实验一 概率密度估计--Parzen窗和k最近邻方法研究

姓名:文荟俨 学号:19S103256

一、实验目的

在之前的模式识别研究中，我们假设概率密度函数的参数形式已知，即判别函数的参数是已知的。本实验使用非参数化的方法来处理任意形式的概率分布，而不必事先考虑概率密度的参数形式。在模式识别中有多种令人感兴趣的非参数化方法， Parzen窗估计和k最近邻估计就是两种经典的估计法。我们希望通过实验加深对它们的理解。

二、实验计划

本实验采用Python编程，主要分为三部分:生成随机样本、Parzen窗估计和KNN估计。

其中，生成随机样本采用numpy.random.randn函数，它的功能是产生标准正态分布，向其传入参数(*a*, *b*)，可生成*a*个*b*维的正态分布样本。生成(*μ*, *σ*^2)的正态分布函数操作如下: numpy.random.randn(*a*)\* *σ*^2+ *μ*。



图1 Parzen窗算法流程图

Parzen窗估计采用高斯分布函数作为窗函数，它的方法是:根据确定的体积函数Vn = h/sqrt(n)来逐渐收缩一个给定的初始区间，其算法流程如图1所示。KNN则是先确定kn为n的某个函数，如kn=sqrt(n)，这样体积需要逐渐生长，直到最后能包含进x的kn个相邻点。

三、实验过程

1.绘制初始分布

样本的初始分布如图3所示。

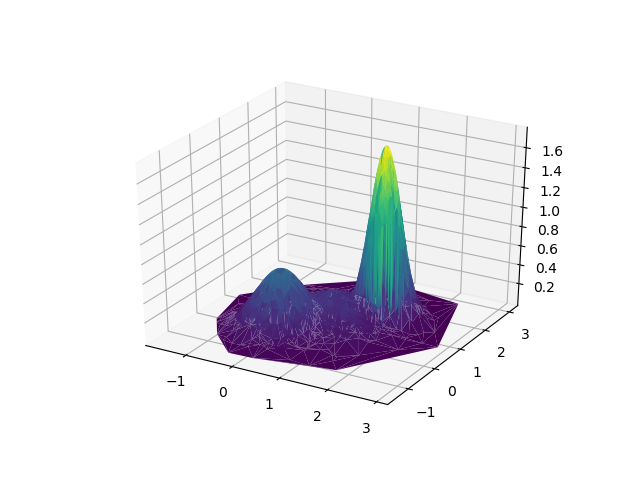


图3 样本初始概率密度分布

进一步，将sample.txt中的样本点打乱，得到sample\_shuffle.txt，对其按照2:1划分训练集和测试集。样本点的可视化分布情况如图4所示。其中红绿蓝三种颜色点分别对应三类样本，可观察到样本间存在一定交叉。

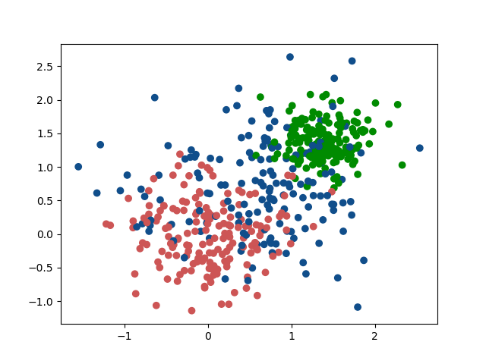
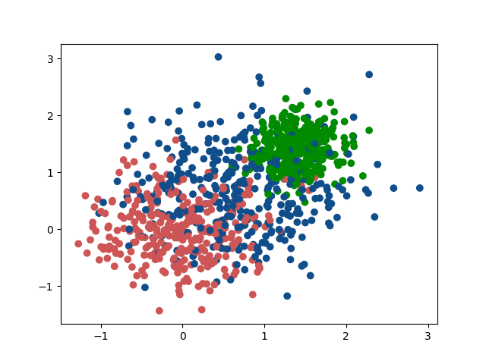


图4 样本点可视化二维坐标点图(左为训练集，右为测试集)

2. 编写函数

各部分代码具体说明如表1所示。

表1 代码功能模块说明表

|  |  |  |
| --- | --- | --- |
| 方法名 | 所属文件 | 功能 |
| generateDate | parzen\_main.py | 生成样本 |
| ShowData | parzen\_main.py | 绘制样本点 |
| dist | parzen\_main.py | 计算L2距离 |
| gaussCore | parzen\_main.py | 高斯核函数 |
| PN | parzen\_main.py | Pn公式计算 |
| calculate\_acc | parzen\_main.py | 准确率计算 |
| inference | parzen\_main.py | 样本点预测 |
| main | parzen\_main.py | 主函数 |
| calc\_statistics | initia\_distribution.py | 计算方差、均值 |
| main | initia\_distribution.py | 绘制初始分布 |
| main | shuffle\_samples.py | 打乱样本 |
| knn | knn.py | KNN计算 |

3.对比试验

测试样本固定为500个，不断调整训练样本N值和Parzen窗的宽度h。

四、结果分析

实验结果如表2和图5-图18所示。

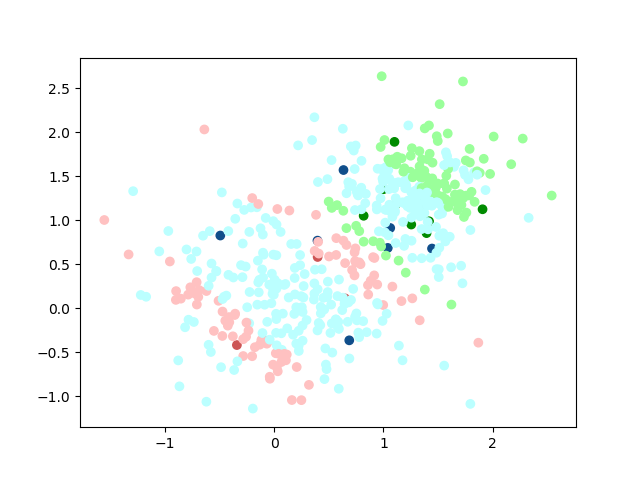
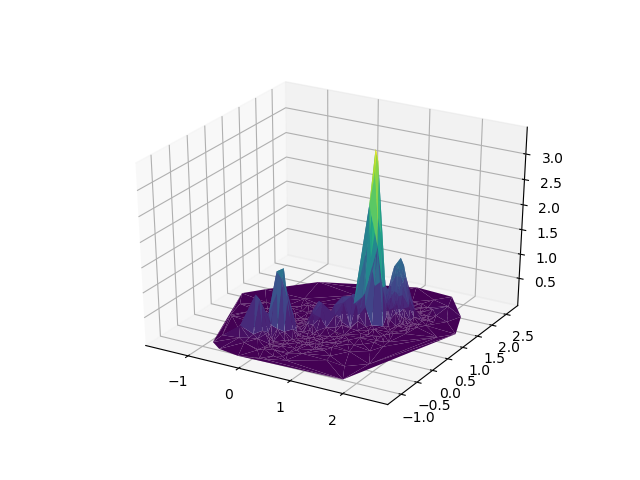


图5 实验结果1(N=20, h=0.5, accuracy=0.454)

说明: 左为概率密度分布图，右为样本点可视化坐标图。其中，较深色点为训练集样本，浅色点为测试集样本，它的颜色是经过Parzen窗估计的类别，可观察到，预测的样本类别大致围绕着训练样本类别。 (下同)

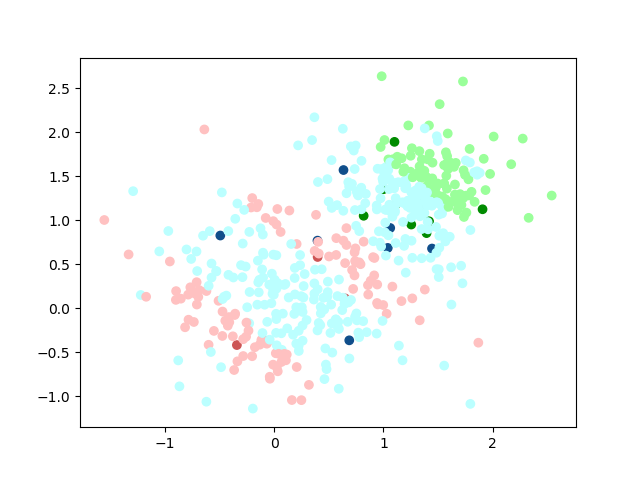
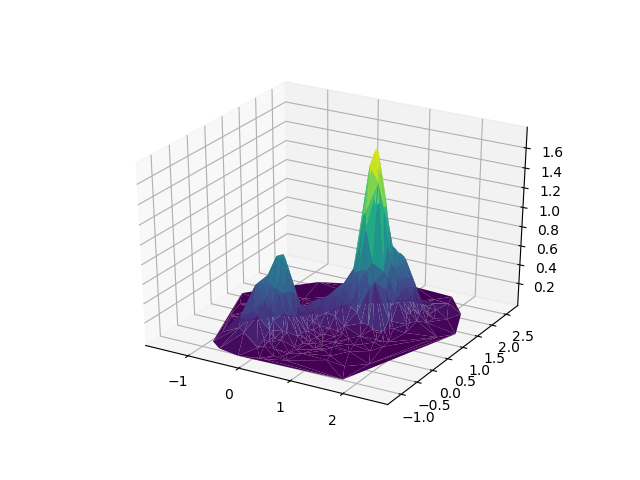


图6 实验结果2(N=20, h=1, accuracy=0.502)

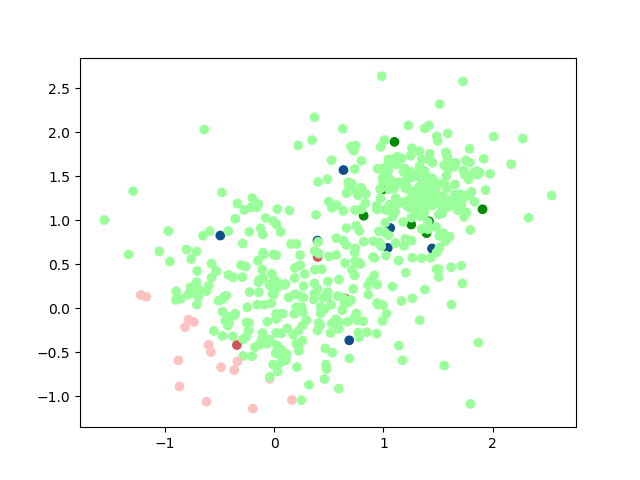
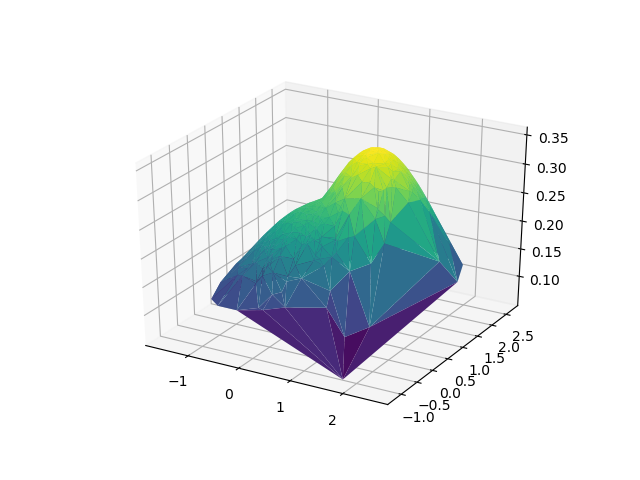


图7 实验结果3(N=20, h=5, accuracy=0.352)

说明:由于h较大，概率密度图将样本主要分为了1类，从右侧的可视化坐标图可以得到验证，几乎所有的测试样本都被预测为了绿色，仅有少数为粉色。

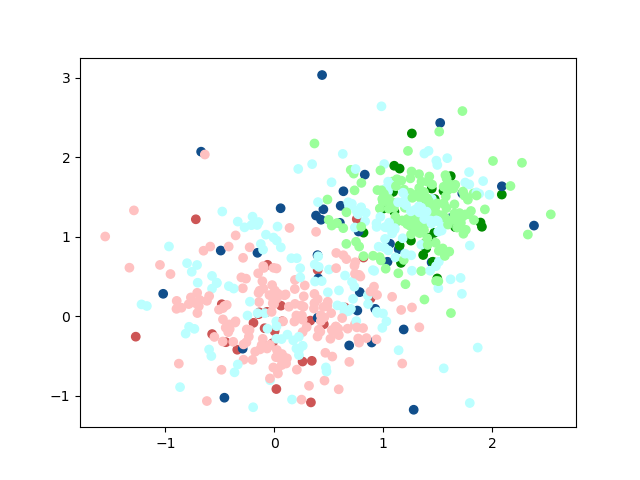
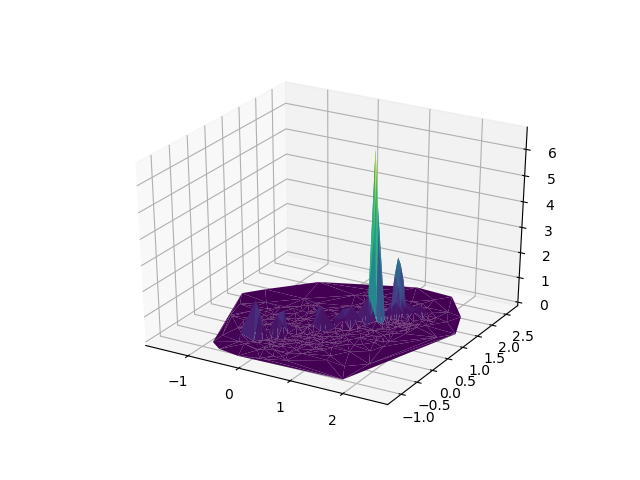


图8 实验结果4(N=100, h=0.5, accuracy=0.58)

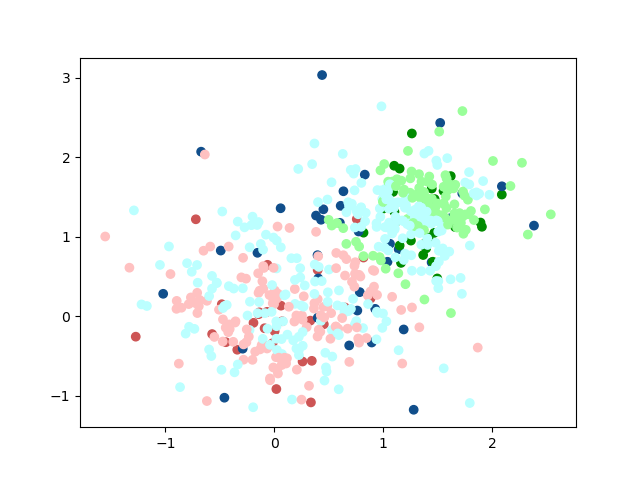
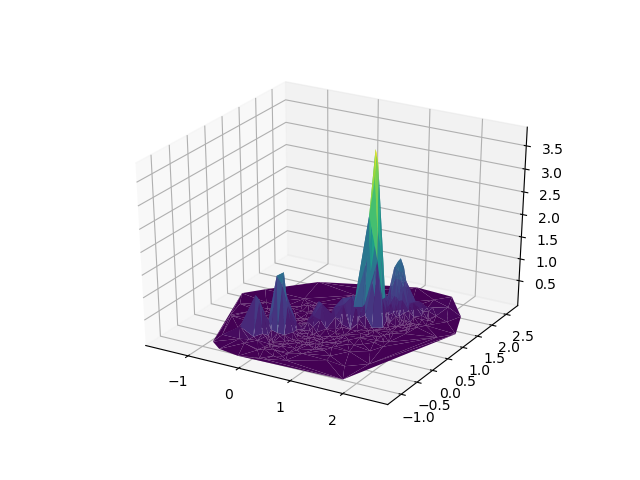


图9 实验结果5(N=100, h=1, accuracy=0.516)

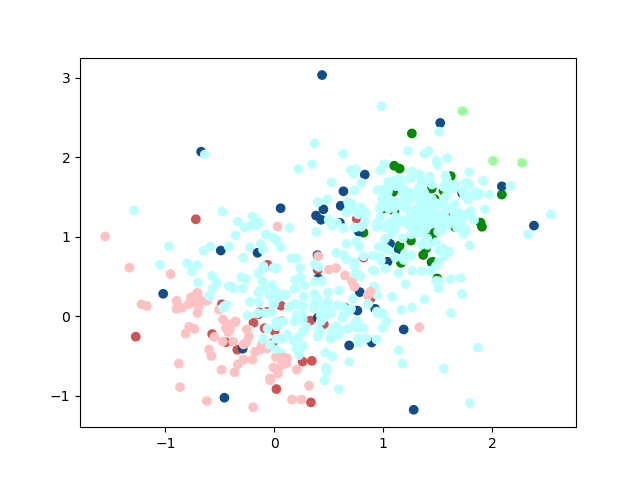
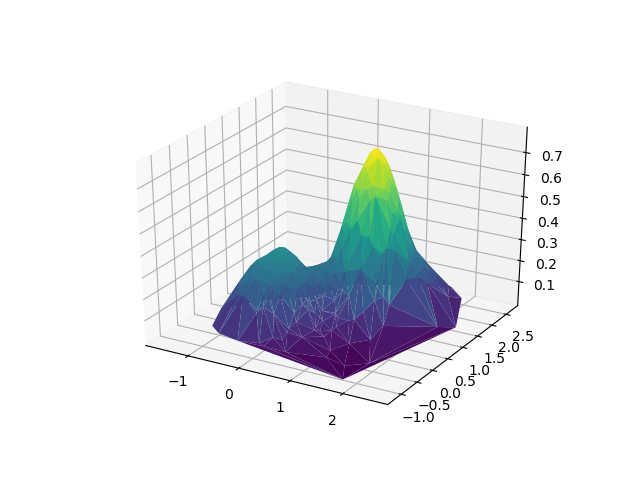


图10 实验结果6(N=100, h=5, accuracy=0.41)

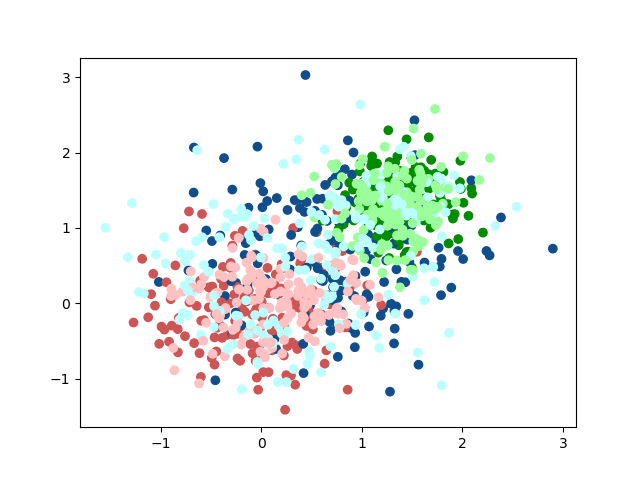
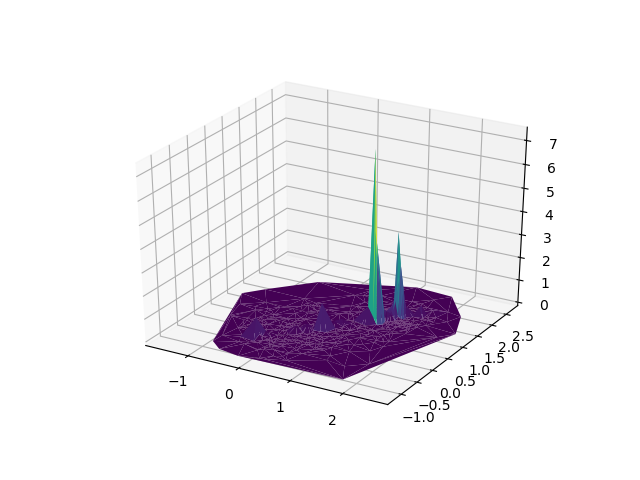


图11 实验结果7(N=500, h=0.5, accuracy=0.59)

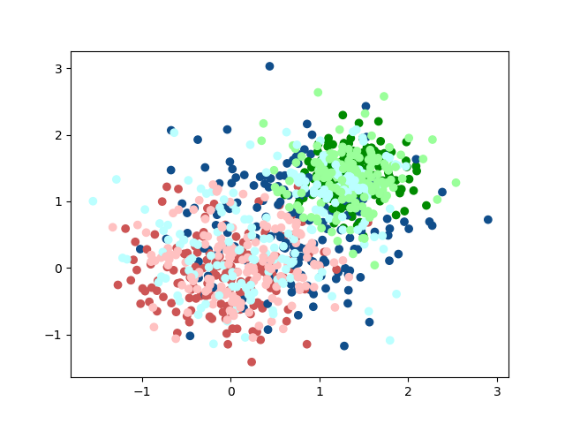
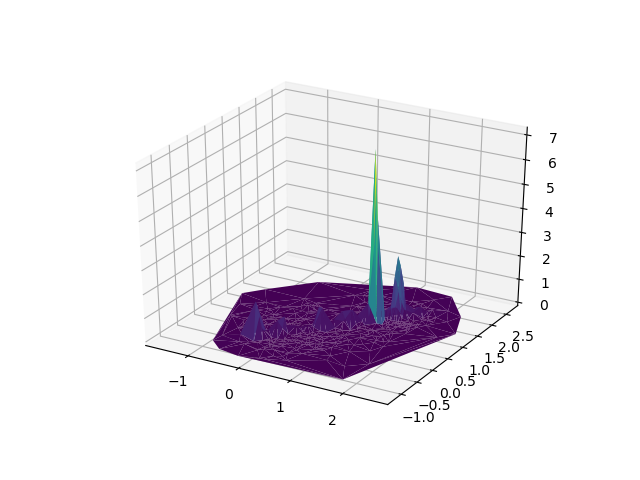


图12 实验结果8(N=500, h=1, accuracy=0.558)

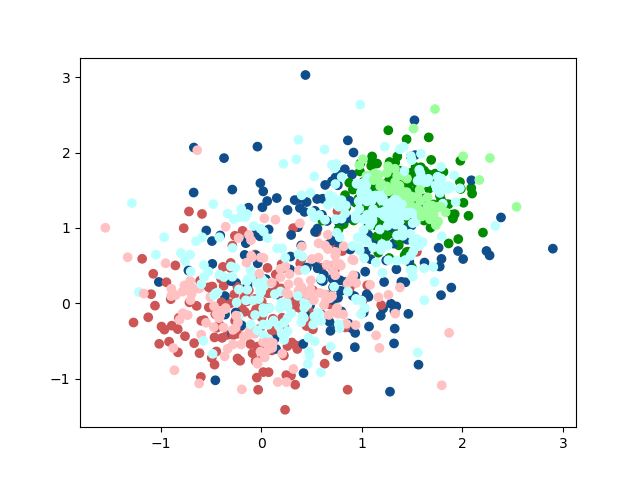
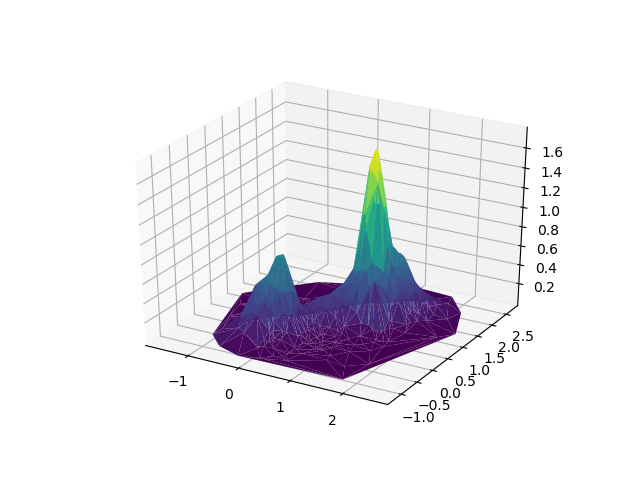


图13 实验结果9(N=500, h=5, accuracy=0.484)

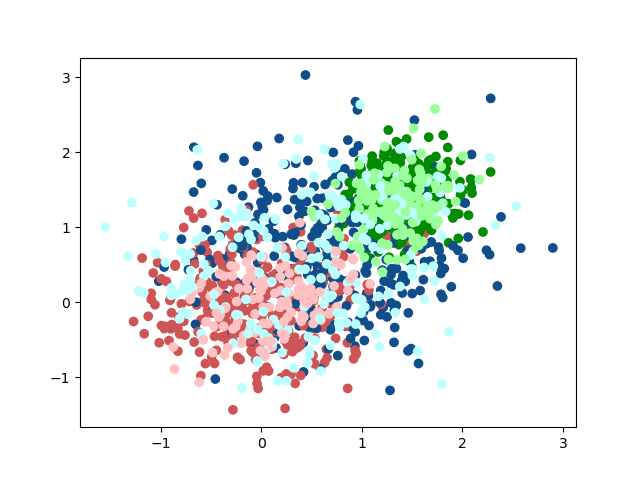
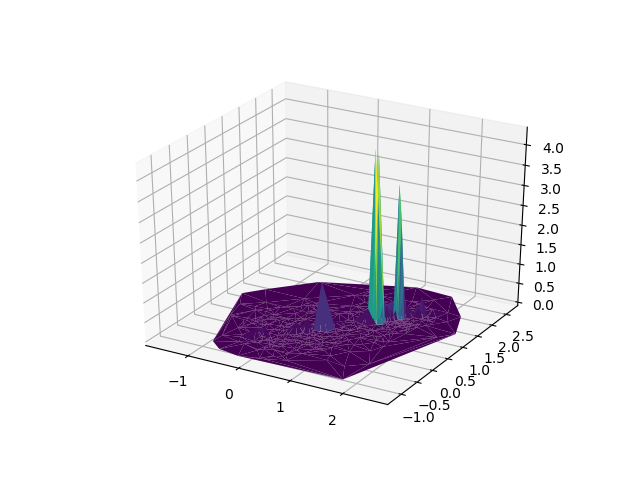


图14 实验结果10(N=1000, h=0.5, accuracy=0.6)

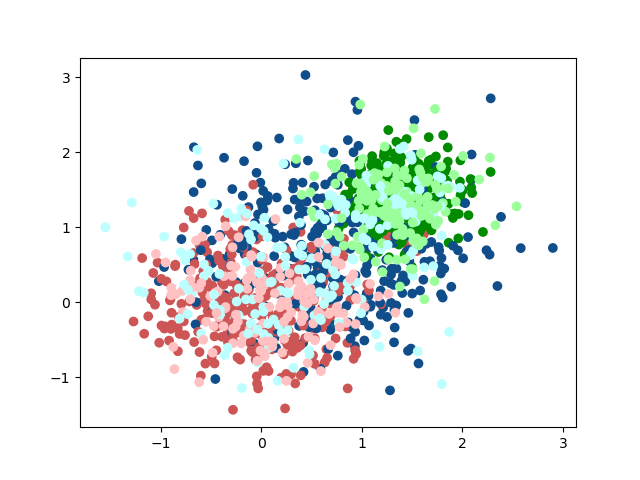
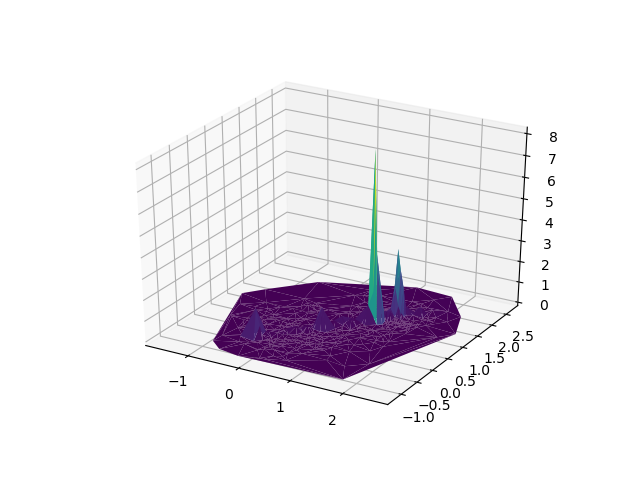


图15 实验结果11(N=1000, h=1, accuracy=0.542)

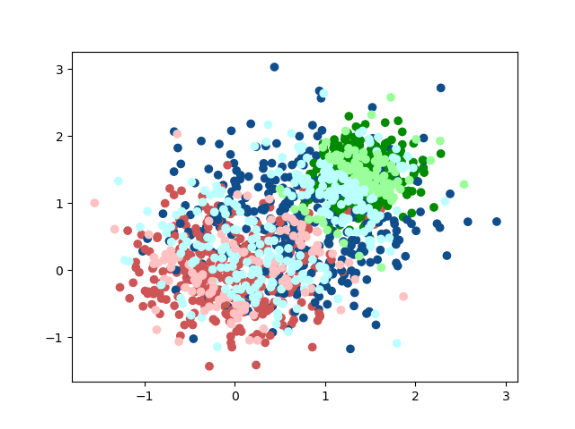
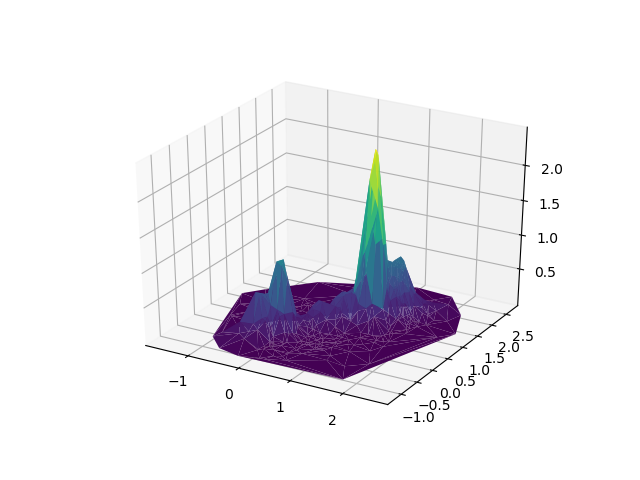


图16 实验结果12(N=1000, h=5, accuracy=0.508)

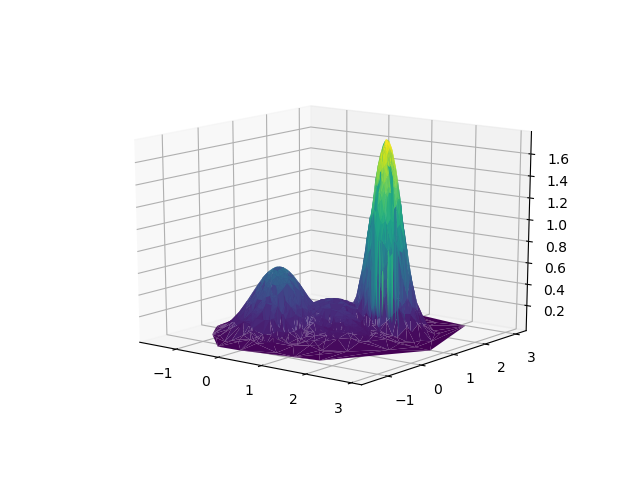


图17 实验结果12(N=∞, h=任意值)

表2 当N=1000时, 不同h的Parzen窗准确率表

|  |  |
| --- | --- |
| h | accuracy |
| 0.05 | 0.352 |
| 0.1 | 0.402 |
| 0.2 | 0.518 |
| 0.3 | 0.56 |
| 0.5 | 0.6 |
| 1.0 | 0.542 |
| 1.5 | 0.562 |
| 5 | 0.508 |
| 10 | 0.444 |

图18 当N=1000时, 不同h的Parzen窗准确率曲线

5.KNN实验结果分析

KNN的实验结果如表3和图19所示。

表3不同N和k取值下的KNN准确率表

|  |  |  |
| --- | --- | --- |
| N | k | Accuracy |
| 20 | 3 | 0.572 |
| 20 | 5 | 0.672 |
| 20 | 10 | 0.504 |
| 100 | 3 | 0.722 |
| 100 | 5 | 0.78 |
| 100 | 10 | 0.75 |
| 500 | 3 | 0.76 |
| 500 | 5 | 0.768 |
| 500 | 10 | 0.78 |

图19 当N=1000时, 不同k的KNN准确率曲线