lab2. nonparametric method of density estimation

该实验中实现了概率密度函数估计的非参数方法，包括K_N近邻估计法和Parzen窗法。

原理

项目结构

```
.
├── data                    # dataset
│   ├── sampled_data_16.xlsx
│   ├── sampeld_data_256.xlsx
│   ├── sampeld_data_1000.xlsx
│   └── sampeld_data_2000.xlsx
├────── main.py             # project entry
├────── PdfEstimation.py    # PdfEstimation class
├────── ref.py              # some reference code, source is commented in file
└────── test.py             # some test code
```

原理介绍

通过定义了PdfEstimation类，抽象了一个进行概率密度函数估计的对象。该对象的成员变量包括

- sample: 输入的样本
- pdf: 概率密度函数的估计，用一个np数组表示.

该对象的成员函数包括

- fit: 传入不同方法(K_N近邻法neighor和Parzen窗法 kernel), 进行模型的拟合.

- neighbor: K_N近邻法. 通过调用函数 `get_elem(sample, idx, num)` 实现.
- kernel: Parzen窗法. 通过调用函数 `uniform(x, y, band_width)` 和 `gaussian(x, y, band_width)` 实现.

具体的实现和代码注释见代码文件.

运行过程和结果

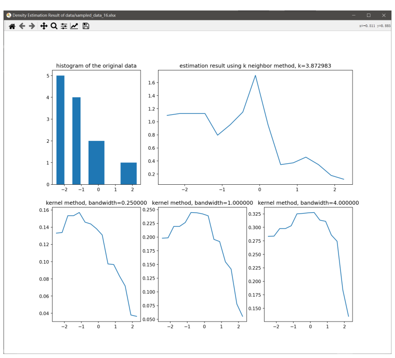
代码入口为 `main.py` 文件，修改 `data_path` 变量来修改数据集。修改数据集后运行该代码文件即可。

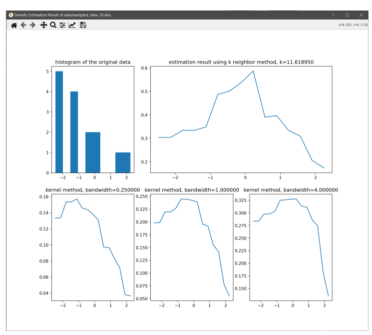
K_N近邻法的小舱大小选为 $\sqrt{N}$ 和 $3\sqrt{N}$.

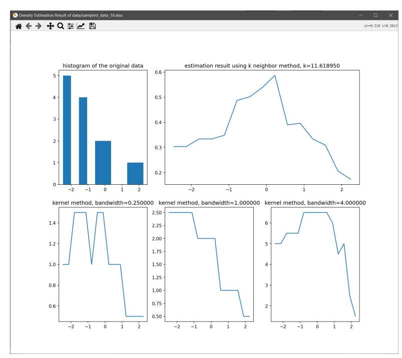
前两个图像使用高斯窗，后一个图像使用方窗.

前一个图像的$K_N = \sqrt{N}$, 后两个图像的$K_N = 2\sqrt{N}$.

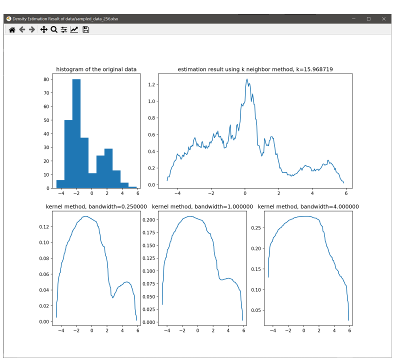
case #1. data_path=data/sampled_data_16.xlsx

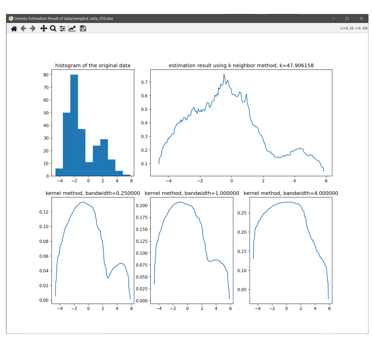


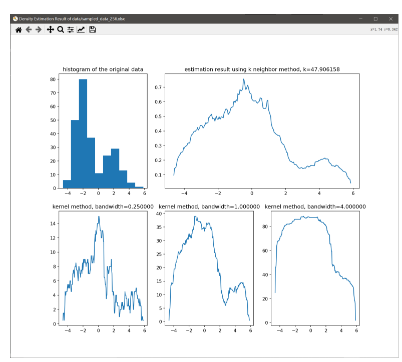




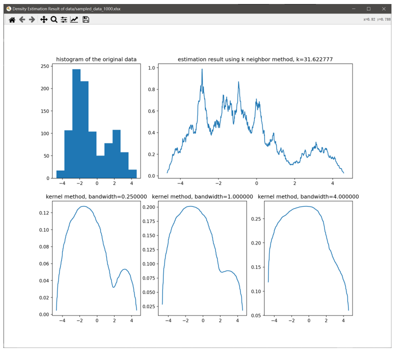
case #2. data_path=data/sampled_data_256.xlsx

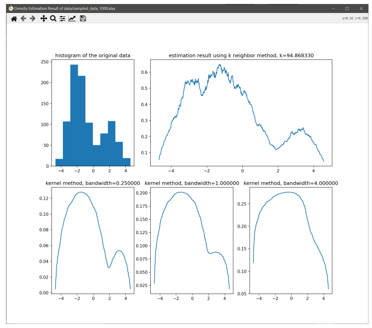


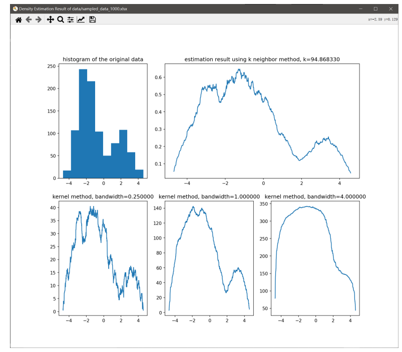




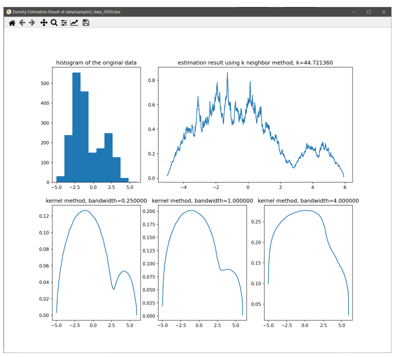
case #3. data_path=data/sampled_data_1000.xlsx

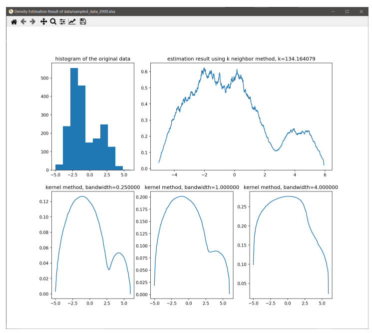


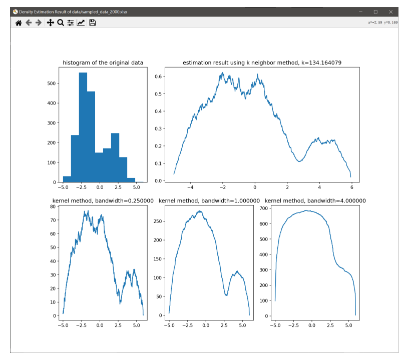




case #4. data_path=data/sampled_data_2000.xlsx







结果分析

根据实验结果分析，对于 K_N近邻法

1. 在相同数量的数据集中，小舱的大小越大，估计的结果变得更加平滑，毛刺减少。
2. 在相同的小舱大小中，数据集的数量越大，估计结果与直方图更接近，准确率更高。

对于Parzen窗法

1. 在相同的窗大小，相同数量的数据集中，高斯窗的效果好于方窗。
2. 在相同的窗种类，相同数量的数据集中，窗的大小适中的情况较好，窗较小时噪声过大，窗较大时会丧失部分信息。
3. 在相同的窗的种类和大小的情况下，数据集的大小越大，预测效果越准确。

实验中遇到的问题

如何在一个figure内plot多个axes对象？如何进行 ~~不规则的~~ 图像的绘制

使用subplot2grid方法，其中包括loc, colspan, rowspan参数，可进行axes在不同位置放置。。函数原型为 `subplot2grid(shape, loc, rowspan=1, colspan=1, fig=None, **kwargs)`.

`**kwargs` 关键字

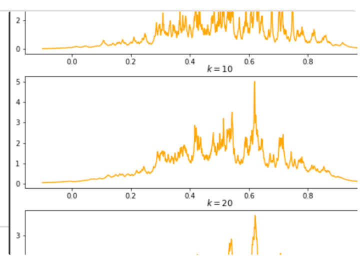
项目代码中PdfEstimation中两个成员函数fit和kernel用到了HoF, 在二者的调用关系中遇到一些参数数量不相等导致代码逻辑混乱的情况，目前采用的方式是传入一个list对象实现。可以考虑使用kwargs参数来增强代码的逻辑和可读性.

> 4. More Control Flow Tools
>
> As well as the while statement just introduced, Python uses a few more that we will encounter in this chapter. if Statements: Perhaps the most well-known statement type is the if statement. For exa...
>
> https://docs.python.org/3/tutorial/controlflow.html#keyword-arguments

REF

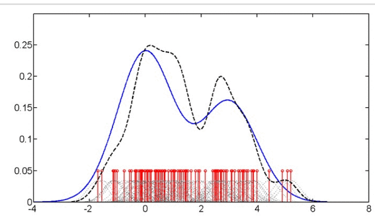
code ref

> Non Parametric Density Estimation Methods in Machine Learning - GeeksforGeeks
>
> A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and programming articles, quizzes and practice/competitive programming/company interview Questions.
>
> https://www.geeksforgeeks.org/non-parametric-density-estimation-methods-in-machine-learning/



theory ref

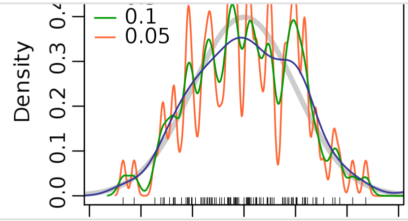
> Density estimation
>
> In statistics, probability density estimation or simply density estimation is the construction of an estimate, based on observed data, of an unobservable underlying probability density function. The unobservable density function is thought of as the
>
> https://en.wikipedia.org/wiki/Density_estimation



Kernel density estimation

In statistics, kernel density estimation (KDE) is the application of kernel smoothing for probability density estimation, i.e., a non-parametric method to estimate the probability density function of a random variable based on kernels as weights. KDE

W https://en.wikipedia.org/wiki/Kernel_density_estimation#Definition



Kernel (statistics)

The term kernel is used in statistical analysis to refer to a window function. The term "kernel" has several distinct meanings in different branches of statistics.

W https://en.wikipedia.org/wiki/Kernel_(statistics)#Kernel_functions_in_common_use