

K 近邻

一、实验名称：K 近邻

二、实验目的

对每个测试样本，在训练集中找出与其距离最近的 K 个点，根据这 K 个点的类别判断测试样本的类别。

三、实验原理

求离测试点最近的 k 个数据点，它们数据标签的众数即为测试点的数据类别。

四、实验步骤

- 1、生成数据集：在 0-10 的 2 维平面中随机生成 $n=2000$ 个数据点，并根据事先设好的标签区间为它们设好标签 1-9。对于在区间间隔部分的数据点，其标签为 0，删去这部分点。
- 2、生成测试集：同上一步，生成 $m=100$ 个的测试集，并得到它们真实的数据标签。
- 3、K 近邻模型：按顺序取测试点，计算其到所有数据点的欧氏距离。二维的欧式距离就是 $distance = (x^2 + y^2)^{1/2}$ 。升序排序，取其中前 k 个，索引得到这 k 个距离测试点最近的数据点的类别标签，这些标签的众数即是测试点的数据类别。
- 4、结果展示：在图像中标注数据集、测试集，不同的数据标签的点用不同的颜色或符号，并显示数据集、测试集的保留率（即数据标签不是 0 的概率） n_rate 和 m_rate 、测试集预测正确率和代码运行时间。
- 5、训练：改动参数（如上标红），进行调试训练，综合考虑，获得合适的参数，这里主要获得适合一组 n 、 m 的 k 。

五、代码

主要的训练部分已标红

% 数据样本生成

```
n = 2000; % 样本量
X = rand(n,2)*10; % 数据点（2 维）：0-10 随机
Y = zeros(n,1); % 类别标签
for i=1:n
    if 0<X(i,1) && X(i,1)<3 && 0<X(i,2) && X(i,2)<3
        Y(i) = 1;
    end
    if 0<X(i,1) && X(i,1)<3 && 3.5<X(i,2) && X(i,2)<6.5
        Y(i) = 2;
    end
    if 0<X(i,1) && X(i,1)<3 && 7<X(i,2) && X(i,2)<10
        Y(i) = 3;
    end
    if 3.5<X(i,1) && X(i,1)<6.5 && 0<X(i,2) && X(i,2)<3
        Y(i) = 4;
    end
end
```

```

if 3.5<X(i,1) && X(i,1)<6.5 && 3.5<X(i,2) && X(i,2)<6.5
    Y(i) = 5;
end
if 3.5<X(i,1) && X(i,1)<6.5 && 7<X(i,2) && X(i,2)<10
    Y(i) = 6;
end
if 7<X(i,1) && X(i,1)<10 && 0<X(i,2) && X(i,2)<3
    Y(i) = 7;
end
if 7<X(i,1) && X(i,1)<10 && 3.5<X(i,2) && X(i,2)<6.5
    Y(i) = 8;
end
if 7<X(i,1) && X(i,1)<10 && 7<X(i,2) && X(i,2)<10
    Y(i) = 9;
end
end
X = X(Y>0,:);          % 去掉类别间隔中的点, 其 Y=0
Y = Y(Y>0,:);
n_rate = length(Y) / n;
n = length(Y);

%{
% 图一: 数据点
figure(1)
set(gcf,'Position',[1,1,700,600], 'color','w')
set(gca,'FontSize',18)
plot(X(Y==1,1),X(Y==1,2),'ro','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==2,1),X(Y==2,2),'ko','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==3,1),X(Y==3,2),'bo','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==4,1),X(Y==4,2),'g*','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==5,1),X(Y==5,2),'m*','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==6,1),X(Y==6,2),'c*','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==7,1),X(Y==7,2),'b+','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==8,1),X(Y==8,2),'r+','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==9,1),X(Y==9,2),'k+','LineWidth',1,'MarkerSize',10);
hold on;

```

```

xlabel('x axis');
ylabel('y axis');
%}

% 测试样本生成
m = 100; % 测试样本量
Xt = rand(m,2)*10;
Yt = zeros(m,1);
for i=1:m
    if 0<Xt(i,1) && Xt(i,1)<3 && 0<Xt(i,2) && Xt(i,2)<3
        Yt(i) = 1;
    end
    if 0<Xt(i,1) && Xt(i,1)<3 && 3.5<Xt(i,2) && Xt(i,2)<6.5
        Yt(i) = 2;
    end
    if 0<Xt(i,1) && Xt(i,1)<3 && 7<Xt(i,2) && Xt(i,2)<10
        Yt(i) = 3;
    end
    if 3.5<Xt(i,1) && Xt(i,1)<6.5 && 0<Xt(i,2) && Xt(i,2)<3
        Yt(i) = 4;
    end
    if 3.5<Xt(i,1) && Xt(i,1)<6.5 && 3.5<Xt(i,2) && Xt(i,2)<6.5
        Yt(i) = 5;
    end
    if 3.5<Xt(i,1) && Xt(i,1)<6.5 && 7<Xt(i,2) && Xt(i,2)<10
        Yt(i) = 6;
    end
    if 7<Xt(i,1) && Xt(i,1)<10 && 0<Xt(i,2) && Xt(i,2)<3
        Yt(i) = 7;
    end
    if 7<Xt(i,1) && Xt(i,1)<10 && 3.5<Xt(i,2) && Xt(i,2)<6.5
        Yt(i) = 8;
    end
    if 7<Xt(i,1) && Xt(i,1)<10 && 7<Xt(i,2) && Xt(i,2)<10
        Yt(i) = 9;
    end
end
Xt = Xt(Yt>0,:);
Yt = Yt(Yt>0,:);
m_rate = length(Yt) / m;
m = length(Yt);

%{
% 图二：数据点与测试点

```

```

figure(2)
set(gcf,'Position',[1,1,700,600], 'color','w')
set(gca,'FontSize',18)
plot(X(Y==1,1),X(Y==1,2),'ro','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==2,1),X(Y==2,2),'ko','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==3,1),X(Y==3,2),'bo','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==4,1),X(Y==4,2),'g*','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==5,1),X(Y==5,2),'b*','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==6,1),X(Y==6,2),'c*','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==7,1),X(Y==7,2),'b+','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==8,1),X(Y==8,2),'r+','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==9,1),X(Y==9,2),'k+','LineWidth',1,'MarkerSize',10);
hold on;
plot(Xt(:,1),Xt(:,2),'ms','MarkerFaceColor','m','LineWidth',1,'MarkerSize',
10);
hold on;
xlabel('x axis');
ylabel('y axis');
%}

```

% K-近邻 （预测输出，并与测试数据的真实输出比较，计算错误率）

```

tic()
Ym = zeros(m,1); % 预测集
dis = zeros(n,1); % 距离集
k = 10; % 近邻数目
k_dis = zeros(k,1); % 近邻类别集
for i=1:m
    for j=1:n % 计算欧氏距离
        dis(j) = norm(Xt(i,:)-X(j,:));
    end
    [a, index] = sort(dis); % 升序排序, index 记录下标
    for j=1:k % k 个最近邻数据点的类别标签
        k_dis(j) = Y(index(j));
    end
    Ym(i) = mode(k_dis); % 标签众数即为所求
end

```

toc()

% 图三：预测结果

```
figure(3)
set(gcf,'Position',[1,1,700,600], 'color','w')
set(gca,'FontSize',18)
plot(X(Y==1,1),X(Y==1,2),'ro','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==2,1),X(Y==2,2),'ko','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==3,1),X(Y==3,2),'bo','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==4,1),X(Y==4,2),'g*','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==5,1),X(Y==5,2),'b*','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==6,1),X(Y==6,2),'c*','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==7,1),X(Y==7,2),'b+','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==8,1),X(Y==8,2),'r+','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(Y==9,1),X(Y==9,2),'k+','LineWidth',1,'MarkerSize',10);
hold on;
plot(Xt(Ym==1,1),Xt(Ym==1,2),'ro','MarkerFaceColor','r','LineWidth',1,'MarkerSize',10);
hold on;
plot(Xt(Ym==2,1),Xt(Ym==2,2),'ko','MarkerFaceColor','k','LineWidth',1,'MarkerSize',10);
hold on;
plot(Xt(Ym==3,1),Xt(Ym==3,2),'bo','MarkerFaceColor','b','LineWidth',1,'MarkerSize',10);
hold on;
plot(Xt(Ym==4,1),Xt(Ym==4,2),'go','MarkerFaceColor','g','LineWidth',1,'MarkerSize',10);
hold on;
plot(Xt(Ym==5,1),Xt(Ym==5,2),'bo','MarkerFaceColor','b','LineWidth',1,'MarkerSize',10);
hold on;
plot(Xt(Ym==6,1),Xt(Ym==6,2),'co','MarkerFaceColor','c','LineWidth',1,'MarkerSize',10);
hold on;
plot(Xt(Ym==7,1),Xt(Ym==7,2),'bo','MarkerFaceColor','b','LineWidth',1,'MarkerSize',10);
```

```

hold on;
plot(Xt(Ym==8,1),Xt(Ym==8,2),'ro','MarkerFaceColor','r','LineWidth',1,'MarkerSize',10);
hold on;
plot(Xt(Ym==9,1),Xt(Ym==9,2),'ko','MarkerFaceColor','k','LineWidth',1,'MarkerSize',10);
hold on;
xlabel('x axis');
ylabel('y axis');

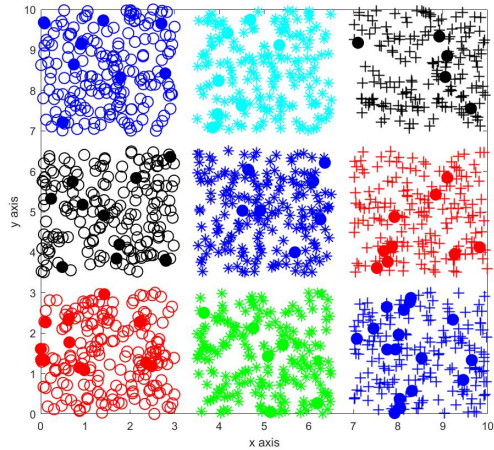
% 结果与错误率
disp(n_rate)
disp(m_rate)
count = 0;
for i=1:m
    if Ym(i)==Yt(i)
        count = count + 1;
    end
end
accuracy = count / m;
disp(accuracy)

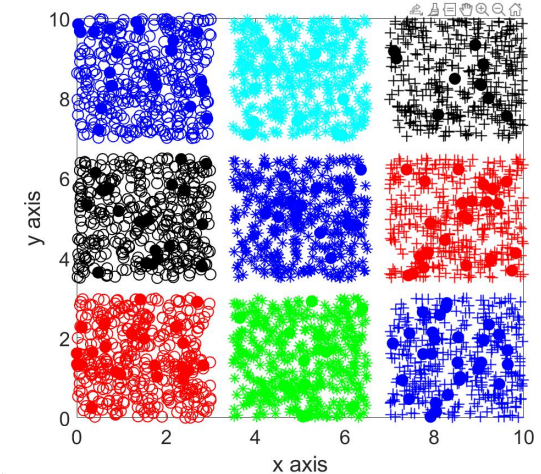
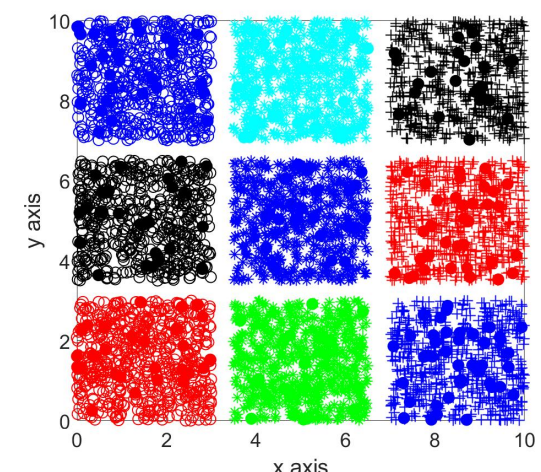
```

六、调试训练

以下通过修改不同参数，调试模型。改变的参数分别为数据集样本量 n 、数据维度、标签数量、距离方式（度量函数）以及最重要的 k 值。经过调试，源代码中的参数是合理的。

1、 $n=2000$ ，欧式距离， $k=10$ ，连续重复 3 次，展示在一张画布上

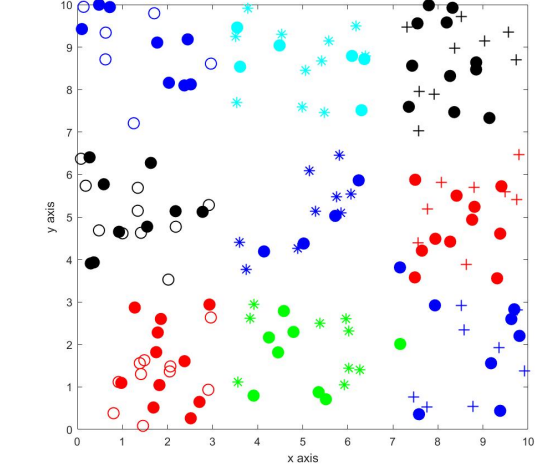
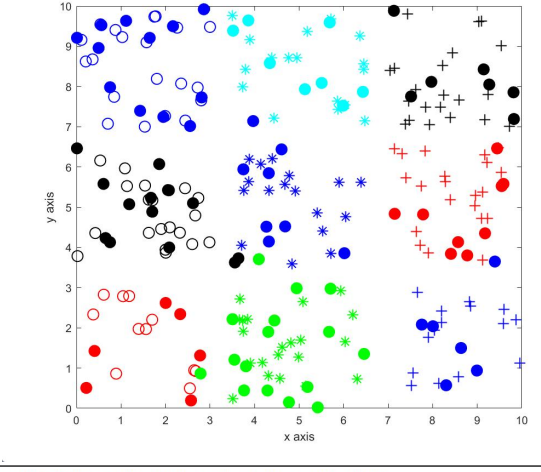
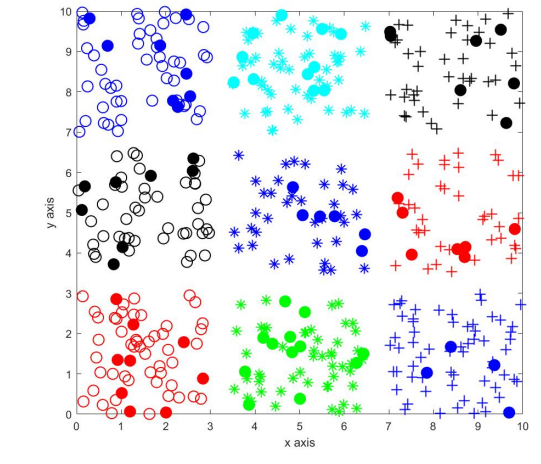
次数	图三（测试）	数据集保留率	测试集保留率	测试集预测正确率	耗时 (s)
1		0.8070	0.8100	1	0.110158

2		0. 8060	0. 8300	1	0. 11 2861
3		0. 8245	0. 7900	1	0. 10 7096

通过连续的测试，发现在该参数下，稳定地获得了 100%的准确度，证明题设背景下，参数是合理的。

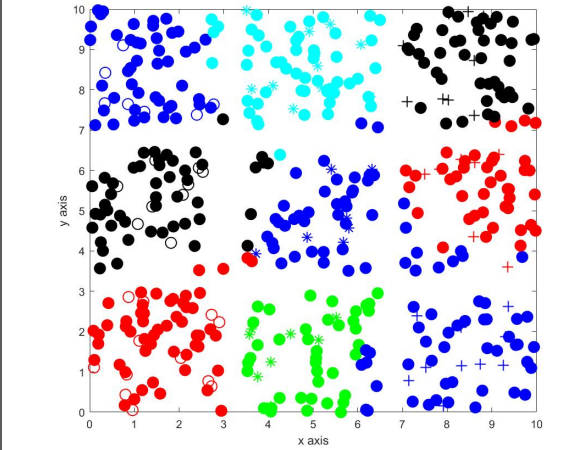
2、欧式距离， $k=10$ ，改动 n

n	图三（测试）	数据集 保留率	测试集 保留率	测试集预 测正确率	耗 时 (s)
---	--------	------------	------------	--------------	------------

100		0.8000	0.7600	0.9737	0.006059
200		0.8150	0.8500	0.9294	0.014495
500		0.8280	0.7300	1	0.027011

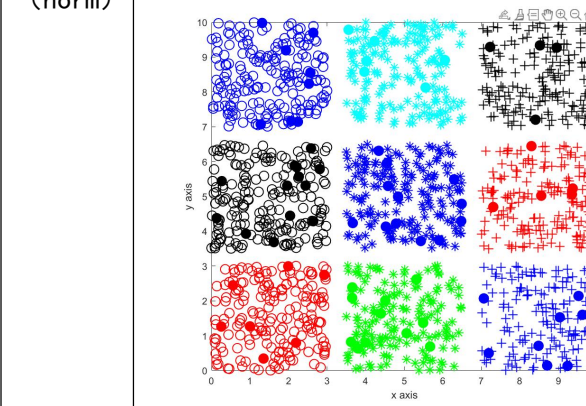
原设定的数据集样本数已经稳定得到 100% 的准确率，这里进行降低，并测试其在增长过程中，模型的准确度。可以发现，运行时间与数据集样本量基本正相关，而在数据集样本量过低时，虽然也保持了较高的准确度，但是不能达到 100%。这说明，该模型需要一定量的数据集进行训练。

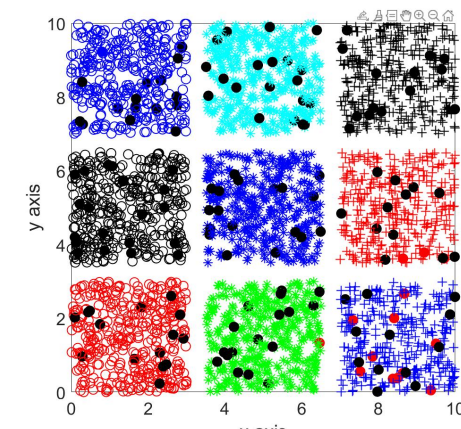
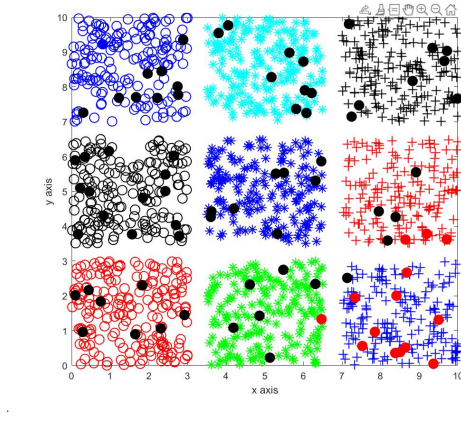
根据附加题要求，这里给出 $n=100, m=500$ 时的测试结果：

图三（测试）	数据集保留率	测试集保留率	测试集预测正确率	耗时(s)
<div>文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H) </div>	0. 7900	0. 8200	0. 9049	0. 030119

可以发现，在此种情况下，产生了较大的误差，而观察图像可发现，误分类点集中在交界处，这说明数据集过少时，边界处带来的误差被放大。

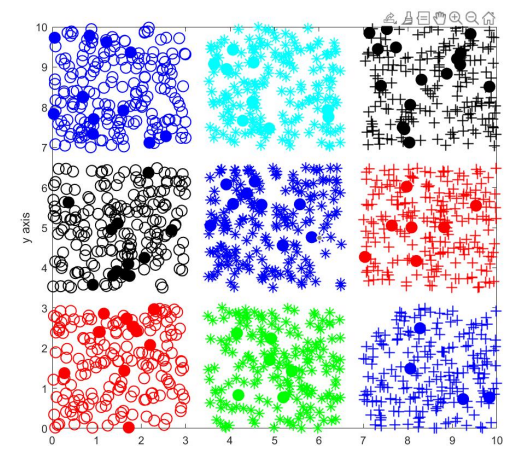
3、 $n=2000, k=10$ ，其他距离方式（度量函数）

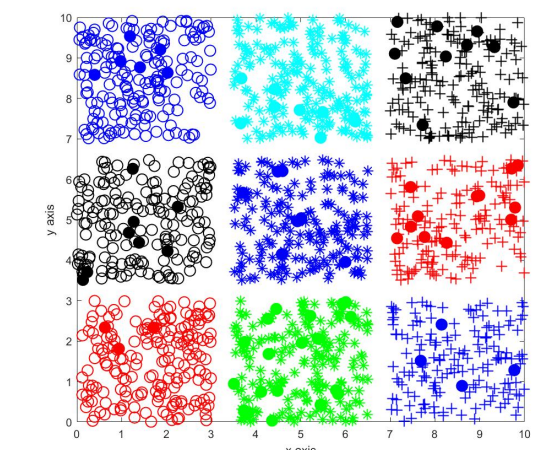
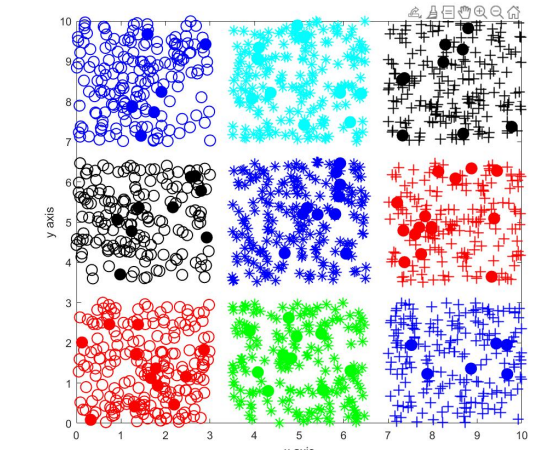
方式(计算使用的函数)	图三（测试）	数据集保留率	测试集保留率	测试集预测正确率	耗时(s)
欧式 (norm)	<div>文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H) </div>	0. 7995	0. 7700	1	0. 102901

棋盘 (sum)		0. 8100	0. 7600	0. 0789	0. 10 7591
切比雪夫 (max)		0. 8280	0. 8200	0. 1463	0. 12 4262

通过对比可以发现，在度量距离的方式中，欧式距离是最为合理的，其他度量方式的正确率过低。扩展到更高维的空间，也能得到相似的结论，这说明，欧式距离较好地运用了不同维度的数据，并且计算函数是综合性的。

4、n=2000，欧式距离，改动 k

k	图三（测试）	数据集保留率	测试集保留率	测试集预测正确率	耗时 (s)
1		0. 8185	0. 8400	1	0. 12 5771

3		0.8050	0.7800	1	0.10 9165
5		0.8015	0.8700	1	0.12 0119

原设定的 k 值已经稳定得到 100% 的准确率，这里进行降低，并测试其在增长过程中，模型的准确度。可以发现，即使 $k=1$ 也具有 100% 的准确度，这说明数据集对空间的覆盖比较好，减少了误差可能。然而，最重要的原因是，在题设背景下，对边界处的数据点和样本点进行了忽略，这部分数据才是造成误差的重点。

5、可以改变维数、增减类别数量、改变划分区间的方式，进行进一步测试，但这些较为低级、繁琐，多数无法输出图像，并且对参数的获取影响不大，就不做展示了。

七、分析

K 近邻模型在匹配的 n 与 k 参数下，可以训练出效果良好的分类模型，并且适用于多元而不同于 2 元的一次性分类。但同时，K 近邻也有很多问题，比如

- 1、在本题设下，对数据集、测试集位于分类间隔中的部分进行了剔除，这忽略了分类真正的难点，即边缘部分，并且在题设逻辑上规避了 a 类点可能落在 b 类区间的可能，而这这就要求在处理原始数据时，应放大类别之间的差别。
- 2、在面对大量数据集时，本代码也需要将测试点逐一与数据点计算距离，这比较消耗时间，可以尝试缩小查找的范围。这里可以尝试如下代码（改动部分已标红）：

```
tic()
```

```
Ym = zeros(m,1);
```

```
% 预测集
```

```

dis = Inf(n,1); % 距离集，初始化为无穷，这样超出范围的
点就自动忽略了，也不用再进行赋值
k = 10; % 近邻数目
k_dis = zeros(k,1); % 近邻类别集
for i=1:m
    for j=1:n % 计算欧氏距离，仅计算方形区间内的点
        if (X(j:1)-Xt(i:1))<1 & (X(j:2)-Xt(i:2))<1
            dis(j) = norm(Xt(i,:)-X(j,:));
        end
    end
    [a, index] = sort(dis); % 升序排序，index 记录下标
    for j=1:k % k 个最近邻数据点的类别标签
        k_dis(j) = Y(index(j));
    end
    Ym(i) = mode(k_dis); % 标签众数即为所求
end
toc()

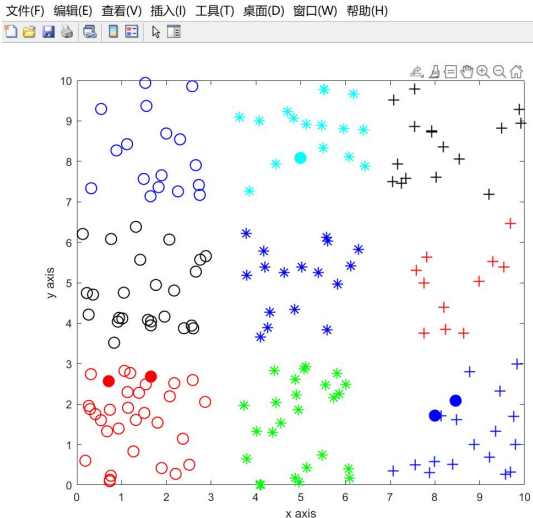
```

这种情况是为了处理数据集过大，运行时间过长，本题设下没有达到这种要求，就使用原代码了。需要注意的是，在 k 过大的情况下，可能出现限制范围内数据点不够的情况，这个代码没有考虑。

八、附加题

- 1、见六、2、。
- 2、见六、3、。
- 3、因素：数据集样本量和测试集样本量的比、距离方式（度量函数）、K 值、划分区间、分布方式等等。

原题设中使用的是随机分布的数据，现在使用高斯分布进行实验：

高 斯 分 布	图三（测试）	数据集 保留率	测试集 保留率	测试集预 测正确率	耗 时 (s)
		0.0870	0.0500	1	0.00 2023

在高斯分布下，虽然仍保持 100%的准确度，但从极低的数据集、测试集保留率可以看出，大部分数据都被清理掉了，而这些数据正对应分类边界处，是产生误差的地方。可见，

不一样的数据分布会对测试结果造成不一样的影响。