

## 第一问的程序

导入数据：

```
1 data = Import[
2     "G:\\GitHub Local \\
3     Repository\\Mathematical-Modeling-Campus-Competition\\
4     2020年兰州大学数学建模竞赛赛题 (1)\\Data.xlsx"];
```

数据预处理

```
1 train = data[[1, 2 ;; All]];
2 test = data[[2, 2 ;; All]];
3 asso=MapThread[Rule,{train[[All,1;;7]],train[[All,8]]}
4 [[2;;All]]];
4 model=Classify[asso,Method->"SupportVectorMachine"];
```

## 主成分分析

```
1 features=train[[All,1;;7]];
2 lables=train[[All,8]];
```

得到降维后的数据

```
1 reduced=
2 (KarhunenLoeveDecomposition[Transpose[features]]
3 [[1]]//Transpose)[[All,1;;2]];
```

求出主向量

```
1 {b, m} =
2 KarhunenLoeveDecomposition[Transpose[features],
3     Standardized -> False];
```

训练集在两个最主成分构成的特征空间中的分布图（有色）：

```
1 ListPlot[{point2,point1},PlotRange->
2 {Full,All},AxesLabel->{"第一主成分","第二主成
3 分"},PlotLabel->"训练集在两个最主成分构成的特征空间中的分
4 布",PlotLegends->{0,1}]
```

计算协方差矩阵，

```
1 Covariance[features]
```

计算协方差矩阵的特征值

```
1 eig=Covariance[features]//Eigenvalues;
```

## 第二问的程序

在某个点(x,y)处以r为窗口宽度，计算某个点群point的程序：

```
1 count[x_,y_][r_,point_]:=If[(x-r)<#[[1]]<(x+r)&&(y-r)
  <#[[2]]<(y+r),1,0]&/@point//Total)
```

对于某个区域已经数出来的两种点的个数，我们可以这样计算明显度函数：

```
1 obvious[x_,y_]:=If[
2 x!=0||y!=0,Abs@(x-y)/(x+y),0.3
3 ]//N
```

计算区域上的明显度数组，

```
1 p[r_,h_,point_]:=ParallelTable[{x,y,count[x,y]
  [r,point]},{x,0,2.5,h},{y,0,2,h}];
2 list[r_,h_]:=MapThread[{#1[[1]],#1[[2]],Obvious[#1[[3]]
  ,#2[[3]]]}&,{p[r,h,point1],p[r,h,point2]},2];
3 contour=Flatten[list[0.05,0.1],1];
```

绘制图像：

```
1 contour=Flatten[list[0.05,0.1],1];
2
3 con=ListContourPlot[contour,PlotLegends->Automatic,Contours->10,PlotLabel->"明显度函数的等高
  图",FrameLabel->{"第一主特征","第二主特征"}]
4
5 lis=ListPlot[{point1,point2},PlotStyle->
  {Green,Red},AxesLabel->{"第一主成分","第二主成
  分"},PlotLabel->"训练集在两个最主成分构成的特征空间中的分
  布",PlotLegends->{0,1}]
6
7 show=Show[con,lis,PlotLabel->"点集与明显度函数对比图"]
```

## 第三问的程序

数据降维

```
1 transformedtest = Transpose[m.Transpose[test]][[All, 1
  ;; 2]];
```

作图

```

1 compare =
2   ListPlot[{point1, point2, transformedtest},
3     PlotStyle -> {Green, Blue, Red}, PlotLegends -> {1,
4       0, "测试集"},
5     PlotLabel -> "训练集和测试集对比", AxesLabel -> {"第一主
      向量", "第二主向量"}]

```

## 训练线性核

```

1 model2 = Classify[asso2,
2   Method -> {"SupportVectorMachine", "KernelType" ->
   "Linear"}];

```

## 作图

```

1 linearplot=Show[dens,ListPlot[{point1,point2},PlotStyle
->{Yellow,Blue,Red},PlotLegends->{1,0}],PlotLabel->"线
性核下的训练结果",FrameLabel->{"第一主向量","第二主向量"}]

```

```

1 linearplot2=Show[dens,plot,PlotLabel->"线性核下的训练结
果",FrameLabel->{"第一主向量","第二主向量"}]

```

## 训练RBF核

```

1 modelRBF=Classify[asso2,Method->
{"SupportVectorMachine","KernelType"-
>"RadialBasisFunction"}];

```

## 作图

```

1 RBFplot=Show[dens2,ListPlot[{point1,point2},PlotStyle->
{Yellow,Blue,Red},PlotLegends->{1,0}],PlotLabel->"高斯径
向基函数核下的训练结果",FrameLabel->{"第一主向量","第二主向
量"}]

```

```

1 RBFplot2=Show[dens2,plot,PlotLabel->"高斯径向基函数核下的
训练结果",FrameLabel->{"第一主向量","第二主向量"}]

```

## 训练多项式核

```

1 modelPolynomial=Classify[asso2,Method->
{"SupportVectorMachine","KernelType"->"Polynomial"}];

```

## 作图

```
1 polynomialplot=Show[densPolynomial,ListPlot[{point1,point2},PlotStyle->{Yellow,Blue,Red},PlotLegends->{1,0}],PlotLabel->"多项式核下的训练结果",FrameLabel->{"第一主向量","第二主向量"}]
```

```
1 polynomialplot2=Show[densPolynomial,plot,PlotLabel->"多项式核下的训练结果",FrameLabel->{"第一主向量","第二主向量"}]
```