

**模式识别实验报告**

**学 号:1817000346**

**姓 名：杨露瑶**

**老 师：马家君**

**班 级：大数据181**

**专 业：数据科学与大数据技术**

**学 院：大数据与信息工程学院**

2021年 6 月 18 日

# 实验一、图像的读取

## 实验目的

1. 熟悉并掌握MATLAB软件的使用。
2. 简单实现图像的读取、显示、代数运算和简单变换。

## 实验仪器设备及软件

计算机一台

MATLAB软件

## 实验原理

1. MATLAB中读写图像文件的函数：
2. Imread()

/Imread函数用于读取多种类型的图像文件。/

1. Imwrite()

/imwrite函数用于对多种类型的图像文件做写入操作。/

1. Imfinfo()

/imfinfo函数多用于读取图像文件的有关信息。/

1. 图像的显示
2. Imshow()

/imshow函数用于图像文件的显示/

1. Figure()

/figure函数用于设定图像显示窗口/

1. Subplot()

/subplot函数把图形窗口分成多个矩形部分，每个部分可以分别用来进行显示。

## 实验步骤

1. 准备好测试样本，本次实验采用的测试样本为400张92×112px的人脸图像，文件后缀命为bmp。
2. 编写好程序，并在MATLAB中新建脚本用于运行程序。
3. 得到结果

## 实验数据

实验代码：

clear all;

clf

clc

path='D:\face\_recognition\ORL\s'; % 文件路径

file='.bmp'; % 文件后缀名

m=1;

n=1;

for i=1:40

for j=1:10

filename = [path num2str(i) '\_' num2str(j) file];

data(:,:,i,j) = imread(filename);

if(j<=10)

X(:,m)=reshape(data(:,:,i,j),10304,1);m=m+1;

else

Y(:,n)=reshape(data(:,:,i,j),10304,1);n=n+1;

end

end

end

% 显示人脸数据库

for i=1:40

figure(i);

for j=1:10

subplot(2,5,j);imshow(data(:,:,i,j));xlabel(['s' num2str(i) '\\_' num2str(j) file]);

end

end

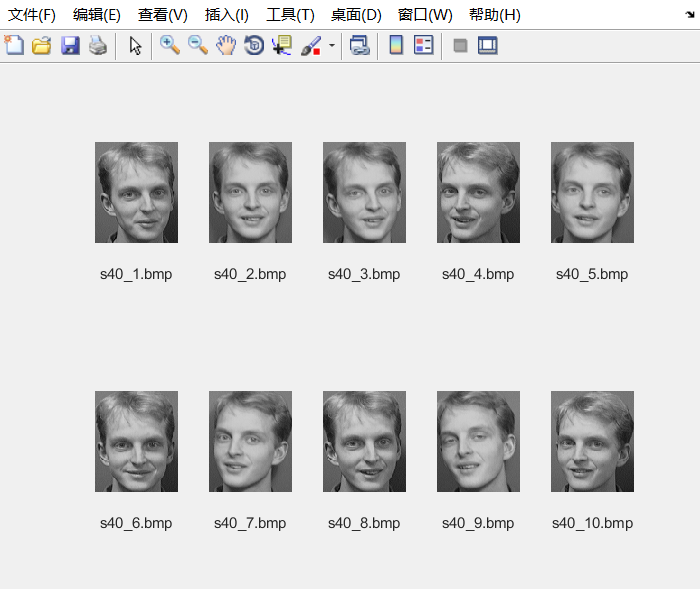
% 数据存储

save('ORL.mat','data','X');

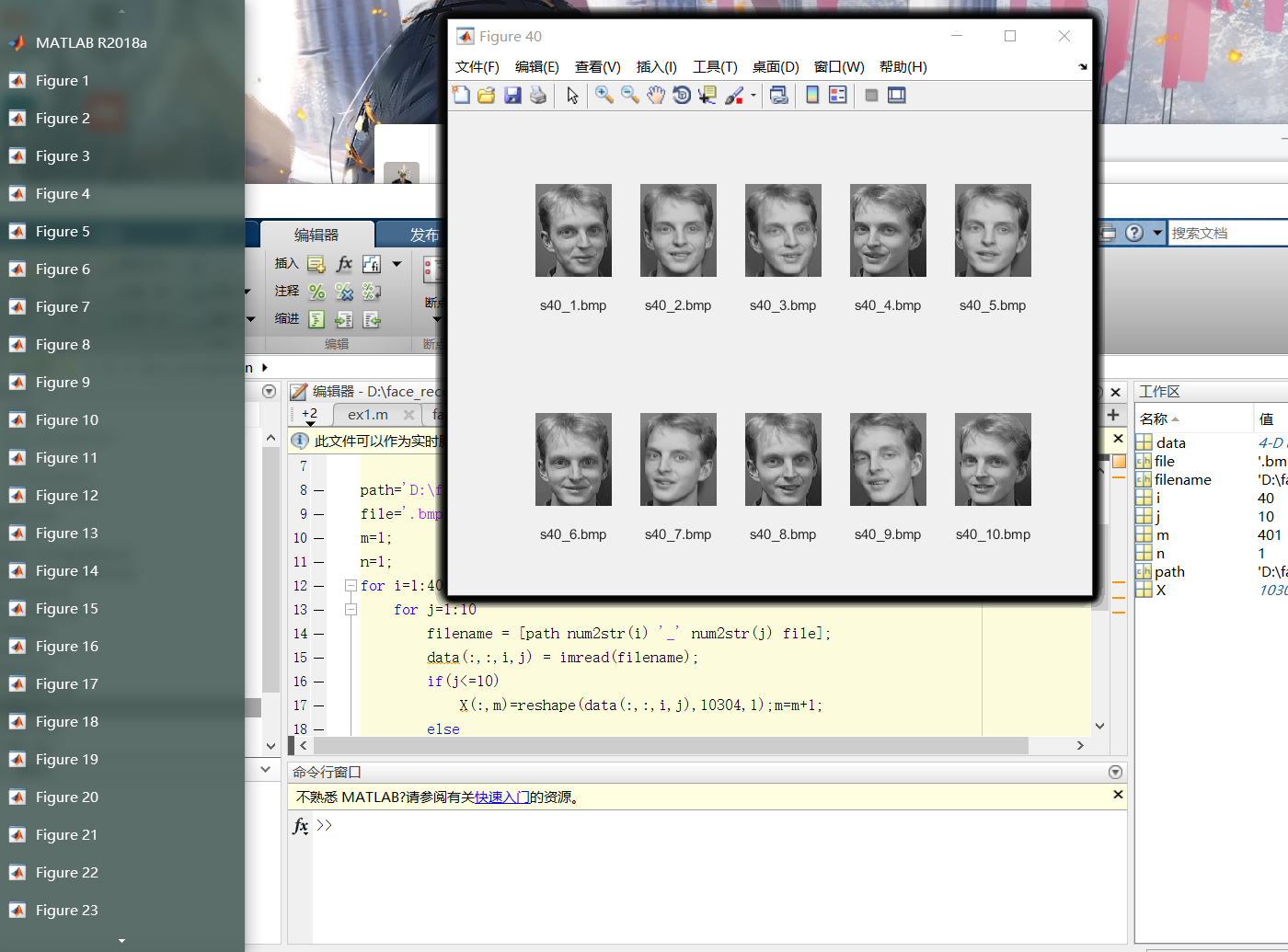
实验结果：

1. 成功读取四百个图像文件的信息。

按照2行5列的排列规则，有序展示40幅画布。



3.成功读取四百个图像文件，并分四十个画布分别展示。



## 实验总结

通过本次实验，对MATLAB软件对图像文件的读取有了一点基本的认识。学习并掌握了MATLAB编程语言的基本语法架构，并熟练掌握了imread、imshow、figure等图像处理函数，

# 实验二、PCA降维

## 实验目的

学习PCA降维的基本原理，并对实验一中读取的图像做PCA降维处理，揭示这些图像的本质，发现这种特性维度之间的线性关系，检测出这种线性关系并去除。

## 实验仪器设备及软件

计算机一台

MATLAB软件

## 实验原理

1. PCA算法基本思想：提取出空间原始数据中的主要特征(主元) ，减少数据冗余，使得数据在一个低维的特征空间被处理，同时保持原始数据的绝大部分的信息，从而解决数据空间维数过高的带来的问题。
2. 步骤：
   1. 对原始数据进行标准化处理；
   2. 计算样本相关系数矩阵；
   3. 求出矩阵R的特征值和相应的特征向量；
   4. 选择重要的主成分，并写出主成分表达式；
   5. 计算主成分得分；
   6. 依据主成分得分的数据，则可以进行下一步的统计分析。其中，常见的应用有主成分回归，变量子合集的选择、综合评价等方法。

## 实验步骤

1.新建一个脚本用于存储实验一中所读取图像的数据，并转化为一维向量，方便做PCA降维处理。

2.执行脚本，并保存该实验的结果，存储为ORL.mat文件。

3.再次新建一个脚本，读取上述过程中保存的ORL.mat文件中的数据，做PCA降维处理。

1.执行脚本，并保存实验结果。

## 五、实验数据

实验代码：

clear all;

clf

clc

%%

load ORL.mat;

% 求散布矩阵

X=double(X);

[n,p]=size(X);

m=mean(X')';

for i=1:p

Sr(:,i)=X(:,i)-m;

end

P=Sr'\*Sr;

%%

% 计算样本相关系数矩阵，并求出该矩阵的特征值和特征向量，选择主成分

[e,lambda]=eig(P,'nobalance');

for i=1:p

e(:,i)=e(:,i)/sqrt(e(:,i)'\*e(:,i));

end

lambda=diag(lambda);

k\_vector=find(lambda>lambda(p-100));

ee=e(:,k\_vector);

sum(lambda(k\_vector))/sum(lambda)

Eigenvector=X\*ee;

% 特征脸显示

k=length(k\_vector);

for i=1:8

figure(i)

for j=1:10

% Eigenvector(:,(i-1)\*10+j)=Eigenvector(:,(i-1)\*10+j)+m;

subplot(2,5,j);imshow(uint8(reshape(Eigenvector(:,(i-1)\*10+j),112,92)));xlabel([num2str((i-1)\*10+j)]);

end

end

% 训练图像在特征集中的坐标 A

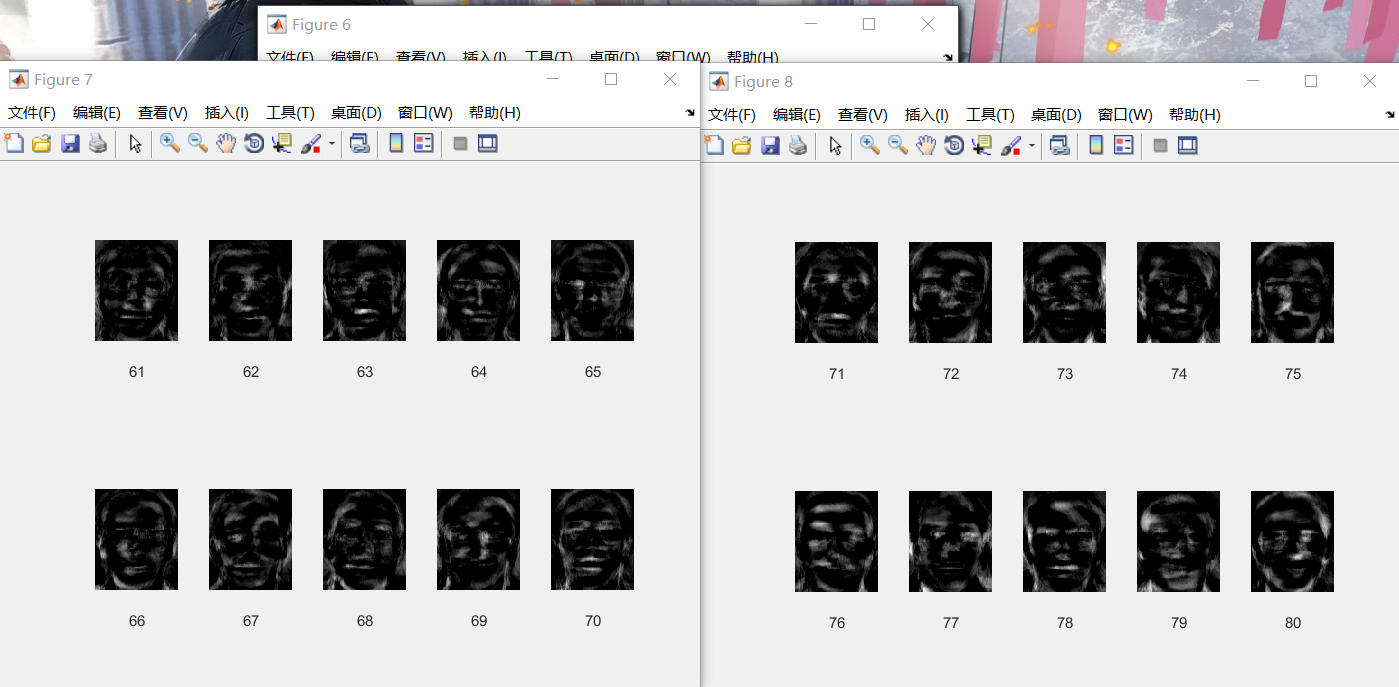
% Y=double(Y);

A=Eigenvector'\*X;

% B=Eigenvector'\*Y;

save('PCA.mat','A');

实验结果：



1. 实验总结

降维后的数据没有打乱原来数据的顺序。

结果数据没有打乱样本的排列顺序。

# 实验三、K-近邻法

## 实验目的

学习KNN算法的原理，并运用到实际问题中，通过MATLAB语言编写出能够并表达出KNN算法的程序

## 二、实验仪器设备及软件

计算机一台

MATLAB软件

## 实验原理

k近邻法是一种基本分类与回归方法。分类时，对于新的实例，根据与它最接近的k个训练实例的类别，通过多数表决等方式，进行预测。对于给定的训练集，当k值，距离度量和分类决策规则（统称三要素）确定后，基于k近邻法的模型就已经确定了。所以，它实际上利用训练集对特征向量空间进行划分，并没有显示的学习过程。一件事物的类别通常与它附近的事物具有相似性。

## 实验步骤

1. 计算已知类别的数据集(样本集)中的点与当前点之间的距离。
2. 按照距离递增排序。
3. 选取与当前点距离最小的k个点。
4. 确定k个点所属类别出现的概率。
5. 返回k个点中类别出现频率最高的类别。
6. 归纳总结，选取准确度最高的一组结果。

## 五、实验数据

实验代码：

clear all;

clf

clc

%%

K=1;

load ORL.mat;

load PCA.mat;

[xn,xp]=size(A);

[yn,yp]=size(B);

for j=1:yp

for i=1:xp

dist(i)=norm(A(:,i)-B(:,j));

end

[D,Ix]=sort(dist);

Ix=Ix(1:K);

R=fix((Ix-1)/9)+1;

for k=1:40

N(k)=length(find(R==k));

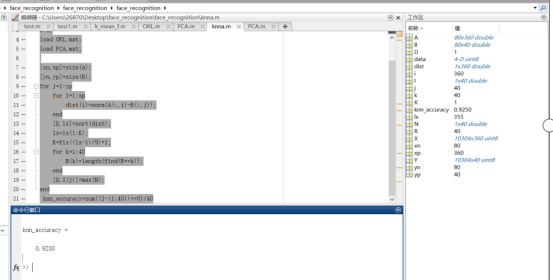
end

[D,I(j)]=max(N);

end

knn\_accuracy=sum((I-(1:40))==0)/40

实验结果：



## 实验总结

K仅近邻算法运行时，需要将待分类数据与训练集每条数据进行运算，需要耗费大量时间。训练集越庞大，所需时间越长。大多数据的每个特征对分类结果的影响是不尽相同的，甚至有些数据的某些特征对分类结果有决定性的影响，有些特征对分类结果毫无影响。后者的存在往往会使分类效果变差。而KNN算法则忽略了特征对分类结果的影响。为了避免这个缺陷，最直接的想法则是对数据特征进行加权处理。

# 实验四、神经网络模式识别

## 一、实验目的

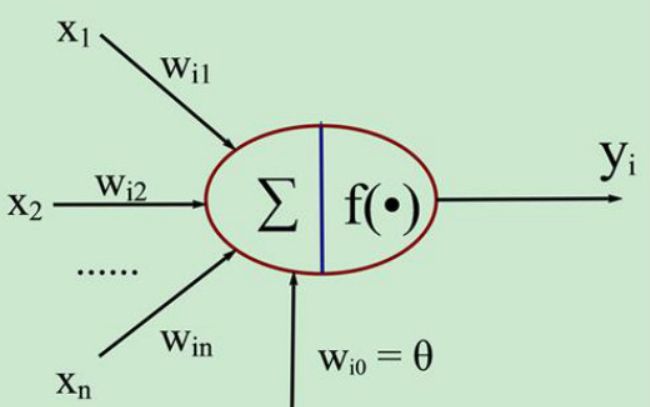
将模式识别方法与图像处理技术相结合，掌握利用**神经网络与KNN分类算法相结合**的方式对图像进行分类，通过实验加深对基本概念的理解。

## 二、实验仪器设备及软件

计算机一台

MATLAB软件

## 实验原理



上图是简化的神经元模型，Xi是一系列的输入变量，对应在其箭头方向上的是一个权重系数。权重系数呢是为了消除样本数据的偏差，通常是一个rand函数在0-1之间的取值。通过输入变量Xi和其自己的权重系数Wi的乘积的求和输入到神经元上。此时神经元得到了输入变量和其权重的乘积累加和。通过映射函数F（x）来进行映射得到结果。以上就是一个简单的神经元模型和信息传递过程。其中对输入样本集进行归一化处理，初始化权重，设置学习速率为0.1，结合前面的实验，对PCA降维后获得的数据采用激励函数进行网络训练。

## 实验步骤

1.从样本数据中随机性的选取一个样本数据{Ai，Bi},其中Ai是输出，Bi是期望输出

2.通过训练得出实际输出Oi

3.求误差D = Bi - Oi

4.根据步骤3得到的误差D，来调整各隐含层之间连接权值

5.对每个样本重复用以上步骤，直到对整体样本数据来说，误差收敛到控制范围内为止。

## 五、实验数据

实验代码：

clear all;

clc

%%

load ORL.mat;

load PCA.mat;

[xn,xp]=size(A);

[yn,yp]=size(B);

[input,minI,maxI] = premnmx(A);

E=eye(40);

for i=1:xp;

output(:,i)=E(:,fix((i-1)/9+1));

end

net = newff(minmax(input),[120,40],{'tansig','logsig'},'trainscg');

net.trainparam.epochs=10000;

net.trainparam.goal=0.000000001;

net.trainParam.lr=0.01; %设置学习速率为0.1

net=init(net); %初始化权重

% net = feedforwardnet(100);

net = train(net,input,output);

testInput = tramnmx (B , minI, maxI ) ;

YY = sim( net , testInput );

[M I]=max(YY);

net\_accuracy=sum((I-(1:40))==0)/40

%%

mm=find(M<0.2);

C=B(:,mm);

K=1;

[xn,xp]=size(A);

[yn,yp]=size(C);

for j=1:yp

for i=1:xp

dist(i)=norm(A(:,i)-C(:,j));

end

[D,Ix]=sort(dist);

Ix=Ix(1:K);

R=fix((Ix-1)/9)+1;

for k=1:40

N(k)=length(find(R==k));

end

[D,II(j)]=max(N);

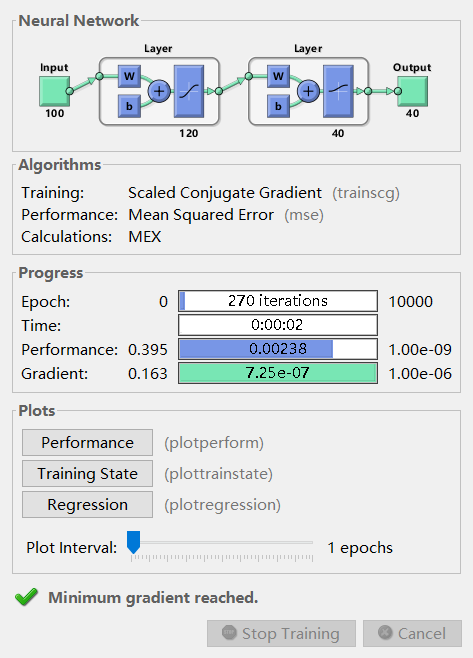
end

%%

I(mm)=II;

BPnet\_knn\_accuracy=sum((I-(1:40))==0)/40

实验结果：



## 六、实验总结

学习了BP神经网络和离散Hopfield神经网络的结构和原理，将KNN算法与神经网络结合，了解了反向传播学习算法对神经元的训练过程，了解了反向传播公式。通过构建了BP网络和离散Hopfield网络模式识别的实例，对前馈网络和反馈网络的原理及结构，模式识别的原理，识别过程的程序设计方法都有所学习。

# 实验五、K均值聚类算法

## 实验目的

学习K-mean算法的原理，并合理地运用解决实际问题。

## 二、实验仪器设备及软件

计算机一台

MATLAB软件

## 实验原理

K-Means算法的思想很简单，对于给定的样本集，按照样本之间的距离大小，将样本集划分为K个簇。让簇内的点尽量紧密的连在一起，而让簇间的距离尽量的大。

　　　　如果用数据表达式表示，假设簇划分为(C1,C2,...Ck)(C1,C2,...Ck)，则我们的目标是最小化平方误差E：

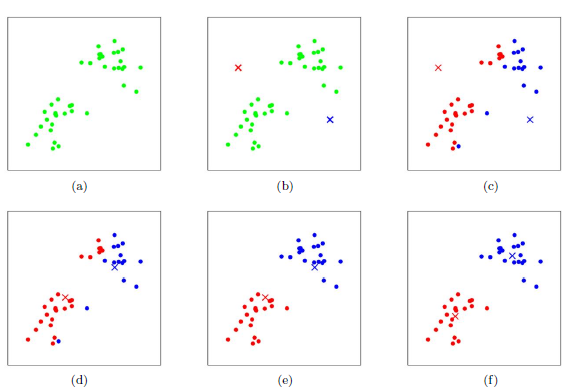
E=∑i=1k∑x∈Ci||x−μi||22E=∑i=1k∑x∈Ci||x−μi||22

　　　　其中μiμi是簇CiCi的均值向量，有时也称为质心，表达式为：

μi=1|Ci|∑x∈Cixμi=1|Ci|∑x∈Cix

　　　　如果我们想直接求上式的最小值并不容易，这是一个NP难的问题，因此只能采用启发式的迭代方法。

　　　　K-Means采用的启发式方式很简单，用下面一组图就可以形象的描述。



　　　　上图a表达了初始的数据集，假设k=2。在图b中，我们随机选择了两个k类所对应的类别质心，即图中的红色质心和蓝色质心，然后分别求样本中所有点到这两个质心的距离，并标记每个样本的类别为和该样本距离最小的质心的类别，如图c所示，经过计算样本和红色质心和蓝色质心的距离，我们得到了所有样本点的第一轮迭代后的类别。此时我们对我们当前标记为红色和蓝色的点分别求其新的质心，如图4所示，新的红色质心和蓝色质心的位置已经发生了变动。图e和图f重复了我们在图c和图d的过程，即将所有点的类别标记为距离最近的质心的类别并求新的质心。最终我们得到的两个类别如图f。

　　　　当然在实际K-Mean算法中，我们一般会多次运行图c和图d，才能达到最终的比较优的类别。

## 实验步骤

假设要把样本集分为c个类别，算法如下：

（1）适当选择c个类的初始中心；

（2）在第k次迭代中，对任意一个样本，求其到c个中心的距离，将该样本归到距离最短的中心所在的类，

（3）利用均值等方法更新该类的中心值；

（4）对于所有的c个聚类中心，如果利用（2）（3）的迭代法更新后，值保持不变（目标函数收敛），则迭代结束，否则继续迭代。

## 五、实验数据

实验代码：

clear all;

clc

load PCA.mat

x = A;

m1(:,1:40)=x(:,(1:40)\*10-5);

k=40;

step = 0;

while 1

step = step + 1;

for i=1:400

for j=1:40

dist(j)=norm(x(:,i)-m1(:,j));

[value index(i)]=min(dist);

end

end

for k=1:40

if find(index==k)

m2(:,k) = mean(x(:,find(index==k))')';

else

m2(:,k) = m1(:,k);

end

end

% if (m1==m2)

% break;

% else

F(step)=norm(m1-m2);

m1 = m2;

step

out = reshape(index,10,40)';

% pause;

if(step == 10)break;

% end

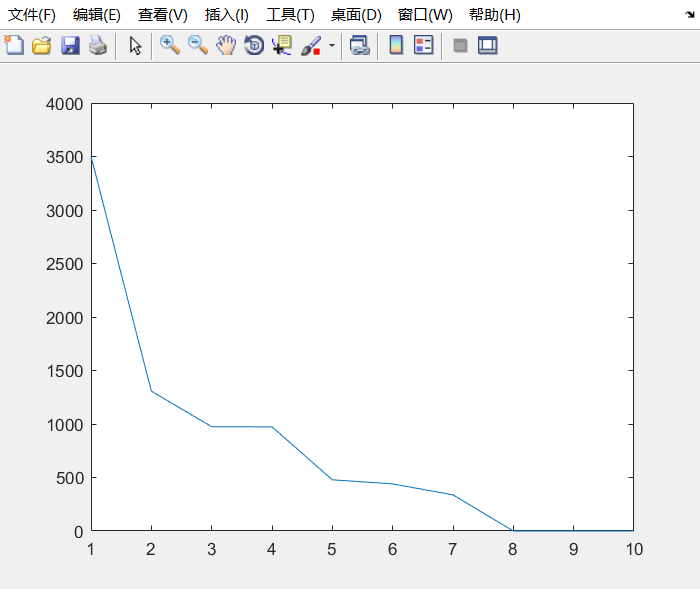
end

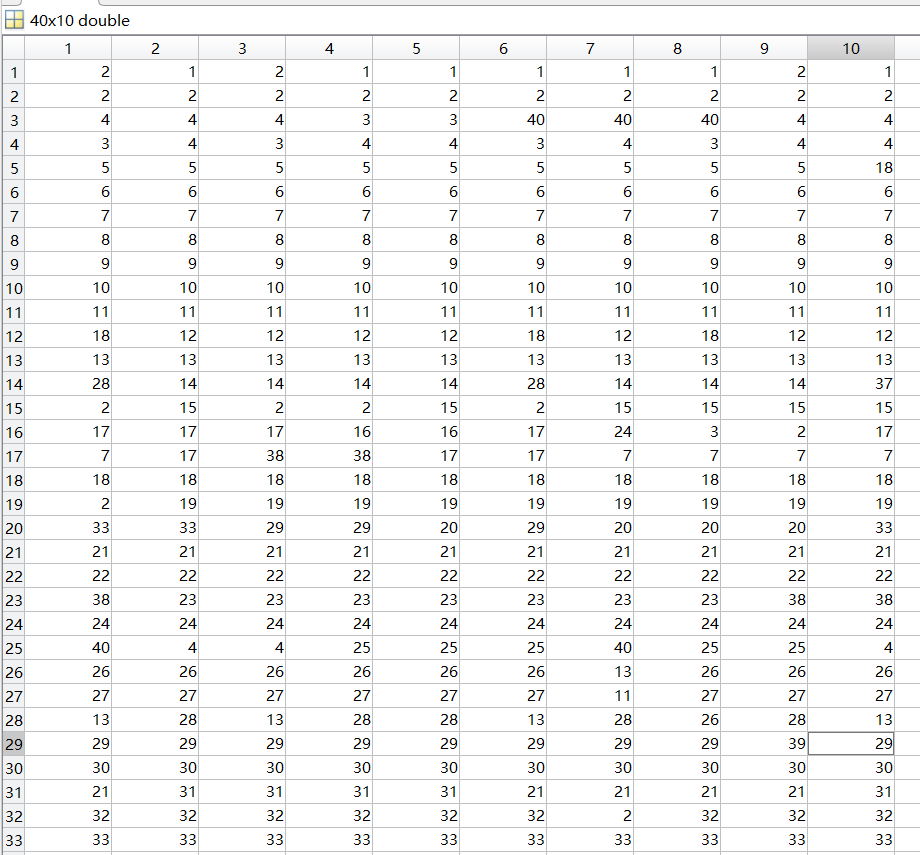
end

out = reshape(index,10,40)'

plot(F)

实验结果：





## 六、实验总结

K-Means是个简单实用的聚类算法，这里对K-Means的优缺点做一个总结。  
  
　　　　K-Means的主要优点有：  
  
　　　　1）原理比较简单，实现也是很容易，收敛速度快。  
  
　　　　2）聚类效果较优。  
  
　　　　3）算法的可解释度比较强。  
  
　　　　4）主要需要调参的参数仅仅是簇数k。  
  
　　　　K-Means的主要缺点有：  
  
　　　　1）K值的选取不好把握  
  
　　　　2）对于不是凸的数据集比较难收敛  
  
　　　　3）如果各隐含类别的数据不平衡，比如各隐含类别的数据量严重失衡，或者各隐含类别的方差不同，则聚类效果不佳。  
  
　　　　4） 采用迭代方法，得到的结果只是局部最优。  
  
　　　　5） 对噪音和异常点比较的敏感。