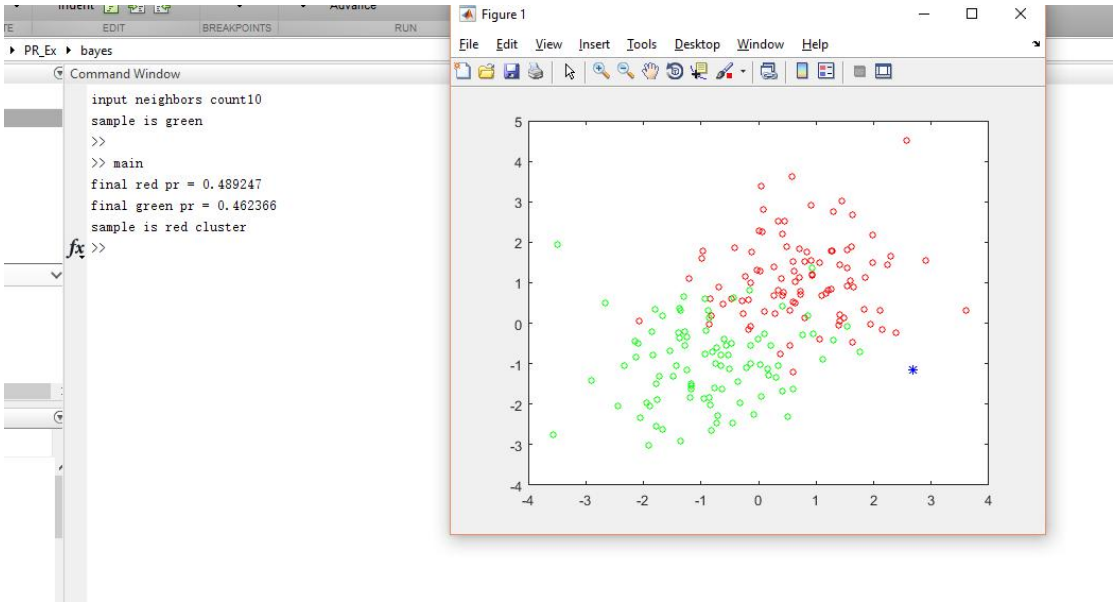

图 4-1 实验结果

八、实验分析

对于某些类型的概率模型，在监督式学习的样本集中能获取得非常好的分类效果。在许多实际应用中，朴素贝叶斯模型参数估计使用最大似然估计方法；换而言之，在不用到贝叶斯概率或者任何贝叶斯模型的情况下，朴素贝叶斯模型也能奏效。

实验五 特征提取算法

一、实验目的

编程实现特征提取算法。

二、实验内容

harris 特征提取算法。

三、实验原理方法和手段

图像中的特征点即为图像边缘直线形成夹角而成的角点。图像中较突出的尖锐点和其它有特殊意义的点，这些点反映了地物的特征，对研究具有重要意义。在点特征提取的算法中，主要包含了 Susan 算法、Harris 算法和 Moravec 算法，这些算法可以对图像中角点进行提取，从而应用于实践生产中，例如对建筑物角点提取，人脸中眼睛部位角点的提取。

四、实验条件

Matlab2014b

五、实验步骤

(1) 计算图像 I(x,y)在 x 和 y 方向的梯度$\mathrm{I_x}$，$\mathrm{I_y}$.

$$I_x = \frac{\partial I}{\partial_x} = I \otimes \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}, I_y = \frac{\partial I}{\partial x} = I \otimes \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}^T$$

(2)计算图像两个方向梯度的乘积

$$\mathrm{I_x}^2 = \mathrm{I_x I_y},\ \mathrm{I_y}^2 = I_y I_y, I_{xy} = I_x I_y$$

(3)使用高斯函数对$\mathrm{I_x}^2$，$\mathrm{I_y}^2$和$\mathrm{I_{xy}}$进行高斯加权(取$\sigma = 1$)

$$\mathrm{A = g(I_x^2) = I_x^2 \otimes w}, C = g(I_y^2) = I_y^2 \otimes w, B = g(I_{xy}) = I_{xy} \otimes w$$

(4)计算每个像素的 Harris 响应值 R，并对小于某一阈值 t 的 R 置为零

$$\mathrm{R = \{R : detM - \alpha (traceM)^2 < t\}}$$

(5)在 3*3 或 5*5 的邻域内进行非最大值抑制，局部最大值点即为图像中的角点

六、实验代码

```
%function:
%       Harris角点检测
%注意：
```

```
%       matlab自带的corner函数即可实现harris角点检测。但考虑到harris角点的经典性，本程序将其实现，纯粹出于学习目的，了解特征点检测的方法。
%       其中所有参数均与matlab默认保持一致
%
%清空变量，读取图像
clear;close all
src= imread('images/girl.jpg');

gray=rgb2gray(src);
gray = im2double(gray);
%缩放图像，减少运算时间
gray = imresize(gray, 0.2);

%计算X方向和Y方向的梯度及其平方
X=imfilter(gray,[-1 0 1]);
X2=X.^2;
Y=imfilter(gray,[-1 0 1]');
Y2=Y.^2;
XY=X.*Y;

%生成高斯卷积核，对X2、Y2、XY进行平滑
h=fspecial('gaussian',[5 1],1.5);
w=h*h';
A=imfilter(X2,w);
B=imfilter(Y2,w);
C=imfilter(XY,w);

%k一般取值0.04-0.06
k=0.04;
RMax=0;
size=size(gray);
height=size(1);
width=size(2);
R=zeros(height,width);
for h=1:height
    for w=1:width
        %计算M矩阵
        M=[A(h,w) C(h,w);C(h,w) B(h,w)];
        %计算R用于判断是否是边缘
        R(h,w)=det(M) - k*(trace(M))^2;
        %获得R的最大值，之后用于确定判断角点的阈值
        if(R(h,w)>RMax)
            RMax=R(h,w);
        end
```

```
    end
end

%用Q*RMax作为阈值，判断一个点是不是角点
Q=0.01;
R_corner=(R>=(Q*RMax)).*R;

%寻找3x3邻域内的最大值，只有一个交点在8邻域内是该邻域的最大点时，才认为该点是角点
fun = @(x) max(x(:));
R_localMax = nlfilter(R,[3 3],fun);

%寻找既满足角点阈值，又在其8邻域内是最大值点的点作为角点
%注意：需要剔除边缘点
[row,col]=find(R_localMax(2:height-1,2:width-1)==R_corner(2:height-1,2:width-1));

%绘制提取到的角点
figure('name','Result');
subplot(1,2,1),imshow(gray),title('my-Harris'),
hold on
plot(col,row, 'b*'),
hold off

%用matlab自带的edge函数提取Harris角点，对比效果
C = corner(gray);
subplot(1,2,2),imshow(gray),title('matlab-conner'),
hold on
plot(C(:,1), C(:,2), 'r*');
hold off
```

七、实验结果

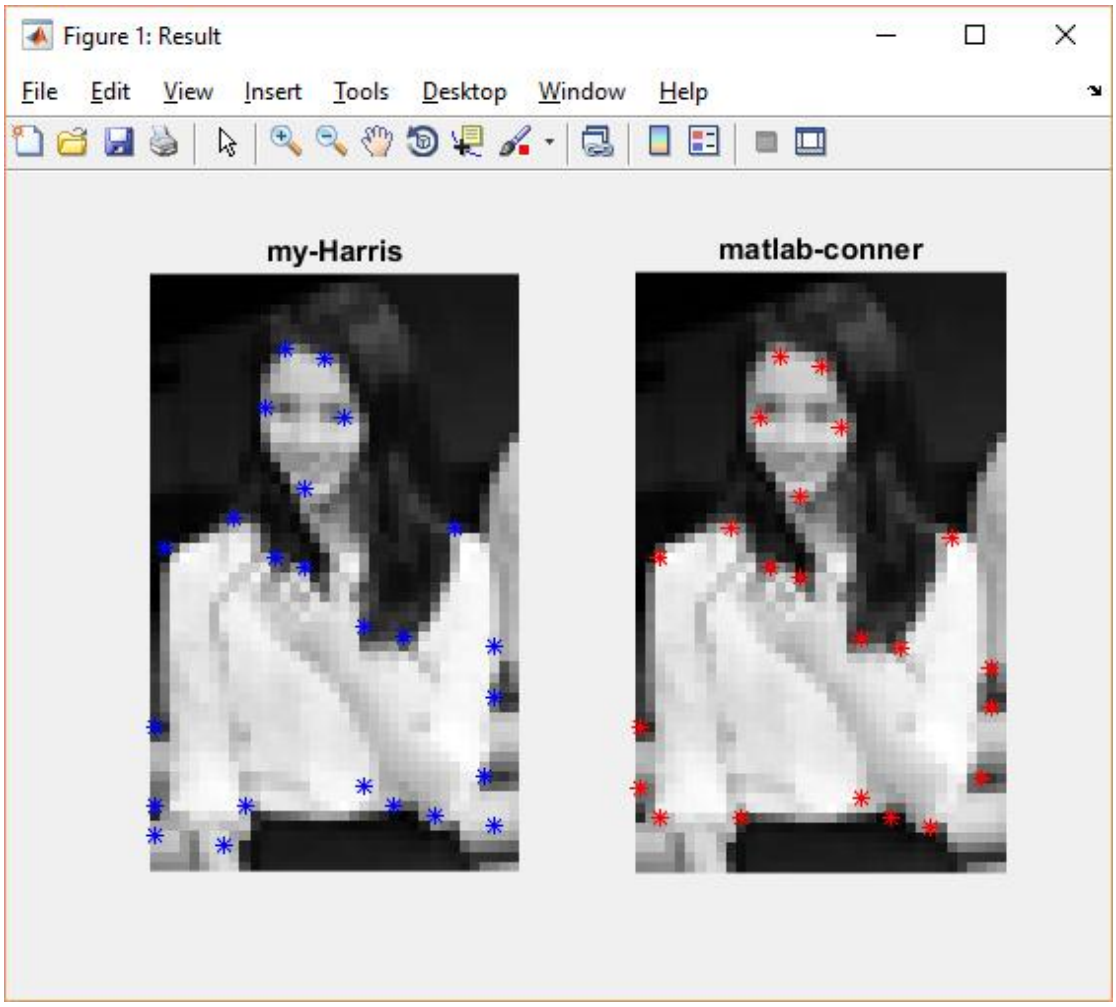

图 5-1 实验结果

八、实验分析

对于某些类型的概率模型，在监督式学习的样本集中能获取得非常好的分类效果。在许多实际应用中，朴素贝叶斯模型参数估计使用最大似然估计方法；换而言之，在不用到贝叶斯概率或者任何贝叶斯模型的情况下，朴素贝叶斯模型也能奏效。