石家庄铁道大学
SHIJIAZHUANG TIEDAO UNIVERSITY

在线开放课程

MATLAB在科学研究中的应用

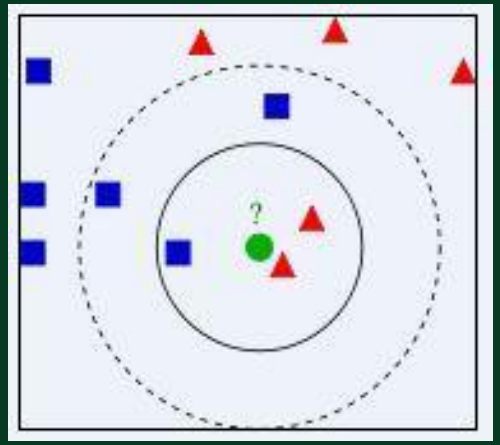
# K近邻

主讲 ： 卞建鹏

# 1、k近邻



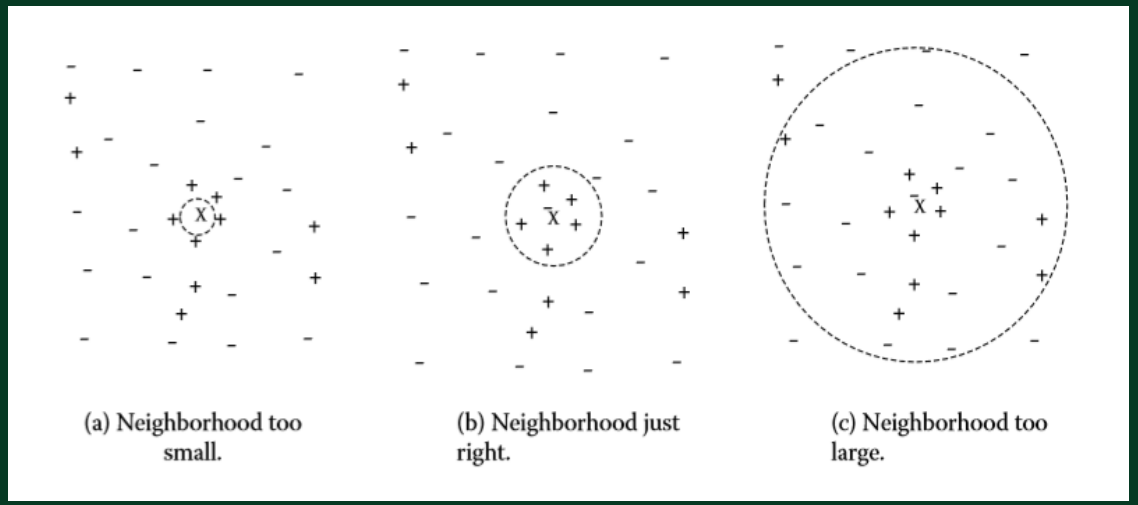
基本思想：在训练数据集中找到k个最近邻的实例，类别由这k个近邻中占最多的实例类别所决定。如当k=3时，即选择最近的3个点，三角形占2/3，可得绿色圆圈属于三角形类；同理，当k=5，属于正方形类。



# 1、k近邻



K值选取太小或太大都会影响精度，一般取3-10较为合适。



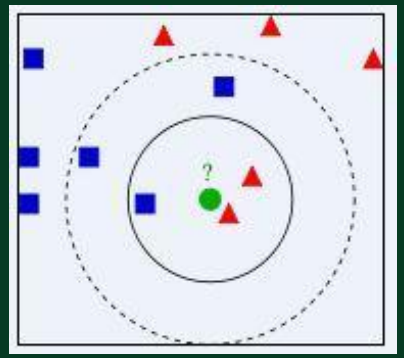
# 1、k近邻



欧式距离 $d(x_i, x_j) = [\sum_{k=1}^{p} (x_{ik} - x_{jk})^2]^{1/2}$ **pdist(x)**

绝对距离 $d(x_i, x_j) = \sum_{k=1}^{p} | x_{ik} - x_{jk} |$ **pdist(x,'cityblock')**

切式距离 $d(x_i, x_j) = \max_{1 \le k \le p} | x_{ik} - x_{jk} |$ **max(abs(xi-xj))**



# 1、k近邻



## 计算过程

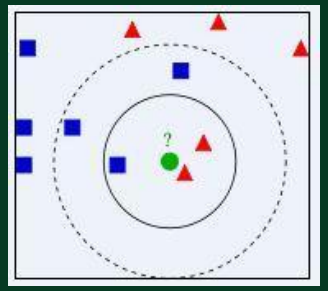
（1）计算已知类别数据集中的点与当前待分类点之间的距离；

（2）按照距离递增次序排序；

（3）选取与当前点距离最小的k个点；

（4）确定前k个点所在类别的出现频率；

（5）返回前k个点出现频率最高的类别作为当前点的预测分类。

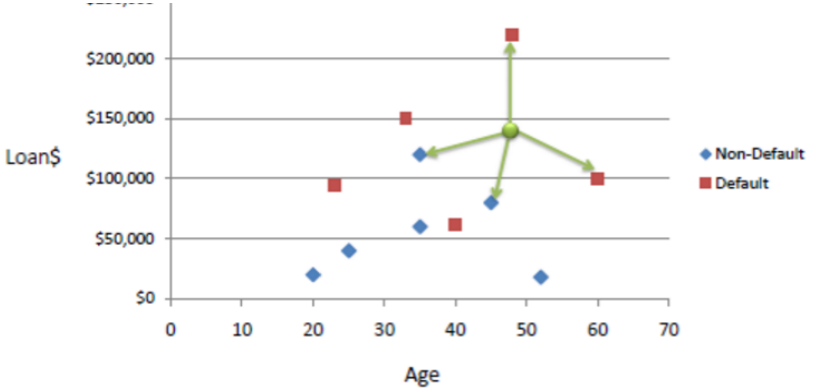


| Age | Loan | Default | Distance | |
|---|---|---|---|---|
| 25 | $40,000 | N | 102000 | |
| 35 | $60,000 | N | 82000 | |
| 45 | $80,000 | N | 62000 | |
| 20 | $20,000 | N | 122000 | |
| 35 | $120,000 | N | 22000 | 2 |
| 52 | $18,000 | N | 124000 | |
| 23 | $95,000 | Y | 47000 | |
| 40 | $62,000 | Y | 80000 | |
| 60 | $100,000 | Y | 42000 | 3 |
| 48 | $220,000 | Y | 78000 | |
| 33 | $150,000 | Y | 8000 | 1 |
| | | | | |
| 48 | $142,000 | ? | | |

Euclidean Distance

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

D = Sqrt[(48-33)^2 + (142000-150000)^2] = 8000.01 >> Default=Y




k=3
k=1

# 1、k近邻



归一化距离：由于数据单位与尺度的不同，会影响距离计算的精度，因此需要归一化。

| Age | Loan | Default | Distance | |
|---|---|---|---|---|
| 25 | $40,000 | N | 102000 | |
| 35 | $60,000 | N | 82000 | |
| 45 | $80,000 | N | 62000 | |
| 20 | $20,000 | N | 122000 | |
| 35 | $120,000 | N | 22000 | 2 |
| 52 | $18,000 | N | 124000 | |
| 23 | $95,000 | Y | 47000 | |
| 40 | $62,000 | Y | 80000 | |
| 60 | $100,000 | Y | 42000 | 3 |
| 48 | $220,000 | Y | 78000 | |
| 33 | $150,000 | Y | 8000 | 1 |
| | | | | |
| 48 | $142,000 | ? | | |

| Age | Loan | Default | Distance |
|---|---|---|---|
| 0.125 | 0.11 | N | 0.7652 |
| 0.375 | 0.21 | N | 0.5200 |
| 0.625 | 0.31 | N | 0.3160 |
| 0 | 0.01 | N | 0.9245 |
| 0.375 | 0.50 | N | 0.3428 |
| 0.8 | 0.00 | N | 0.6220 |
| 0.075 | 0.38 | Y | 0.6669 |
| 0.5 | 0.22 | Y | 0.4437 |
| 1 | 0.41 | Y | 0.3650 |
| 0.7 | 1.00 | Y | 0.3861 |
| 0.325 | 0.65 | Y | 0.3771 |
| | | | |
| 0.7 | 0.61 | ? | |

Standardized Variable

$$X_s = \frac{X - Min}{Max - Min}$$

# 2、k近邻应用



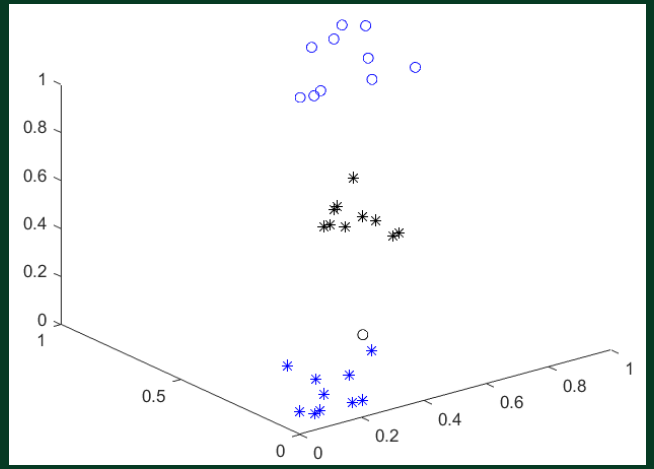
```
clc;  clear
traindata1=randn(10,3);
traindata2=randn(10,3)+5;
traindata3=randn(10,3)+10;
[row1,col1]=size(traindata1);
[row2,col2]=size(traindata2);
[row3,col3]=size(traindata3);

testdata1=rand(1,3)+1;
testdata=repmat(testdata1,30,1)

 k=4;                %the forth nearest
```

```
traindata1 =

    0.5377   -1.3499    0.6715
    1.8339    3.0349   -1.2075
   -2.2588    0.7254    0.7172
    0.8622   -0.0631    1.6302
    0.3188    0.7147    0.4889
   -1.3077   -0.2050    1.0347
   -0.4336   -0.1241    0.7269
    0.3426    1.4897   -0.3034
    3.5784    1.4090    0.2939
    2.7694    1.4172   -0.7873
```

```
traindata=[traindata1;traindata2;traindata3]
max1=max(traindata);
 min1=min(traindata);
maax=repmat(max1,30,1);
 minx=repmat(min1,30,1);
trainguiyi=(traindata-minx)./(maax-minx)
testguiyi=(testdata-minx)./(maax-minx)
```

```
trainguiyi =

    0.2468    0.0606    0.0277
    0.1157    0.0496    0.1021
    0.1208    0.0941         0
    0.1375    0.0936    0.0080
         0         0    0.0954
    0.2822    0.0664    0.2171
    0.2105    0.0557    0.0351
    0.1410    0.1187    0.1253
    0.2778    0.1557    0.0781
    0.0794    0.1570    0.1843
    0.4414    0.4174    0.6034
    0.5136    0.5258    0.5143
    0.5471    0.4510    0.5068
    0.5824    0.5370    0.6167
    0.6110    0.4055    0.4276
    0.5171    0.3548    0.5463
    0.4155    0.3532    0.5572
    0.4638    0.5051    0.4719
    0.4432    0.4522    0.5083
    0.6629    0.4507    0.3990
    0.7596    0.9307    0.7174
    0.8403    0.7933    0.8201
    0.8800    0.8718    0.9935
    1.0000    0.8206    0.8017
    0.7906    0.8880    0.9624
    0.8456    0.8162    0.8962
    0.8282    0.9028    1.0000
    0.7090    0.9227    0.7314
    0.8052    1.0000    0.8708
    0.7180    0.8484    0.7910
```

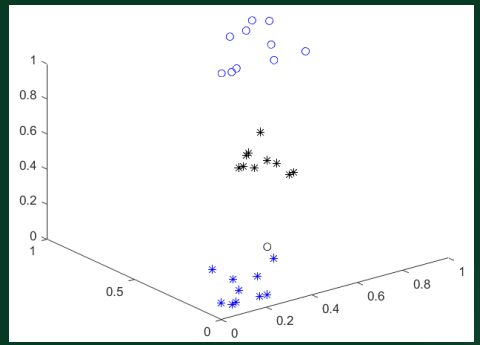
```
plot3(trainguiyi(1:10,1),trainguiyi(1:10,2),trainguiyi(1:10,3),'b*')
hold on
plot3(trainguiyi(11:20,1),trainguiyi(11:20,2),trainguiyi(11:20,3),'k*')
hold on
plot3(trainguiyi(21:30,1),trainguiyi(21:30,2),trainguiyi(21:30,3),'bo')
hold on
plot3(testguiyi(:,1),testguiyi(:,2),testguiyi(:,3),'ko')
```



```
d=sqrt(sum((trainguiyi-testguiyi).^2,2))
[b,index]=sort(d');
y=index(1:k)                    %k=4
```

在线开放课程

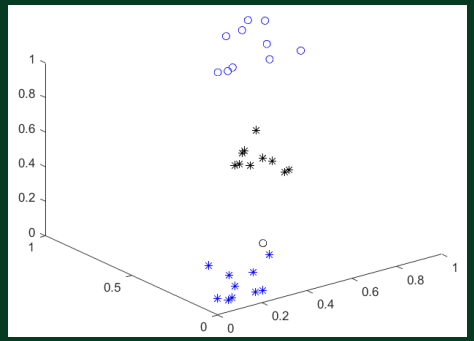
```
b =

  1 至 11 列

    0.1067    0.2286    0.2334    0.2556    0.2641    0.2792    0.2938    0.3009    0.3088    0.3383    0.4039

  12 至 22 列

    0.4155    0.4263    0.4295    0.4494    0.4544    0.5106    0.5340    0.5520    0.5691    1.0015    1.0497

  23 至 30 列

    1.0632    1.0871    1.0982    1.1310    1.1547    1.1637    1.1649    1.1821
```




```
index =

  1 至 18 列

     3     6     1     9     4     7    10     5     2     8    17    12    14    20    13    15    16    19

  19 至 30 列

    11    18    22    26    28    30    27    21    24    29    25    23
```

```
class1=zeros(1,30); class2=zeros(1,30); class3=zeros(1,30);
for i=1:k
  if y(i)>=1&y(i)<=10
    class1(i)=1;
  else if  y(i)>=11&y(i)<=20
     class2(i)=1;
      else y(i)>=21&y(i)<=30
     class3(i)=1;
      end
  end
end
```

```
y =

     3     6     1     9
```




```
class1 =

  1 至 18 列

     1     1     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0

  19 至 30 列

     0     0     0     0     0     0     0     0     0     0     0     0
```

# 小结

1. 基本思想
2. 计算过程
3. k近邻应用