南京信息工程大学

数据分析课程设计报告

学院名称：　数学与统计学院

专　　业：　信息与计算科学（嵌入式培养）

班　　级：　19 信嵌（1）班

学　　号：　201983160037

姓　　名：　强盛周

二〇二二 年 十一 月 三十 日

目　录

1　数据的描述性分析

实验目的：掌握和理解相关系数和及数据的数字特征等。

1.1　实验题目

通过模拟方法生成二元正态分布向量（样本容量为 500），其均值设定为(0，0）’，而协方差矩阵分别为如下情形：

$$1),\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix};\qquad 2),\begin{pmatrix}0.2 & 0\\ 0 & 0.2\end{pmatrix};$$

$$3),\begin{pmatrix}4 & 0\\ 0 & 4\end{pmatrix};\qquad 4),\begin{pmatrix}0.2 & 0\\ 0 & 4\end{pmatrix};$$

$$5),\begin{pmatrix}4 & 0\\ 0 & 0.2\end{pmatrix};\qquad 6),\begin{pmatrix}1 & 0.5\\ 0.5 & 1\end{pmatrix};$$

$$7),\begin{pmatrix}0.3 & 0.5\\ 0.5 & 4\end{pmatrix};\qquad 8),\begin{pmatrix}4 & 0.5\\ 0.5 & 0.3\end{pmatrix};$$

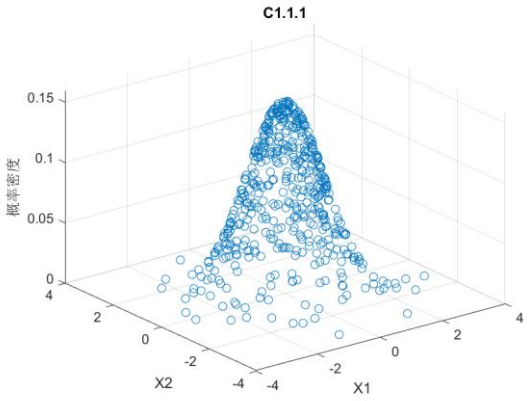
请画出图像，并给出评价。

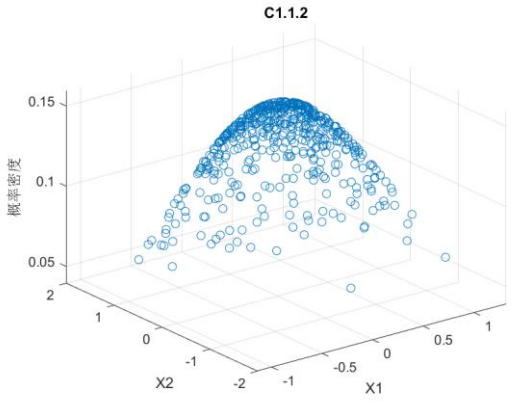
1.1.1　实验过程描述

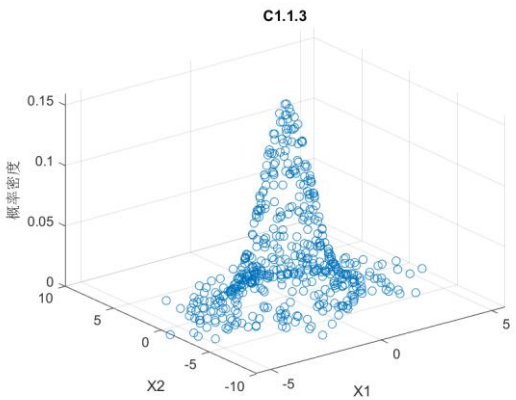
```
%% C1.1 bivariate normal distribution vector
% Author: Alephant
% Date: 15 Nov 2022
clc;
close all;
clear;


%% paramters
n = 500;
mu = [0; 0];
Sigmas = {[1, 0; 0, 1]; [0.2, 0; 0, 0.2];
          [4, 0; 0, 4]; [0.2, 0; 0, 4];
          [4, 0; 0, 0.2]; [0.2, 0; 0, 4];
          [0.3, 0.5; 0.5, 4]; [4, 0.5; 0.5, 0.3]
          };
figpath = 'figures/';
figtype = '.png';
```

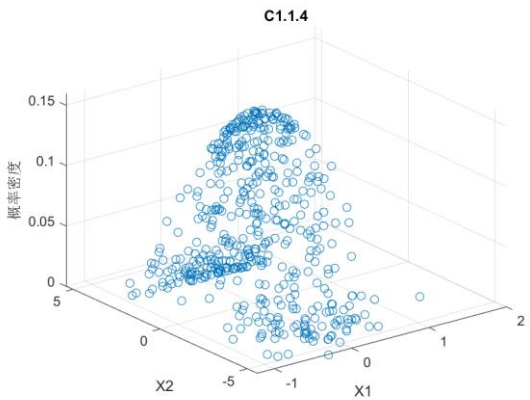
```
%% run
for i = 1:8
    % define paramaters
    figure(i)
    Sigma = cell2mat(Sigmas(i));
    figname = ['C1.1.', num2str(i)];

    % calculate
    X = mvnrnd(mu, Sigma, n);
    y = mvnpdf(X);

    % plot fig
    scatter3(X(:,1),X(:,2),y)
    xlabel('X1')
    ylabel('X2')
    zlabel('概率密度')
    title(figname)

    % save fig
    saveas(gcf, [figpath, figname, figtype])
end
```

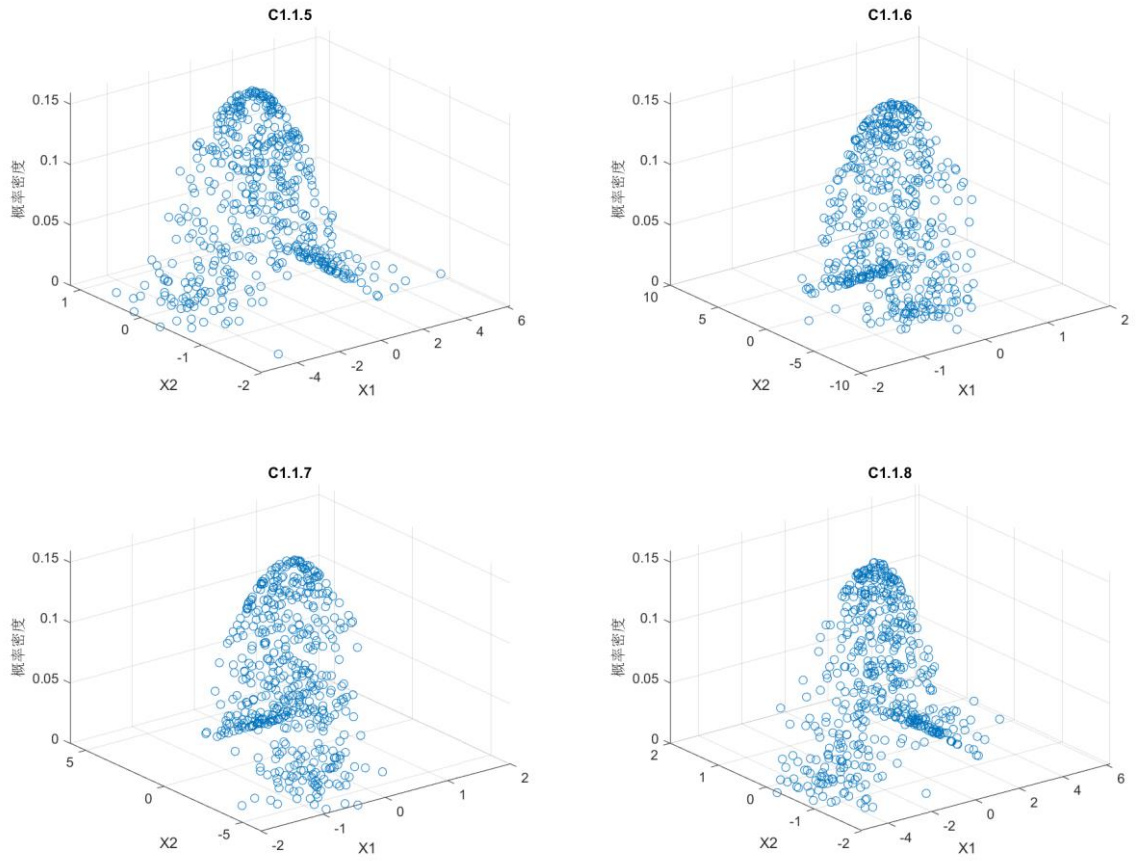
1.1.2　结果分析











多变量正态分布亦称为多变量高斯分布。它是单维正态分布向多维的推广。它同矩阵正态分布有紧密的联系。

1.2　实验题目

对 data_1 的数据剔除 ID 为 841 的观测值，

1. 对四个变量 EXPE、QUAL、LOYA 和 SATI 画出散点图矩阵；
2. 画出各变量的箱线图；
3. 计算观测数据的 pearson 相关系数矩阵，并做相关性的显著性检验。

参考资料：描述性分析参考文档

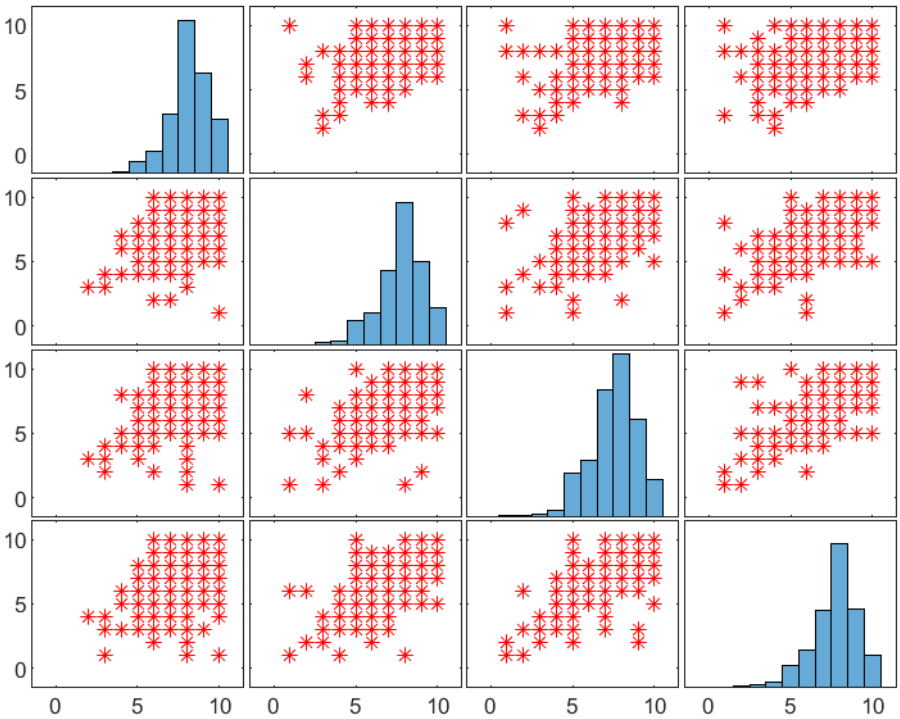
1.2.1　实验过程描述

```
%% C1.2 Analyze Observations
% Author: Alephant
% Date: 26 Nov 2022
clc;
close all;
clear;


%% load data
T = readtable('data_1.xls');
% delete 841
```

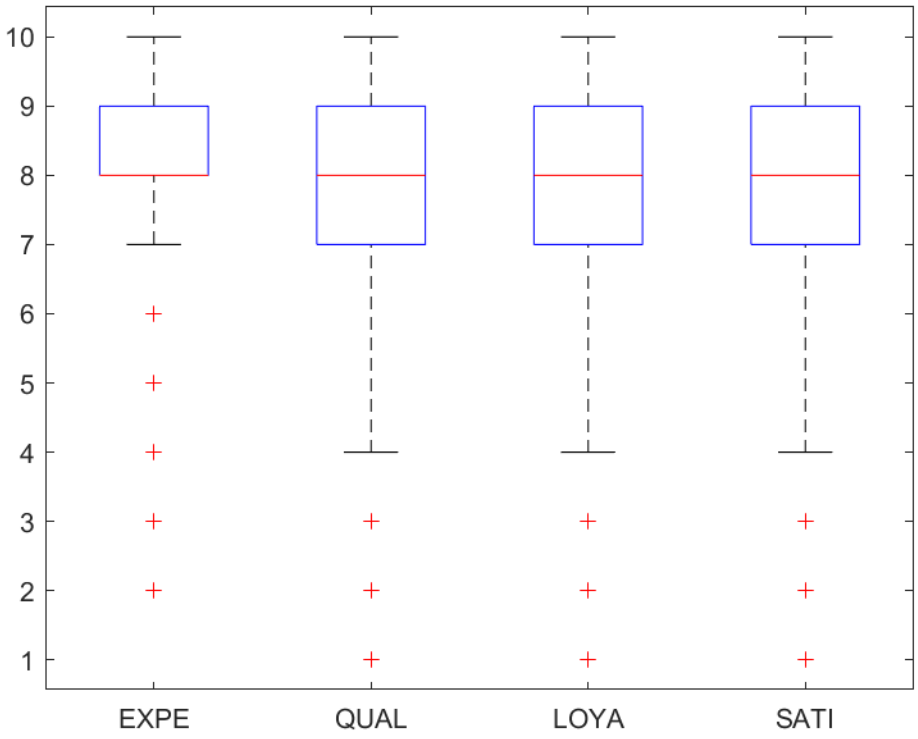
```
T(T.ID==841,:) = [];
A = table2array(T(:,2:5));


%% paramters
figpath = 'figures/';
figtype = '.png';


%% C1.2.1 plotmatrix
figure
plotmatrix(A,'*r');
saveas(gcf, [figpath, 'C1.2.1', figtype])


%% C1.2.2 boxplot
figure
boxplot(A,{'EXPE','QUAL','LOYA','SATI'})
saveas(gcf, [figpath, 'C1.2.2', figtype])


%% C1.2.3 Pearson correlation coefficient
[R, P] = corrcoef(A)
```

1.2.2　结果分析

EXPE、QUAL、LOYA 和 SATI 的散点图矩阵，图窗的第 i 行、第 j 列中的子图是 X 的第 i 列相对于 X 的第 j 列的散点图。



箱线图（Boxplot）也称箱须图（Box-whisker Plot），是利用数据中的五个统计量：最小值、第一四分位数、中位数、第三四分位数与最大值来描述数据的一种方法，它也可以粗略地看出数据是否具有有对称性，分布的分散程度等信息，特别可以用于对几个样本的比较。



在统计学中，皮尔逊积矩相关系数（英语：Pearson product-moment correlation coefficient，又称作 PPMCC 或 PCCs，文章中常用 r 或 Pearson's r 表示）用于度量两个变量 X 和 Y 之间的相关（线性相关），其值介于-1 与 1 之间。在自然科学领域中，该系数广泛用于度量两个变量之间的相关程度。通常情况下通过以下取值范围判断变量的相关强度：

相关系数 （均取绝对值后）：

0.8-1.0 极强相关

0.6-0.8 强相关

0.4-0.6 中等程度相关

0.2-0.4 弱相关

0.0-0.2 极弱相关或无相关

R =

1.0000	0.5913	0.5362	0.5470
0.5913	1.0000	0.6515	0.7104
0.5362	0.6515	1.0000	0.7552
0.5470	0.7104	0.7552	1.0000

可以看出 EXPE 与 QUAL，LOYA，SATI 都是中等程度相关，QUAL 与 LOYA，SATI 和 LOYA 与 SATI 都是强相关。

讨论两变量是否相关必须讨论显著性水平，不谈 P 值之谈相关系数大小是无意义的，两者之间的相关关系可能只是偶然因素引起的，所以我们要对两个变量之间的相关关系的显著性水平进行判断；

采用假设检验的方法：

原假设 H0：　R=0 两变量之间不存在线性关联

备择假设 H1：　R 不等于 0，两变量之间存在线性关联

根据假设检验方法，在零假设成立的条件下，即假设两变量不存在相关性的前提下，计算出两变量不存在相关性的概率值（P 值），如果这个 P 值很小，说明两变量不存在相关性的概率很小，我们就可以拒绝原假设，接受备择假设，那么这里我们就需要一个阈值

这里以 5%为阈值（这里的阈值也称为显著水平），如果 p<0.05,则说明可以拒绝原假设。接受备择假设，即两变量之间存在显著的线性关联。

所有变量的 p 值都为 0，即都小于 0.05；这说明**它们两两之间都存在显著的线性关联。**

P =

1.0000	0.0000	0.0000	0.0000
0.0000	1.0000	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000
0.0000	0.0000	0.0000	1.0000

1.3　实验题目

已知8个乳房肿瘤病灶组织的样本，其中前3个为良性肿瘤，后5个为恶性肿瘤。数据为细胞核显微图像的5个量化特征：细胞核直径，质地，周长，面积，光滑度。已知样本的数据如下：

13.54,14.36,87.46,566.3,0.09779
13.08,15.71,85.63,520,0.1075
9.504,12.44,60.34,273.9,0.1024
17.99,10.38,122.8,1001,0.1184
20.57,17.77,132.9,1326,0.08474
19.69,21.25,130,1203,0.1096
11.42,20.38,77.58,386.1,0.1425
20.29,14.34,135.1,1297,0.1003

试根据已知样本利用距离判别（分别用协方差矩阵相等、协方差矩阵不等）对下面未知种类的三个样本进行分类：

1. 16.6,28.08,108.3,858.1,0.08455
2. 20.6,29.33,140.1,1265,0.1178
3. 7.76,24.54,47.92,181,0.05263

1.3.1　实验过程描述

```
%% C1.3 Sample Classification
% Author: Alephant
% Date: 25 Nov 2022
```

```
clc;
close all;
clear;


%% load data
% the first three are benign tumors
% the last five are malignant tumors
group = [1; 1; 1; 2; 2; 2; 2; 2];

training = [13.54,14.36,87.46,566.3,0.09779;
13.08,15.71,85.63,520,0.1075;
9.504,12.44,60.34,273.9,0.1024;
17.99,10.38,122.8,1001,0.1184;
20.57,17.77,132.9,1326,0.08474;
19.69,21.25,130,1203,0.1096;
11.42,20.38,77.58,386.1,0.1425;
20.29,14.34,135.1,1297,0.1003];

sample = [16.6,28.08,108.3,858.1,0.08455;
20.6,29.33,140.1,1265,0.1178;
7.76,24.54,47.92,181,0.05263];

training1 = training(1:3, :);
training2 = training(4:8, :);
mu1 = mean(training1);
mu2 = mean(training2);


% fail to use matlab 'classify'
% [C,err] = classify(sample, training, group, 'mahalanobis');
% Error using classify (line 282)
% The covariance matrix of each group in TRAINING must be positive definite.


%% C1.3.1 The covariance matrices are equal
Sigma = cov(training);
a1 = inv(Sigma) * mu1';
b1 = (-1/2) * mu1 * inv(Sigma) * mu1';
W1 = a1' * sample' + b1;

a2 = inv(Sigma) * mu2';
b2 = (-1/2) * mu2 * inv(Sigma) * mu2';
W2 = a2' * sample' + b2;

result_equal = 2 * ones(3,1);
```

```
result_equal((W1-W2)>=0) = 1;
disp("协方差矩阵相同时，样本分类结果：")
disp("（1=良性肿瘤，2=恶性肿瘤）")
disp(result_equal)
disp("-----------")


%% C1.3.2 The covariance matrices are unequal
warning("off")
Sigma1 = cov(training1);
Sigma2 = cov(training2);

d1_square = zeros(3,1);
d2_square = zeros(3,1);
for i = 1:3
    x = sample(i,:);
    d1_square(i) = (x - mu1) * inv(Sigma1) * (x - mu1)';
    d2_square(i) = (x - mu2) * inv(Sigma2) * (x - mu2)';
end

result_unequal = 2 * ones(3,1);
result_unequal((d2_square-d1_square)>=0) = 1;
disp("协方差矩阵不同时，样本分类结果：")
disp("（1=良性肿瘤，2=恶性肿瘤）")
disp(result_unequal)
disp("-----------")
```

1.3.2　结果分析

马哈拉诺比斯距离是基于样本分布的一种距离。物理意义就是在规范化的主成分空间中的欧氏距离。所谓规范化的主成分空间就是利用主成分分析对一些数据进行主成分分解。再对所有主成分分解轴做归一化，形成新的坐标轴。由这些坐标轴张成的空间就是规范化的主成分空间。换句话说，主成分分析就是把椭球分布的样本改变到另一个空间里，使其成为球状分布。而马哈拉诺比斯距离就是在样本呈球状分布的空间里面所求得的欧式距离。当然，上面的解释只是对椭球分布而言，对一般分布，只能消除分布的二阶相关性，而不能消除高阶相关性。

协方差矩阵相同时，样本分类结果：

（1=良性肿瘤，2=恶性肿瘤）

1

1

1

三个样本的 W_1 都大于 W_2，说明它们都是良性肿瘤。

协方差矩阵不同时，样本分类结果：

（1=良性肿瘤，2=恶性肿瘤）

2

2

2

三个样本的d_1^2都大于d_2^2，说明它们都是恶性肿瘤。

2　主成分分析

实验目的：利用主成分分析进行数据降维。

2.1　实验题目

附表（data21.xls）中列出了2007年我国31个省、市、自治区和直辖市的农村居民家庭平均每人全年消费性支出的8个主要变量数据.

(1)试根据这8个主要变量的观测数据进行主成分分析，给出各主成分的贡献率;

(2)试分析前两个主成分的意义，并按第一主成分得分将31个省、市、自治区和直辖市排序;

(3)画出前两个主成分得分的散点图，并在散点图上标注上各个地区的名称.

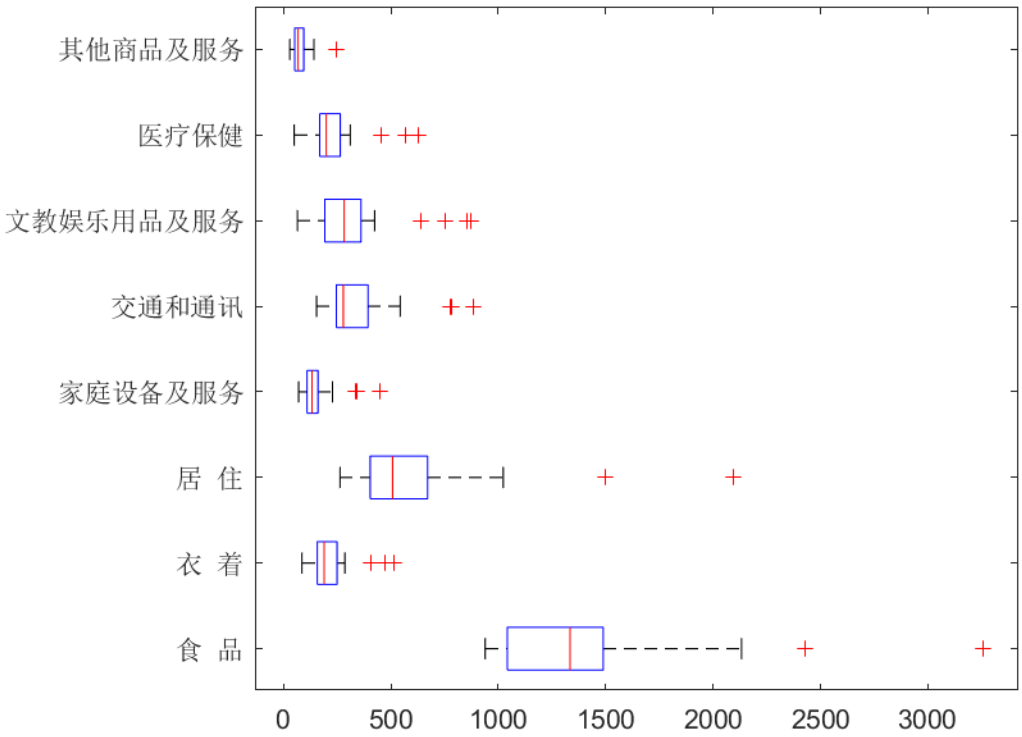
参考资料：Matlab　帮助文档help中的princomp，这里提供一个简单的注释pca.docx及相应程序pcatest.m.

2.1.1　实验过程描述

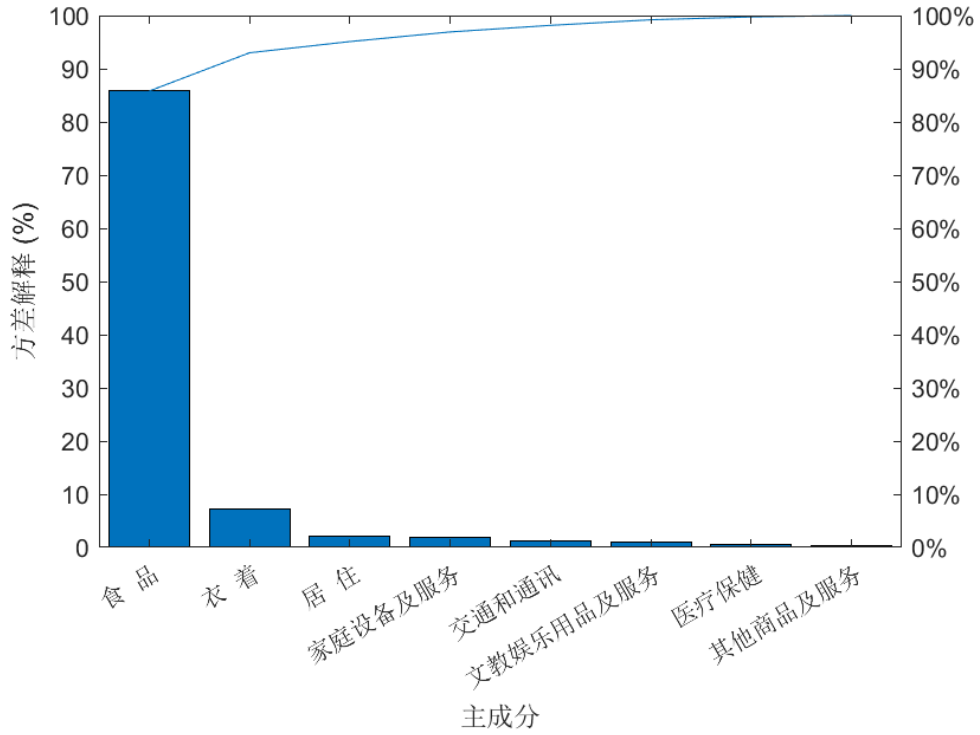
```
%% C2.1 Principal Component Analysis
% Author: Alephant
% Date: 22 Nov 2022
clc;
close all;
clear;


%% C2.1.0 load data
ratings = xlsread('data21.xls', "B4:I34");
[~, names] = xlsread('data21.xls', "A4:A34");
[~, categories] = xlsread('data21.xls', "B3:I3");


%% paramters
figpath = 'figures/';
```

```
figtype = '.png';


%% boxplot
figure
boxplot(ratings,'orientation','horizontal','labels',categories)
saveas(gcf, [figpath, 'C2.1-data-boxplot', figtype])


%% C2.1.1 pca
% normalization
stdr = std(ratings);
sr = ratings./repmat(stdr,31,1);

% pca
[coefs,scores,variances,t2] = pca(sr);

% var - Contribution rate of each component
percent_explained = 100*variances/sum(variances);
% plot pareto
figure
pareto(percent_explained, categories,1)
xlabel('主成分')
ylabel('方差解释 (%)')
saveas(gcf, [figpath, 'C2.1.1', figtype])


%% C2.1.2 ranking
figure
c1 = scores(:,1);
[c1_sorted,index] = sort(c1);
bar(c1_sorted);
set(gca, 'XTickLabel', names(index),'XTick',1:31)
saveas(gcf, [figpath, 'C2.1.2', figtype])



%% C2.1.3 scatter
% The first two principal component coefficients.
figure
plot(scores(:,1),scores(:,2),'x')
xlabel('第一主成分：食品');
ylabel('第二主成分：衣着');
% label the names of the regions
% gname(names);
saveas(gcf, [figpath, 'C2.1.3', figtype])
```

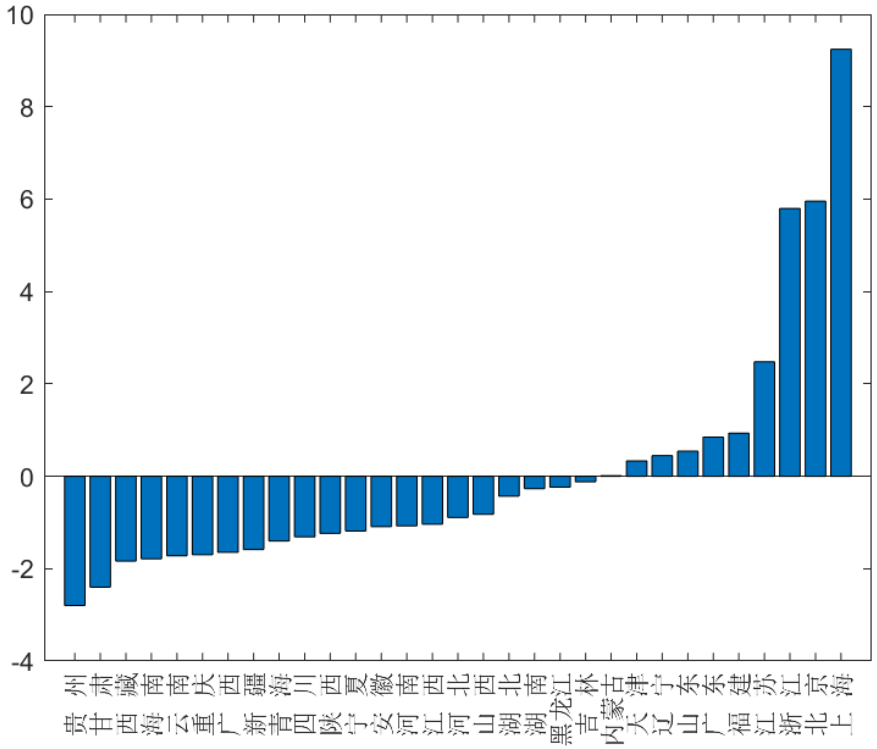
2.1.2 结果分析



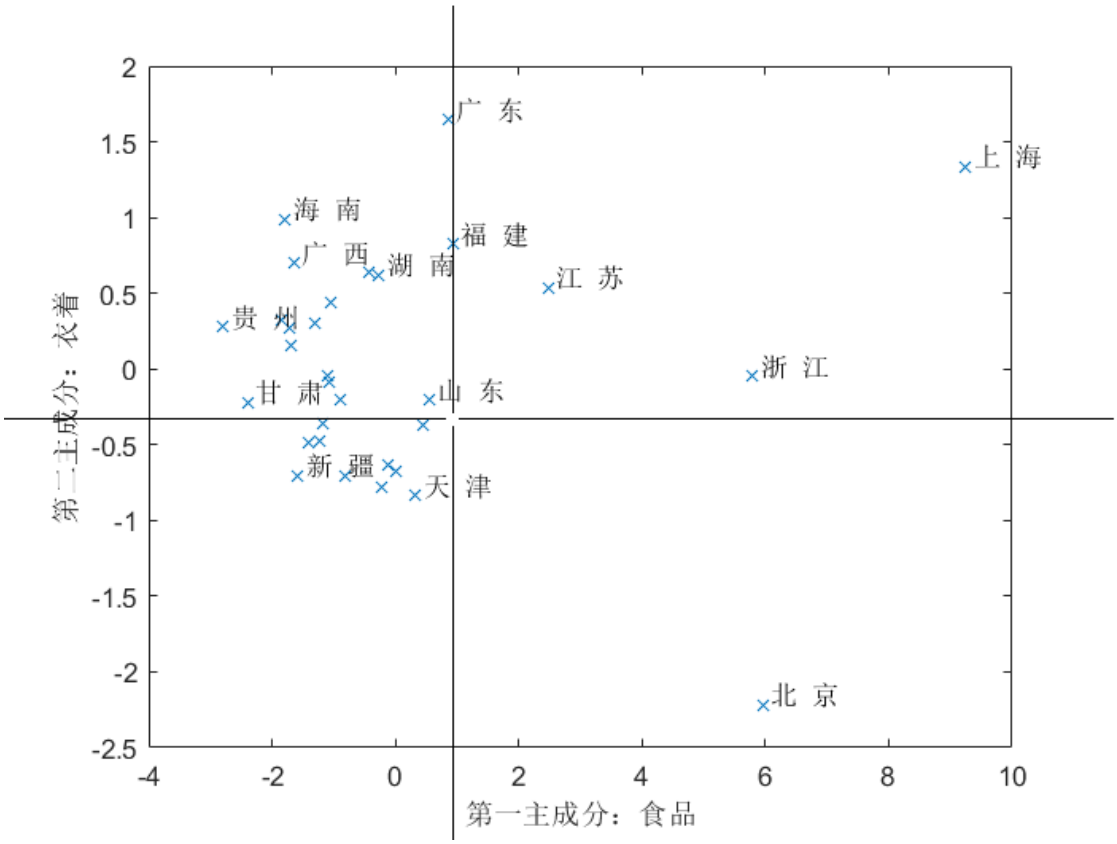
在多元统计分析中，主成分分析（英语：Principal components analysis，PCA）是一种统计分析、简化数据集的方法。它利用正交转换来对一系列可能相关的变量的观测值进行线性转换，从而投影为一系列线性不相关变量的值，这些不相关变量称为主成分（Principal Components）。具体地，主成分可以看做一个线性方程，其包含一系列线性系数来指示投影方向。PCA 对原始数据的正则化或预处理敏感（相对缩放）。



从上图可以看出**食品的方差解释度超过 85%，占绝大多数解释**；其次是衣着、居住等。



分析前两个主成分分别是食品和衣着，且食品方差解释性有着绝对优势。**按第一主成分得分将 31 个省、市、自治区和直辖市排序可以看出东部经济发达沿海城市得分较高；山区省份贵州、云南、四川等得分较低；另外中西部省份甘肃、西藏等得分也较低。**



前两个主成分得分的散点图，并在散点图上标注上各个地区的名称.

2.2　实验题目

（该题建议采用 matlab） 参考资料中 TrainDatabase 图像库作为训练图像进行训练，该图像库中有 10 个人，每人有两幅图像，每幅图像大小为**200×180**的。在 TestDatabase 中有 10 幅测试图像，其大小也为**200×180**的。

1）请利用 TrainDatabase 中的图像进行主成分分析，并将 TestDatabase 中的测试图像进行分类，选前三个最大的特征值。请显示特征脸，并给出分类依据；

2）若选前两个和前四个最大的特征值，结果如何，请列出相应的结果。

参考资料：pcaadd.doc 主成分分析人脸识别说明文档,示例程序等文件

2.2.1　实验过程描述

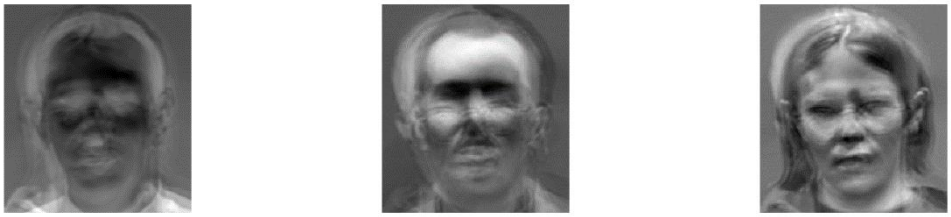
```
%% C2.2 Principal Component Analysis Figures
% Author: Alephant
% Date: 22 Nov 2022
clc;
close all;
clear;

%% paramters
figpath = 'figures/';
figtype = '.png';


%% Train
train_path = 'TrainDatabase\';
train_files = dir([train_path, '*.jpg']);
train_files_num = size(train_files);
for i= 1 : train_files_num
    train_file_name = [train_path, num2str(i), '.jpg'];
    train_file_image_read=imread(train_file_name);
    train_file_image_gray=rgb2gray(train_file_image_read);
    train_data(:,i)=double(train_file_image_gray(:));
end
% pca, the number of components
components_num = 4;
[eigenvectors,m,lambda] = cvpca(train_data,components_num);

feature_space_coordinates = zeros([components_num train_files_num], 'double');
for j = 1 : train_files_num
    for i = 1 : components_num
        feature_space_coordinates(i,j)=eigenvectors(:,i)'*train_data(:,j);
    end
end
```

```
% plot feature faces
for i = 1 : components_num
    e1=eigenvectors(:,i);
    feature_face=reshape(e1',200,180);
    figure;
    imshow(feature_face,[])

    % save figure
    figname_feature_face = ['C2.2--',num2str(components_num), ' components--feature
faces ', num2str(i)];
    saveas(gcf, [figpath, figname_feature_face, figtype])
end

%% Test
test_path = 'TestDatabase\';
test_files = dir([test_path, '*.jpg']);
test_file_num = size(test_files);

for i = 1:test_file_num
    test_file_name = [test_path, num2str(i), '.jpg'];
    test_file_image_read=imread(test_file_name);
    test_file_image_gray=rgb2gray(test_file_image_read);

    % projection of test file in feature space
    projection = eigenvectors' * double(test_file_image_gray(:));
    projections(:, i) = projection(:);

    % Projected coordinates to the distance of the 20 training files
    distance = dist(projection', feature_space_coordinates);
    distances(:, i) = distance(:);

    % Corresponds to the most recent training file
    position = find(distance==min(distance));
    positions(:,i) = position(:);

    % plot test and the result train
    figure
    subplot(1,2,1)
    imshow(test_file_name)
    title(['test ', num2str(i)])

    subplot(1,2,2)
    result_file_name = [train_path, num2str(position), '.jpg'];
    imshow([result_file_name])
```

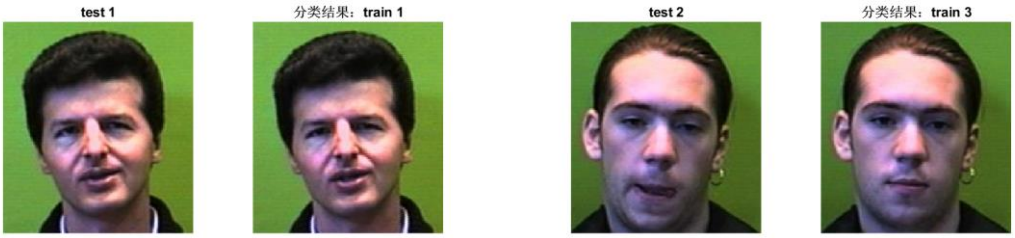
```
    title(['分类结果：train ', num2str(position)])

    % save figure
    figname_result = ['C2.2--',num2str(components_num), ' components--test ',
num2str(i)];
    saveas(gcf, [figpath, figname_result, figtype])
end
%% Classification basis
fprintf("选前%d 个最大的特征值\n",components_num)
disp("10 个测试文件，20 个训练文件")
disp("分类依据：")
disp("1. 训练文件在特征空间的投影：")
disp(feature_space_coordinates)
disp("------------------------------")
disp("2. 测试文件在特征空间的投影：")
disp(projections)
disp("------------------------------")
disp("3. 在特征空间中，测试文件投影（列）与训练文件投影（行）的距离：")
disp(distances)
disp("------------------------------")
disp("4. 距离最小训练文件序号：")
disp(positions)
```

2.2.2　结果分析

选前三个最大的特征值，显示特征脸：



对测试图像进行分类，左边显示测试图像，右边显示结果：




test 3
分类结果：train 5
test 4
分类结果：train 7


test 5
分类结果：train 9
test 6
分类结果：train 12


test 7
分类结果：train 13
test 8
分类结果：train 15


test 9
分类结果：train 5
test 10
分类结果：train 19

选前 3 个最大的特征值

10 个测试文件，20 个训练文件

分类依据：

1. 训练文件在特征空间的投影：

1.0e+04 *

Columns 1 through 8

-1.7385	-1.6843	-1.8671	-1.8982	-1.2477	-1.2104	-1.7748	-1.7468
-0.9553	-0.9635	0.0268	0.2523	-0.5811	-0.7047	-0.9605	-1.0459
-0.0793	-0.1066	-0.2295	-0.3599	-0.2661	-0.3940	-0.5209	-0.5006

Columns 9 through 16

-0.4228	-0.4243	-2.0317	-2.0203	-0.5129	-0.5102	-1.5909	-1.5651
-0.2514	-0.2546	0.2277	0.2078	-0.1731	-0.1762	-0.3980	-0.5522
-0.0205	-0.0264	-0.4433	-0.4337	-0.2320	-0.2312	-0.3351	-0.3560

Columns 17 through 20

-1.2181	-1.2169	-2.0277	-2.0070
-0.6763	-0.6877	-0.3483	-0.4684
-0.2804	-0.2700	0.4372	0.4734

---

2. 测试文件在特征空间的投影：

1.0e+04 *

Columns 1 through 8

-1.7442	-1.8631	-1.2623	-1.7980	-0.4330	-2.0363	-0.5254	-1.6016
-0.9574	-0.0035	-0.3626	-0.9113	-0.2522	0.1732	-0.1824	-0.4545
-0.0867	-0.2268	-0.2688	-0.5145	-0.0176	-0.4077	-0.2406	-0.3403

Columns 9 through 10

-1.2479	-2.0272

-0.5149　-0.3472
-0.3229　0.4286

---

3. 在特征空间中，测试文件投影（列）与训练文件投影（行）的距离：

1.0e+04 *

Columns 1 through 8

0.0096	0.9712	0.7836	0.4414	1.4841	1.2125	1.4474	0.5811
0.0634	0.9839	0.7520	0.4266	1.4421	1.2275	1.4040	0.5662
1.0021	0.0307	0.7204	0.9830	1.4763	0.2861	1.3580	0.5607
1.2498	0.2905	0.8893	1.1782	1.5870	0.1661	1.4449	0.7668
0.6483	0.8450	0.2190	0.6882	0.9131	1.1004	0.8255	0.3831
0.6657	0.9725	0.3680	0.6344	0.9751	1.2054	0.8750	0.4675
0.4353	1.0051	0.8269	0.0548	1.5986	1.1690	1.4984	0.5646
0.4232	1.0841	0.8691	0.1446	1.6091	1.2565	1.5182	0.6297
1.4996	1.4760	0.8824	1.6033	0.0106	1.7127	0.2524	1.2381
1.4965	1.4743	0.8789	1.5989	0.0127	1.7108	0.2476	1.2347
1.2706	0.3588	0.9854	1.1650	1.7226	0.0652	1.5743	0.8130
1.2467	0.3349	0.9629	1.1439	1.7042	0.0460	1.5570	0.7891
1.4670	1.3608	0.7738	1.5087	0.2421	1.5721	0.0177	1.1296
1.4676	1.3639	0.7757	1.5097	0.2396	1.5755	0.0189	1.1316
0.6309	0.4914	0.3371	0.5819	1.2095	0.7280	1.0912	0.0577
0.5184	0.6377	0.3678	0.4565	1.2191	0.8666	1.1095	0.1055
0.6271	0.9336	0.3170	0.6681	0.9302	1.1863	0.8517	0.4470
0.6200	0.9421	0.3282	0.6689	0.9316	1.1964	0.8570	0.4553
0.8520	0.7661	1.0414	1.1294	1.6610	0.9930	1.6564	0.8929
0.7887	0.8527	1.0567	1.1027	1.6629	1.0904	1.6694	0.9092

Columns 9 through 10

0.7028	0.8434
0.6621	0.8854
0.8279	0.7737
1.0064	0.9989

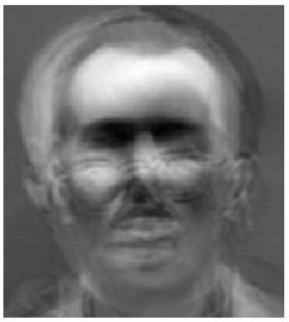
0.0873　1.0701
0.2062　1.2131
0.7179　1.1583
0.7499　1.1960
0.9174　1.6688
0.9132　1.6688
1.0864　1.0444
1.0635　1.0255
0.8157　1.6612
0.8169　1.6631
0.3625　0.8810
0.3210　0.9334
0.1696　1.1250
0.1834　1.1228
1.1016　0.0087
1.1011　0.1308

---

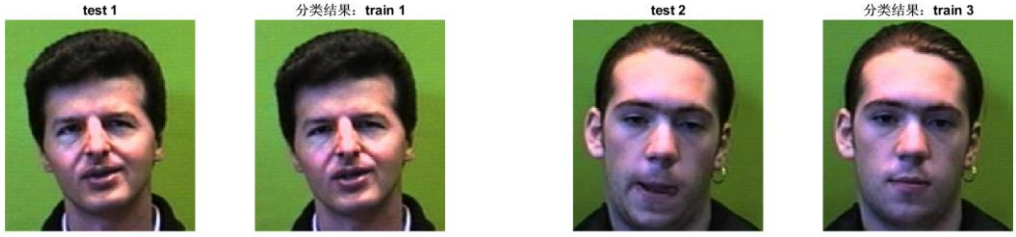
4. 距离最小训练文件序号：

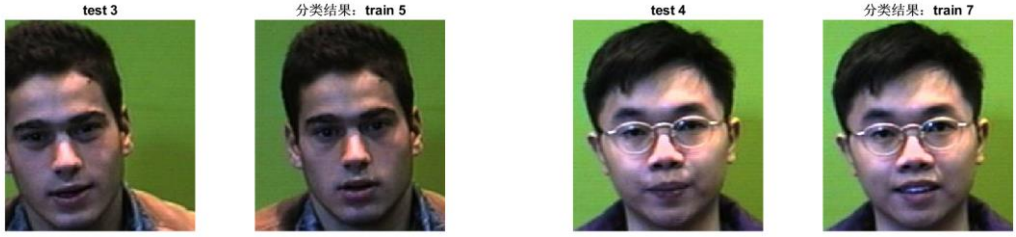
1　3　5　7　9　12　13　15　5　19

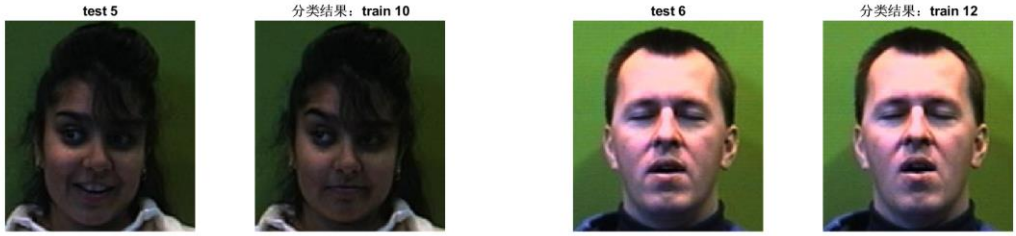
选前两个最大的特征值，显示特征脸：



对测试图像进行分类，左边显示测试图像，右边显示结果：


test 1
分类结果：train 1
test 2
分类结果：train 3


test 3
分类结果：train 5
test 4
分类结果：train 7


test 5
分类结果：train 10
test 6
分类结果：train 12


test 7
分类结果：train 13
test 8
分类结果：train 15

选前 2 个最大的特征值

10 个测试文件，20 个训练文件

分类依据：

1. 训练文件在特征空间的投影：

1.0e+04 *

Columns 1 through 8

-1.7385	-1.6843	-1.8671	-1.8982	-1.2477	-1.2104	-1.7748	-1.7468
-0.9553	-0.9635	0.0268	0.2523	-0.5811	-0.7047	-0.9605	-1.0459

Columns 9 through 16

-0.4228	-0.4243	-2.0317	-2.0203	-0.5129	-0.5102	-1.5909	-1.5651
-0.2514	-0.2546	0.2277	0.2078	-0.1731	-0.1762	-0.3980	-0.5522

Columns 17 through 20

-1.2181	-1.2169	-2.0277	-2.0070
-0.6763	-0.6877	-0.3483	-0.4684

---

2. 测试文件在特征空间的投影：

1.0e+04 *

Columns 1 through 8

-1.7442	-1.8631	-1.2623	-1.7980	-0.4330	-2.0363	-0.5254	-1.6016

-0.9574　-0.0035　-0.3626　-0.9113　-0.2522　0.1732　-0.1824　-0.4545

Columns 9 through 10

-1.2479　-2.0272
-0.5149　-0.3472

---

3. 在特征空间中，测试文件投影（列）与训练文件投影（行）的距离：

1.0e+04 *

Columns 1 through 8

0.0061	0.9600	0.7603	0.0740	1.4828	1.1672	1.4384	0.5192
0.0602	0.9765	0.7343	0.1251	1.4393	1.1900	1.3976	0.5157
0.9919	0.0306	0.7194	0.9407	1.4610	0.2237	1.3579	0.5497
1.2195	0.2582	0.8846	1.1680	1.5496	0.1591	1.4400	0.7665
0.6230	0.8441	0.2190	0.6418	0.8786	1.0913	0.8251	0.3759
0.5906	0.9581	0.3461	0.6229	0.8995	1.2054	0.8614	0.4644
0.0308	0.9611	0.7875	0.0544	1.5173	1.1635	1.4719	0.5349
0.0885	1.0489	0.8377	0.1440	1.5349	1.2530	1.4958	0.6090
1.4981	1.4615	0.8468	1.5253	0.0102	1.6684	0.1236	1.1961
1.4953	1.4606	0.8449	1.5226	0.0090	1.6678	0.1242	1.1941
1.2195	0.2861	0.9698	1.1628	1.6692	0.0547	1.5612	0.8065
1.1975	0.2633	0.9487	1.1410	1.6526	0.0380	1.5450	0.7835
1.4598	1.3608	0.7729	1.4820	0.1124	1.5622	0.0155	1.1244
1.4605	1.3639	0.7748	1.4829	0.1083	1.5656	0.0164	1.1263
0.5800	0.4794	0.3306	0.5535	1.1671	0.7243	1.0871	0.0574
0.4430	0.6245	0.3573	0.4281	1.1712	0.8650	1.1035	0.1043
0.5965	0.9320	0.3168	0.6257	0.8923	1.1794	0.8508	0.4430
0.5923	0.9411	0.3282	0.6227	0.8967	1.1885	0.8565	0.4499
0.6718	0.3821	0.7655	0.6081	1.5976	0.5216	1.5114	0.4391
0.5552	0.4867	0.7523	0.4898	1.5888	0.6423	1.5090	0.4057

Columns 9 through 10

0.6593　0.6732
0.6258　0.7053
0.8227　0.4068
1.0057　0.6132
0.0663　0.8139
0.1936　0.8917
0.6901　0.6633
0.7286　0.7529
0.8662　1.6072
0.8638　1.6056
1.0797　0.5749
1.0577　0.5550
0.8106　1.5243
0.8118　1.5266
0.3623　0.4393
0.3193　0.5056
0.1641　0.8735
0.1756　0.8790
0.7973　0.0012
0.7605　0.1229

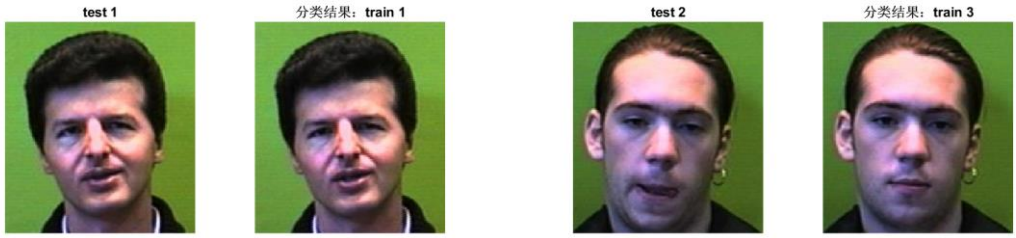
---

4. 距离最小训练文件序号：

1　3　5　7　10　12　13　15　5　19

选前四个最大的特征值，显示特征脸：

对测试图像进行分类，左边显示测试图像，右边显示结果：



选前 4 个最大的特征值

10 个测试文件，20 个训练文件

分类依据：

1. 训练文件在特征空间的投影：

1.0e+04 *

Columns 1 through 8

-1.7385	-1.6843	-1.8671	-1.8982	-1.2477	-1.2104	-1.7748	-1.7468
-0.9553	-0.9635	0.0268	0.2523	-0.5811	-0.7047	-0.9605	-1.0459
-0.0793	-0.1066	-0.2295	-0.3599	-0.2661	-0.3940	-0.5209	-0.5006
-0.6318	-0.7185	-0.1587	-0.2176	-0.1960	-0.0719	-0.1420	-0.2402

Columns 9 through 16

-0.4228	-0.4243	-2.0317	-2.0203	-0.5129	-0.5102	-1.5909	-1.5651
-0.2514	-0.2546	0.2277	0.2078	-0.1731	-0.1762	-0.3980	-0.5522
-0.0205	-0.0264	-0.4433	-0.4337	-0.2320	-0.2312	-0.3351	-0.3560
-0.1340	-0.1341	-0.4250	-0.4271	-0.5344	-0.5256	-0.0548	-0.0325

Columns 17 through 20

-1.2181	-1.2169	-2.0277	-2.0070
-0.6763	-0.6877	-0.3483	-0.4684
-0.2804	-0.2700	0.4372	0.4734
-0.1650	-0.1631	-0.2148	-0.1248

---

2. 测试文件在特征空间的投影：

1.0e+04 *

Columns 1 through 8

-1.7442	-1.8631	-1.2623	-1.7980	-0.4330	-2.0363	-0.5254	-1.6016
-0.9574	-0.0035	-0.3626	-0.9113	-0.2522	0.1732	-0.1824	-0.4545
-0.0867	-0.2268	-0.2688	-0.5145	-0.0176	-0.4077	-0.2406	-0.3403
-0.6292	-0.1762	-0.1882	-0.1331	-0.1333	-0.4282	-0.4914	-0.0399

Columns 9 through 10

-1.2479	-2.0272
-0.5149	-0.3472
-0.3229	0.4286
-0.1407	-0.1924

---

3. 在特征空间中，测试文件投影（列）与训练文件投影（行）的距离：

1.0e+04 *

Columns 1 through 8

0.0099	1.0728	0.9004	0.6660	1.5656	1.2295	1.4542	0.8295
0.1095	1.1235	0.9201	0.7244	1.5563	1.2613	1.4222	0.8838
1.1071	0.0353	0.7210	0.9833	1.4765	0.3930	1.3981	0.5732
1.3158	0.2935	0.8898	1.1812	1.5892	0.2682	1.4707	0.7871
0.7797	0.8452	0.2192	0.6910	0.9152	1.1247	0.8767	0.4137
0.8682	0.9781	0.3860	0.6374	0.9770	1.2570	0.9704	0.4686
0.6533	1.0057	0.8282	0.0555	1.5986	1.2035	1.5386	0.5737
0.5748	1.0859	0.8707	0.1800	1.6127	1.2705	1.5389	0.6608
1.5792	1.4766	0.8841	1.6033	0.0107	1.7378	0.4376	1.2417
1.5763	1.4749	0.8806	1.5989	0.0127	1.7359	0.4347	1.2383
1.2869	0.4367	1.0134	1.2010	1.7471	0.0653	1.5757	0.8996
1.2630	0.4185	0.9921	1.1810	1.7293	0.0461	1.5584	0.8790

1.4701	1.4072	0.8477	1.5612	0.4685	1.5756	0.0465	1.2331
1.4713	1.4080	0.8459	1.5599	0.4597	1.5785	0.0390	1.2314
0.8532	0.5062	0.3626	0.5871	1.2120	0.8182	1.1753	0.0596
0.7904	0.6537	0.3994	0.4674	1.2232	0.9526	1.2007	0.1058
0.7802	0.9337	0.3178	0.6688	0.9308	1.2151	0.9121	0.4642
0.7757	0.9422	0.3292	0.6696	0.9321	1.2255	0.9177	0.4717
0.9475	0.7671	1.0417	1.1323	1.6630	1.0157	1.6794	0.9099
0.9362	0.8543	1.0586	1.1027	1.6630	1.1318	1.7092	0.9132

Columns 9 through 10

0.8574	0.9509
0.8788	1.0299
0.8281	0.7744
1.0093	0.9993
0.1033	1.0701
0.2174	1.2191
0.7179	1.1594
0.7565	1.1969
0.9174	1.6698
0.9133	1.6698
1.1230	1.0700
1.1014	1.0520
0.9057	1.6961
0.9031	1.6962
0.3725	0.8917
0.3388	0.9470
0.1713	1.1253
0.1847	1.1232
1.1040	0.0240
1.1012	0.1472

---

4. 距离最小训练文件序号：

1　3　5　7　9　12　14　15　5　19

3　C 均值聚类

实验目的：掌握利用 C 均值聚类分析的方法。

3.1　实验题目

对习题 5.5 中的鸢尾属植物花的形状数据(见表 5.10)的 150 个样品,利用欧氏距离作如下快速聚类分析.其中　$X=(X_1,X_2,X_3,X_4)^{\mathbf{T}}$ 分别表示花的萼片长、萼片宽、花瓣长、花瓣宽 4 个变量.各种方法均聚分 3 类.

1）用 $X=(X_2,X_4)^{\mathbf{T}}$ 二个变量聚类;

2）用 $X=(X_1,X_2,X_3)^{\mathbf{T}}$ 三个变量聚类;

3）用 $X=(X_1,X_2,X_3,X_4)^{\mathbf{T}}$ 四个变量聚类;

4）将以上各情况下的聚类结果与数据集中的实际分类情况比较,是否所用变量越多，聚类效果就越好？进一步讨论其他一些变量组合下聚为 3 类的情况,以支持你的观点.

表 5.10　鸢尾属植物三个不同品种的花的形状数据

编号	品种	x_1	x_2	x_3	x_4	编号	品种	x_1	x_2	x_3	x_4	编号	品种	x_1	x_2	x_3	x_4
1	1	50	33	14	2	51	2	65	28	46	15	101	3	64	28	56	22
2	1	46	34	14	3	52	2	62	22	45	15	102	3	67	31	56	24
3	1	46	36	10	2	53	2	59	32	48	18	103	3	63	28	51	15
4	1	51	33	17	5	54	2	61	30	46	14	104	3	69	31	51	23
5	1	55	35	13	2	55	2	60	27	51	16	105	3	65	30	52	20
6	1	48	31	16	2	56	2	56	25	39	11	106	3	65	30	55	18
7	1	52	34	14	2	57	2	57	28	45	13	107	3	58	27	51	19
8	1	49	36	14	1	58	2	63	33	47	16	108	3	68	32	59	23
9	1	44	32	13	2	59	2	70	32	47	14	109	3	62	34	54	23
10	1	50	35	16	6	60	2	64	32	45	15	110	3	77	38	67	22
11	1	44	30	13	2	61	2	61	28	40	13	111	3	67	33	57	25
12	1	47	32	16	2	62	2	55	24	38	11	112	3	76	30	66	21
13	1	48	30	14	3	63	2	54	30	45	15	113	3	49	25	45	17
14	1	51	38	16	2	64	2	58	26	40	12	114	3	67	30	52	23
15	1	48	34	19	2	65	2	55	26	44	12	115	3	59	30	51	18
16	1	50	30	16	2	66	2	50	23	33	10	116	3	63	25	50	19
17	1	50	32	12	2	67	2	67	31	44	14	117	3	64	32	53	23
18	1	43	30	11	1	68	2	56	30	45	15	118	3	79	38	64	20
19	1	58	40	12	2	69	2	58	27	41	10	119	3	67	33	57	21
20	1	51	38	19	4	70	2	60	29	45	15	120	3	77	28	67	20
21	1	49	30	14	2	71	2	57	26	35	10	121	3	63	27	49	18
22	1	51	35	14	2	72	2	57	19	42	13	122	3	72	32	60	18
23	1	50	34	16	4	73	2	49	24	33	10	123	3	61	30	49	18
24	1	46	32	14	2	74	2	56	27	42	13	124	3	61	26	56	14
25	1	57	44	15	4	75	2	57	30	42	12	125	3	64	28	56	21
26	1	50	36	14	2	76	2	66	29	46	13	126	3	62	28	48	18
27	1	54	34	15	4	77	2	52	27	39	14	127	3	77	30	61	23
28	1	52	42	15	1	78	2	60	34	45	16	128	3	63	34	56	24
29	1	55	42	14	2	79	2	50	20	35	10	129	3	58	27	51	19
30	1	49	31	15	2	80	2	55	24	37	10	130	3	72	30	58	16
31	1	54	39	17	4	81	2	58	27	39	12	131	3	71	30	59	21
32	1	50	34	15	2	82	2	62	29	43	13	132	3	64	31	55	18
33	1	44	29	14	2	83	2	59	30	42	15	133	3	60	30	48	18
34	1	47	32	13	2	84	2	60	22	40	10	134	3	63	29	56	18
35	1	46	31	15	2	85	2	67	31	47	15	135	3	77	26	69	23
36	1	51	34	15	2	86	2	63	23	44	13	136	3	60	22	50	15
37	1	50	35	13	3	87	2	56	30	41	13	137	3	69	32	57	23
38	1	49	31	15	1	88	2	63	25	49	15	138	3	74	28	61	19
39	1	54	37	15	2	89	2	61	28	47	12	139	3	56	28	49	20
40	1	54	39	13	4	90	2	64	29	43	13	140	3	73	29	63	18
41	1	51	35	14	3	91	2	51	25	30	11	141	3	67	25	58	18
42	1	48	34	16	2	92	2	57	28	41	13	142	3	65	30	58	22
43	1	48	30	14	1	93	2	61	29	47	14	143	3	69	31	54	21
44	1	45	23	13	3	94	2	56	29	36	13	144	3	72	36	61	25
45	1	57	38	17	3	95	2	69	31	49	15	145	3	65	32	51	20
46	1	51	38	15	3	96	2	55	25	40	13	146	3	64	27	53	19
47	1	54	34	17	2	97	2	55	23	40	13	147	3	68	30	55	21
48	1	51	37	15	4	98	2	66	30	44	14	148	3	57	25	50	20
49	1	52	35	15	2	99	2	68	28	48	14	149	3	58	28	51	24
50	1	53	37	15	2	100	2	67	30	50	17	150	3	63	33	60	25

3.1.1　实验过程描述

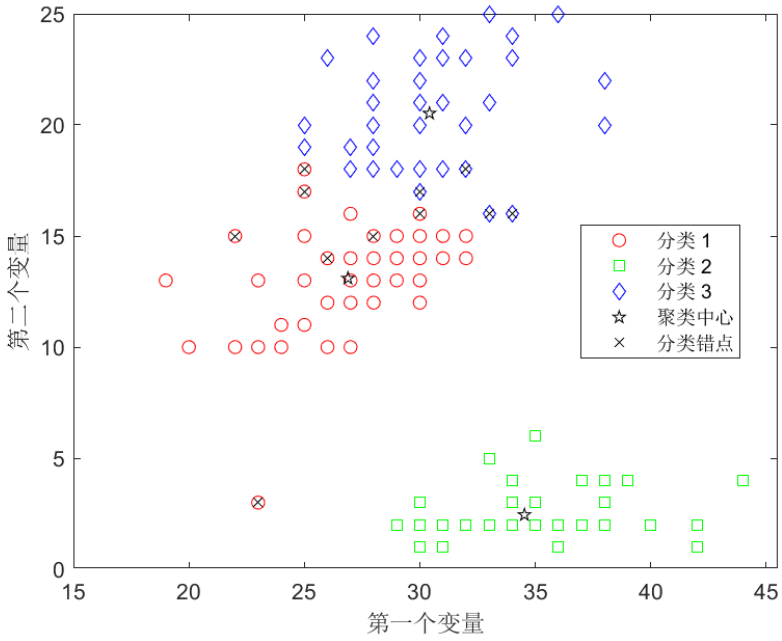
```
%% C3.1 Use Euclidean distance to do the following fast cluster analysis
% Author: Alephant
% Date: 30 Nov 2022
clc;
close all;
clear;


%% load data
raw_data = xlsread("data31.xls", 'C2:F151');
category = xlsread("data31.xls", 'B2:B151');
k = 3; % number of cluster categories


%% paramters
figpath = 'figures/';
figtype = '.png';


%% C3.1.1
% classify
X = raw_data(:, [2, 4]);
[idx, C] = kmeans(X, k);

% Look for misclassification and false positive rate
% This data takes 50 as a class
% and assumes that the classification is roughly accurate
% so it can be calculated with the mode
seg_result = [idx(1:50), idx(51:100), idx(101:150)];
[M, F] = mode(seg_result);
bad_index = [idx(1:50) ~= M(1); idx(51:100) ~= M(2); idx(101:150) ~= M(3)];
false_rate = sum(bad_index)/150 ;

% test hypothesis
disp("若分类数是各一段 50 个中的众数，则假设成立，可以用该方法找分类错点。")
disp([M;F])
disp('变量：X=\left(X_2,X_4\right)^\mathbf{T}')
disp(['分类错误数：', num2str(sum(bad_index))])
disp(['分类错误率：', num2str(false_rate * 100),'%'])
disp('--------');

% plot
figure
gscatter(X(:,1),X(:,2),idx,'rgb','osd')
```

```
hold on
plot(C(:,1),C(:,2),'kp')
plot(X(bad_index,1), X(bad_index,2), 'kx')
legend('分类 1', '分类 2', '分类 3', '聚类中心', '分类错点')
xlabel('第一个变量')
ylabel('第二个变量')
saveas(gcf, [figpath, 'C3.1.1', figtype])


%% C3.1.2
% classify
X = raw_data(:, [1, 2, 3]);
[idx, C] = kmeans(X, k);

% Look for misclassification and false positive rate
% This data takes 50 as a class
% and assumes that the classification is roughly accurate
% so it can be calculated with the mode
seg_result = [idx(1:50), idx(51:100), idx(101:150)];
[M, F] = mode(seg_result);
bad_index = [idx(1:50) ~= M(1); idx(51:100) ~= M(2); idx(101:150) ~= M(3)];
false_rate = sum(bad_index)/150 ;

% test hypothesis
disp("若分类数是各一段 50 个中的众数，则假设成立，可以用该方法找分类错点。")
disp([M;F])
disp('变量：X=\left(X_1,X_2,X_3\right)^\mathbf{T}')
disp(['分类错误数：', num2str(sum(bad_index))])
disp(['分类错误率：', num2str(false_rate * 100),'%'])
disp('--------');

% plot
figure
gscatter(X(:,1),X(:,2),idx,'rgb','osd')
hold on
plot(C(:,1),C(:,2),'kp')
plot(X(bad_index,1), X(bad_index,2), 'kx')
legend('分类 1', '分类 2', '分类 3', '聚类中心', '分类错点')
xlabel('第一个变量')
ylabel('第二个变量')
saveas(gcf, [figpath, 'C3.1.2', figtype])


%% C3.1.3
% classify
X = raw_data;
```

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

% Look for misclassification and false positive rate
% This data takes 50 as a class
% and assumes that the classification is roughly accurate
% so it can be calculated with the mode
seg_result = [idx(1:50), idx(51:100), idx(101:150)];
[M, F] = mode(seg_result);
bad_index = [idx(1:50) ~= M(1); idx(51:100) ~= M(2); idx(101:150) ~= M(3)];
false_rate = sum(bad_index)/150 ;

% test hypothesis
disp("若分类数是各一段 50 个中的众数，则假设成立，可以用该方法找分类错点。")
disp([M;F])
disp('变量：X=\left(X_1,X_2,X_3,X_4\right)^\mathbf{T}')
disp(['分类错误数：', num2str(sum(bad_index))])
disp(['分类错误率：', num2str(false_rate * 100),'%'])
disp('--------');

% plot
figure
gscatter(X(:,1),X(:,2),idx,'rgb','osd')
hold on
plot(C(:,1),C(:,2),'kp')
plot(X(bad_index,1), X(bad_index,2), 'kx')
legend('分类 1', '分类 2', '分类 3', '聚类中心', '分类错点')
xlabel('第一个变量')
ylabel('第二个变量')
saveas(gcf, [figpath, 'C3.1.3', figtype])


%% C3.1.4-1
% classify
X = raw_data;
[idx, C] = kmeans(X(:,[1,2,4]), k);

seg_result = [idx(1:50), idx(51:100), idx(101:150)];
[M, F] = mode(seg_result);
bad_index = [idx(1:50) ~= M(1); idx(51:100) ~= M(2); idx(101:150) ~= M(3)];
false_rate = sum(bad_index)/150 ;

% test hypothesis
disp("若分类数是各一段 50 个中的众数，则假设成立，可以用该方法找分类错点。")
disp([M;F])
disp('变量：X=\left(X_1,X_2,X_4\right)^\mathbf{T}')
```

```
disp(['分类错误数：', num2str(sum(bad_index))])
disp(['分类错误率：', num2str(false_rate * 100),'%'])
disp('--------');

% plot
figure
gscatter(X(:,1),X(:,2),idx,'rgb','osd')
hold on
plot(C(:,1),C(:,2),'kp')
plot(X(bad_index,1), X(bad_index,2), 'kx')
legend('分类 1', '分类 2', '分类 3', '聚类中心', '分类错点')
xlabel('第一个变量')
ylabel('第二个变量')
saveas(gcf, [figpath, 'C3.1.4-1', figtype])


%% C3.1.4-2
% classify
X = raw_data;
[idx, C] = kmeans(X(:,[2,3,4]), k);

seg_result = [idx(1:50), idx(51:100), idx(101:150)];
[M, F] = mode(seg_result);
bad_index = [idx(1:50) ~= M(1); idx(51:100) ~= M(2); idx(101:150) ~= M(3)];
false_rate = sum(bad_index)/150 ;

% test hypothesis
disp("若分类数是各一段 50 个中的众数，则假设成立，可以用该方法找分类错点。")
disp([M;F])
disp('变量：X=\left(X_2,X_3,X_4\right)^\mathbf{T}')
disp(['分类错误数：', num2str(sum(bad_index))])
disp(['分类错误率：', num2str(false_rate * 100),'%'])
disp('--------');

% plot
figure
gscatter(X(:,1),X(:,2),idx,'rgb','osd')
hold on
plot(C(:,1),C(:,2),'kp')
plot(X(bad_index,1), X(bad_index,2), 'kx')
legend('分类 1', '分类 2', '分类 3', '聚类中心', '分类错点')
xlabel('第一个变量')
ylabel('第二个变量')
saveas(gcf, [figpath, 'C3.1.4-2', figtype])
```

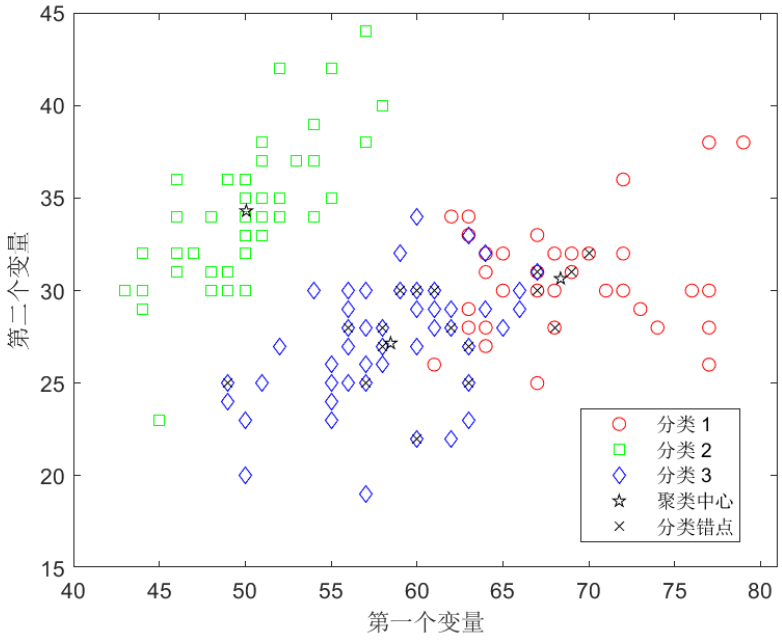
3.1.2　结果分析



$X=(X_2,X_4)^{\mathrm{T}}$两个变量聚类

分类错误数：11

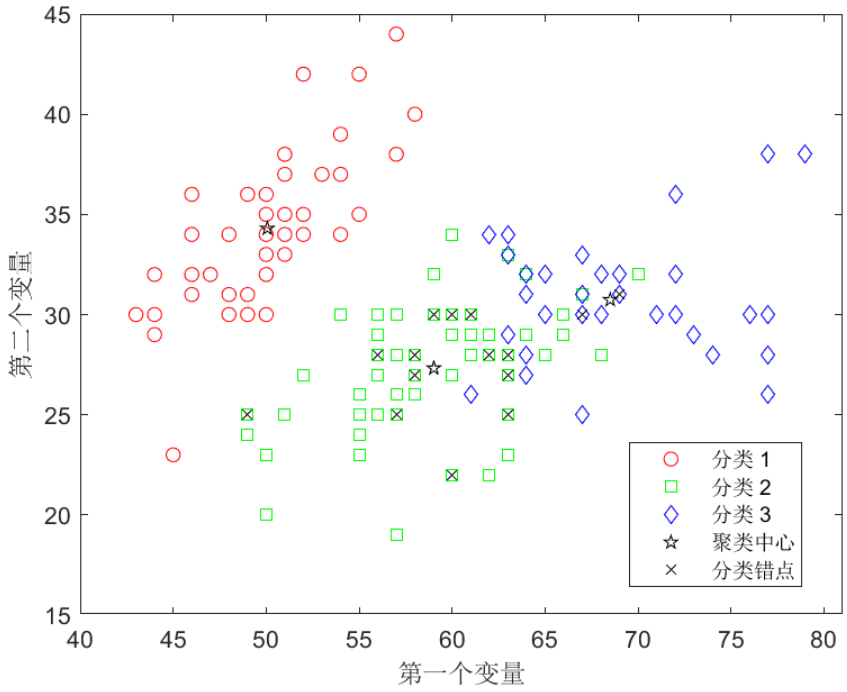
分类错误率：7.3333%



$X=(X_1,X_2,X_3)^{\mathrm{T}}$三个变量聚类

分类错误数：18

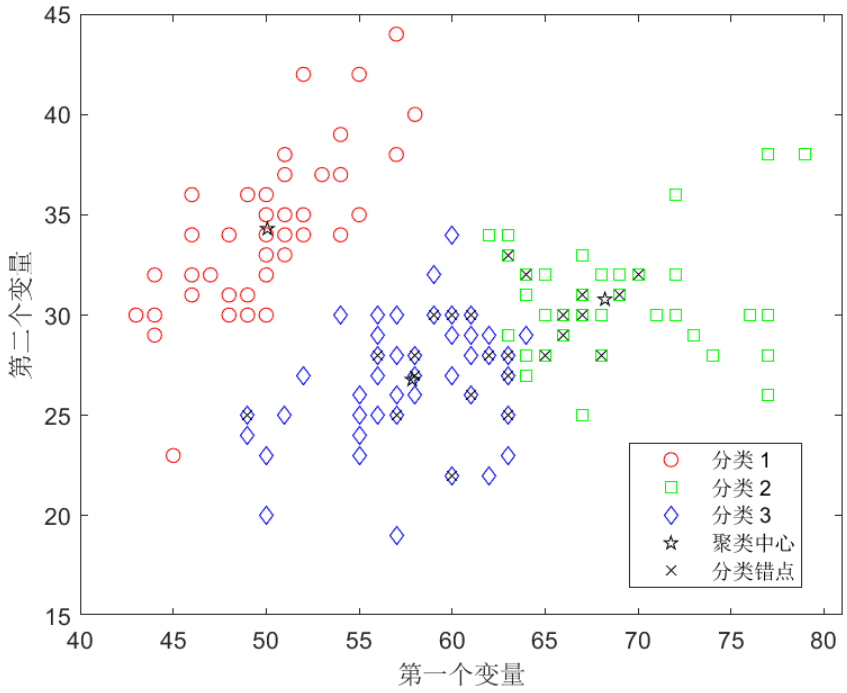
分类错误率：12%



$X=(X_1,X_2,X_3,X_4)^{\mathrm{T}}$ 四个变量聚类

分类错误数：16

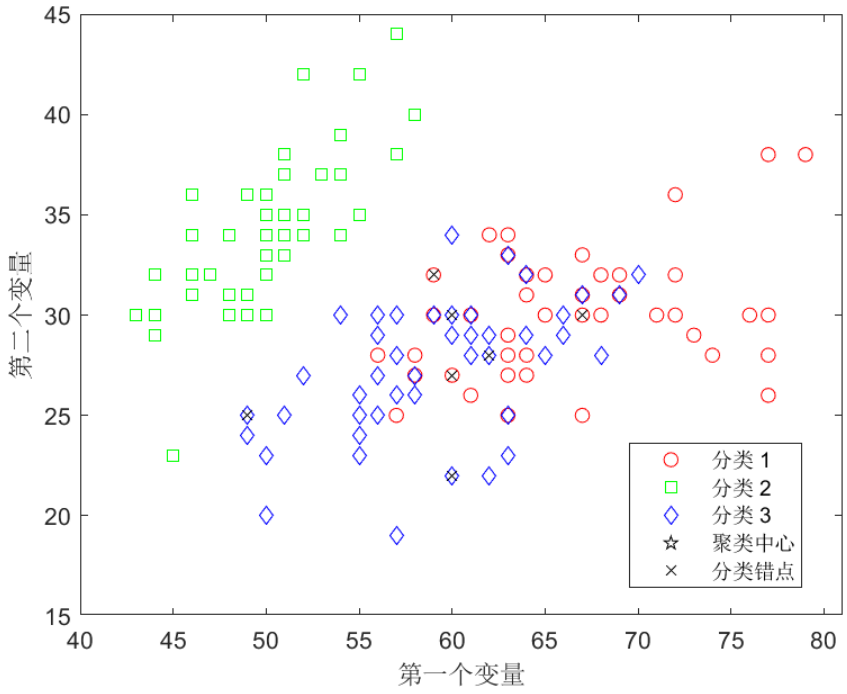
分类错误率：10.6667%



$X=(X_1,X_2,X_4)^{\mathrm{T}}$ 三个变量聚类

分类错误数：26

分类错误率：17.3333%



$X=(X_2,X_3,X_4)^{\mathrm{T}}$ 三个变量聚类

分类错误数：7

分类错误率：4.6667%

K-Means 聚类步骤是一个循环迭代的算法，非常简单易懂：

1. 假定我们要对 N 个样本观测做聚类，要求聚为 K 类，首先选择 K 个点作为**初始中心点**；
2. 接下来，按照**距离初始中心点最小**的原则，把所有观测分到各中心点所在的类中；
3. 每类中有若干个观测，计算 K 个类中**所有样本点的均值**，作为第二次迭代的 K 个中心点；
4. 然后根据这个中心重复第 2、3 步，直到**收敛（中心点不再改变或达到指定的迭代次数）**，聚类过程结束。

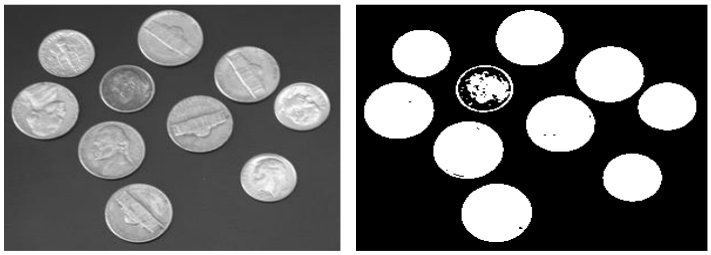
以上各情况下的聚类结果与数据集中的实际分类情况比较所用变量越多，并非聚类效果就越好，从前三张图来看：选取两个变量效果最好，四个变量的次之，三个变量的效果最差；但变量的选取也起着关键作用，分析都是选取三个变量的聚类图：选取 $X=(X_2,X_3,X_4)^{\mathrm{T}}$ 三个变量聚类时分类错误率最低，$X=(X_1,X_2,X_4)^{\mathrm{T}}$ 三个变量的分类错误率最高。

这说明聚类效果不光与所用变量数量有关，与选取的变量也有关系。

3.2　实验题目

（该题建议采用matlab）自己拍一张照片，利用C均值算法进行图像的分割和向量量化（分类数自选）.

参考资料：C均值图像分割（examp_seg.m，　examp_quantity.m）



(a) 原图　　(b) 分割图

图 1，利用 C 均值算法进行图像分割

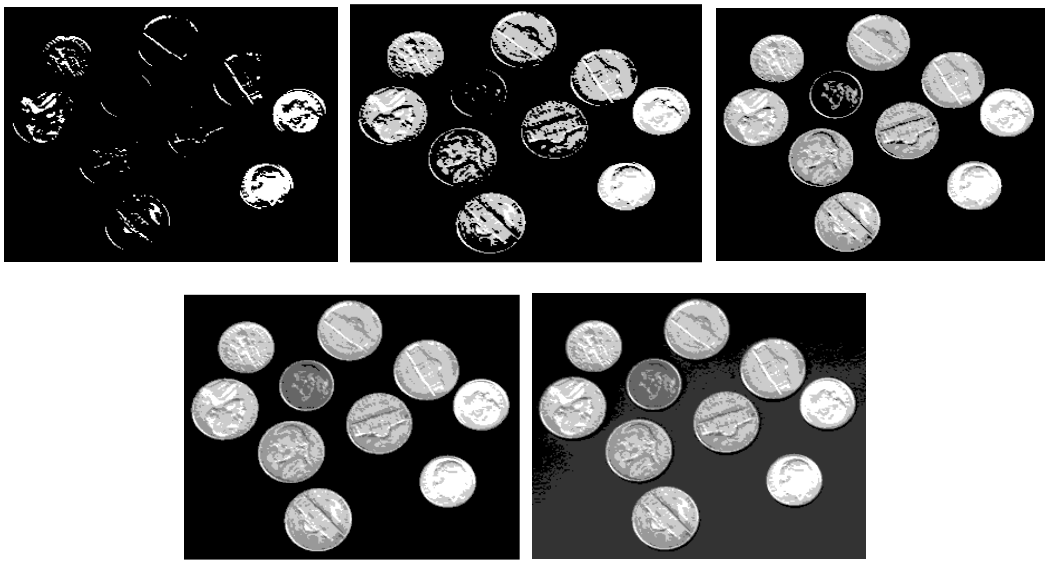


图 2，利用 C 均值算法进行图像向量量化

3.2.1　实验过程描述

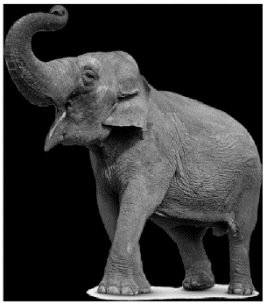
```
%% C3.2.1 C-means image segmentation
% Author: Alephant
% Date: 22 Nov 2022
clc;
close all;
clear;


%% paramters
figpath = 'figures/';
```

```
figtype = '.png';


%% raw png
raw_png = imread('elephant.png');
x = rgb2gray(raw_png);
figure,
imshow(x);
y = double(x(:));
saveas(gcf, [figpath, 'C3.2.1-raw', figtype])


%% segmentation
startdata1 = [0;150];
idpixel = kmeans(y,2,'Start',startdata1);
idbw = (idpixel == 2);%%%%只选标号为 2 的
seg = reshape(idbw, size(x));
figure,
imshow(seg);
saveas(gcf, [figpath, 'C3.2.1-seg', figtype])
```

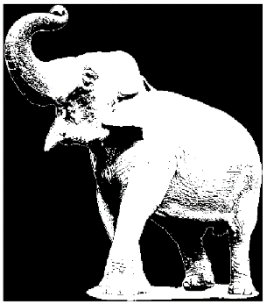
```
%% C3.2.2 Vector quantization with C-means
% Author: Alephant
% Date: 22 Nov 2022
clc;
close all;
clear;


%% paramters
figpath = 'figures/';
figtype = '.png';


%% raw png
raw_png = imread('elephant.png');
x = rgb2gray(raw_png);
figure(1),
imshow(x);
saveas(gcf, [figpath, 'C3.2.2-raw', figtype]) ;
y = double(x(:));


%% Vector quantization
```

```
startdata2 = [0;50;100;150;200;250];
idpixelq = kmeans(y,6,'Start',startdata2);
idq6 = (idpixelq == 6);
idq5 = (idpixelq == 5);
idq4 = (idpixelq == 4);
idq3 = (idpixelq == 3);
idq2 = (idpixelq == 2);
idq1 = (idpixelq == 1);

q5 = idq2.*50+ idq3.*100+ idq4.*150+ idq5.*200+ idq6.*250;
qresult5 = reshape(q5, size(x));
figure(2), imshow(qresult5/250);
saveas(gcf, [figpath, 'C3.2.2-5', figtype]) ;

q4 =  idq3.*100+ idq4.*150+ idq5.*200+ idq6.*250;
qresult = reshape(q4, size(x));
figure(3), imshow(qresult/250);
saveas(gcf, [figpath, 'C3.2.2-4', figtype]) ;

q3 =   idq4.*150+ idq5.*200+ idq6.*250;
qresult = reshape(q3, size(x));
figure(4), imshow(qresult/250);
saveas(gcf, [figpath, 'C3.2.2-3', figtype]) ;

q2 =  idq5.*200+ idq6.*250;
qresult = reshape(q2, size(x));
figure(5), imshow(qresult/250);
saveas(gcf, [figpath, 'C3.2.2-2', figtype]) ;

q1 =  idq6.*250;
qresult = reshape(q1, size(x));
figure(6), imshow(qresult/250);
saveas(gcf, [figpath, 'C3.2.2-1', figtype]) ;
```

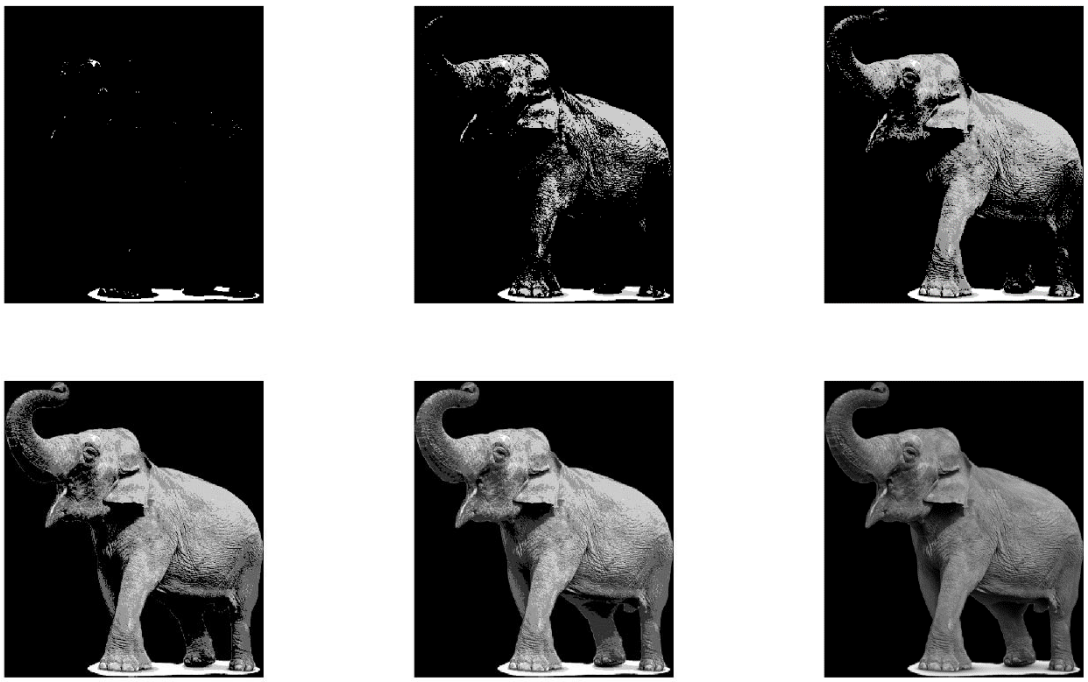
3.2.2　结果分析



(a) 原图



(b) 分割图



利用 C 均值算法进行图像向量量化　最后一张是原图

图像分割就是把图像细分为构成它的对象或子区域，这些区域是互不相交的，每个区域都满足特定区域的一致性。分割的程度主要取决于人们想要解决的问题，当感兴趣的区域或对象已经被区分出来，分割就算完成。图像分割是图像处理中的重要问题，也是计算机视觉研究中的一个经典难题。计算机视觉中的图像理解包括目标检测、特征提取和目标识别等，都依赖于分割的质量。

目前，图像分割算法一般是围绕亮度值的两个基本特性设计的：不连续性和相似性。亮度值的不连续性的应用途径主要是基于像素点特性（如灰度值）的不连续变化分割图像，如最常用的边缘检测。而利用亮度值的相似性可以形成一套机制，即依据事先指定的准则将图像分割为相似的区域。一些实例包括门限处理、区域分离、区域生长和聚类等。而采用模糊 C 均值聚类及其扩展算法进行图像分割的好处是避免了阈值的设定问题，聚类的过程不需要人工干预，只需输入预想的分类数目即可实现自动化的图像分割。

声明：所有代码均由本人强盛周参考老师资料和网络资源独立完成，代码等所有源文件开源在 Github 仓库：https://github.com/Alephant6/Data-Analysis-Course-Design 和国内 Gitee 镜像：https://gitee.com/qiang-shengzhou/Data-Analysis-Course-Design