Assignment1 Report

代码组织

共提交三个python文件，source.py、assessment.py、main.py。
source.py 中提供了三种非参估计方法的绘图函数
assessment.py 给出了一些超参选择后的量化评价函数
main.py调用以上两个python文件所提供的接口，组织 sample_data，
以回答问题和探究。（main.py中所绘的图与报告中所用的图对应）

一、 sample data 大小对算法结果的影响

因为篇幅问题不将图片附在报告里，代码里可以调用 main.py 中的 sample_data_influence()函数可以得到 4x3 张图片。

根据大数定理可以知道，采样数据越多，频率就越接近概率。所以总的来说，sample data 的数目对三种方法影响的总趋势都是，数据量越大，分布越接近真实分布。

对于 histogram，sample data 数量越大，数据就越光滑，不会因为 bin 过大而造成剧烈变化。

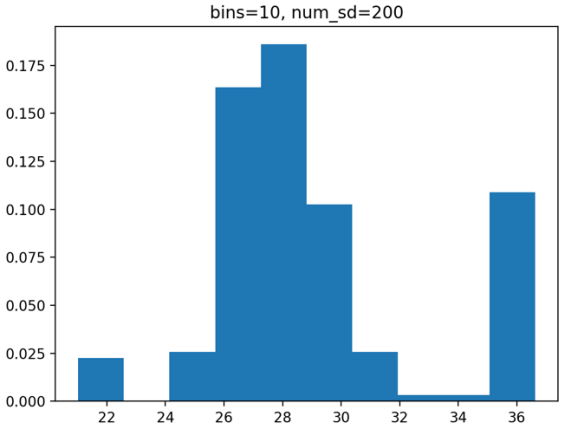
因为 KDE 曲线本身就比较光滑，所以 sample data 数量对其曲线形式影响分布较小。

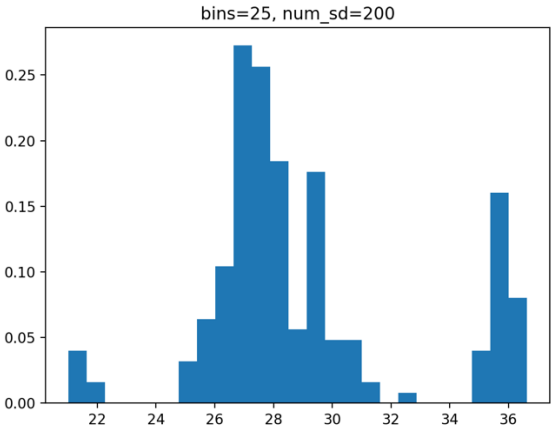
Knn 公式中的 V 会因为 sample data 数量的增大而变小，所以从图中也可以看出，当数据量增大时，其抖动的幅度就越剧烈（在 k 不变的情况下）

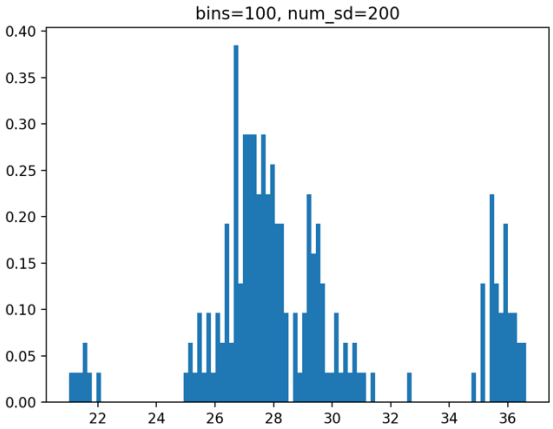
二、 histogram 确定 bins 的数量（num_SD = 200）

histogram 估计方法较为简单清晰，而且实现方式比较简单，但没有直接的数学量化评估方法。

通过对 histogram 的 bins 数目进行随机性的测试后可以得到一些经验性的结果。当 bins 过少的时，绘出的图象细节过少。而当 bins 的数目过多时，可以保留很多细节，但是绘出的直方图会因为样本数据间的间隙以及一些噪声点的存在而产生剧烈的变化。**对于 histogram 估计最简单直接的方法就是目测绘图结果，当直方图保有足够多的信息，而且受到噪声影响较小时的 bins 数量就是较优的结果。**







对于 bins 数目的选择 Wikipedia 给出了许多经验性的估计方法，在此进行了一些实验。

Square-root choice

$$\mathrm{k} = \lceil \sqrt{n} \rceil$$

Sturges' formula

$$\mathrm{k} = \lceil \log_2 n \rceil + 1$$

Rice Rule

$$\mathrm{k} = \lceil 2n^{1/3} \rceil$$

Scott's normal reference rule

$$h = \frac{3.5\hat{\sigma}}{n^{1/3}}$$

其中 k 与 h 的关系为

$$\mathrm{k} = \left\lceil \frac{\max x - \min x}{h} \right\rceil$$

Minimizing cross-validation estimated squared error

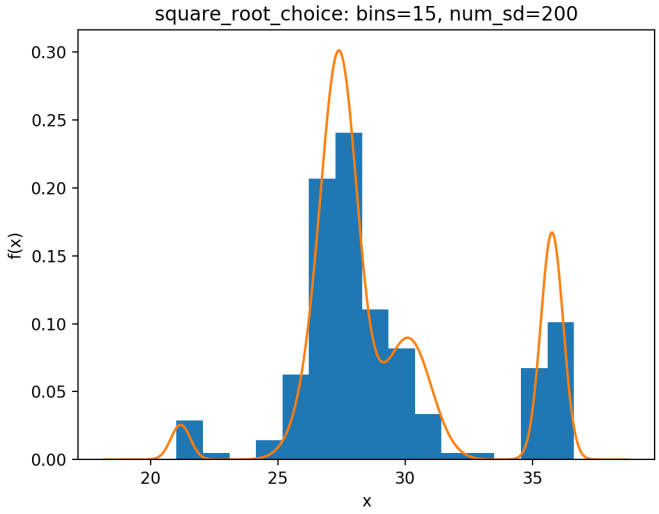
$\arg\min\left(\frac{2}{(n-1)h}\right) - \frac{n+1}{n^2(n-1)h}\sum_k N_k^2, for\ h$ *(优化不稳定，已从源码中删除)*

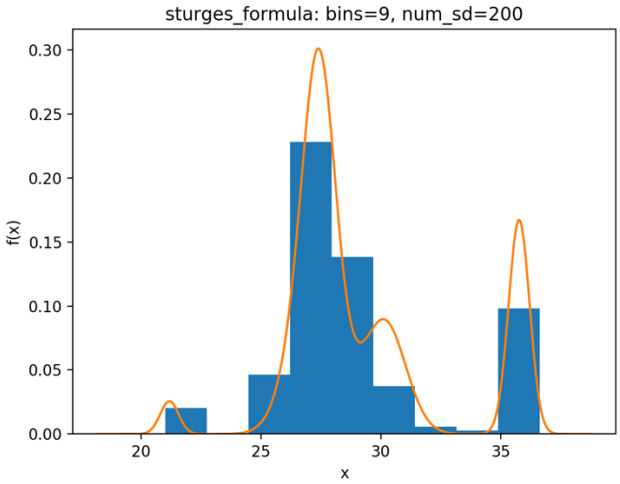
Shimazaki and Shinomoto's choice

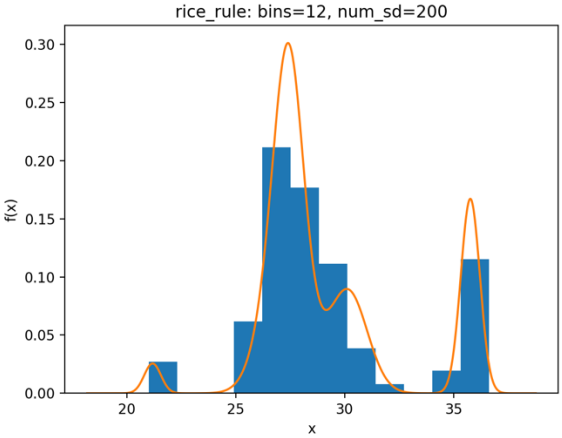
$$\arg\min \frac{2\sum_{i=1}^{k} Ni - \frac{1}{k}\sum_{i=1}^{k}(Ni - \sum_{j=1}^{k} Nj)^2}{h^2}, for\ h$$

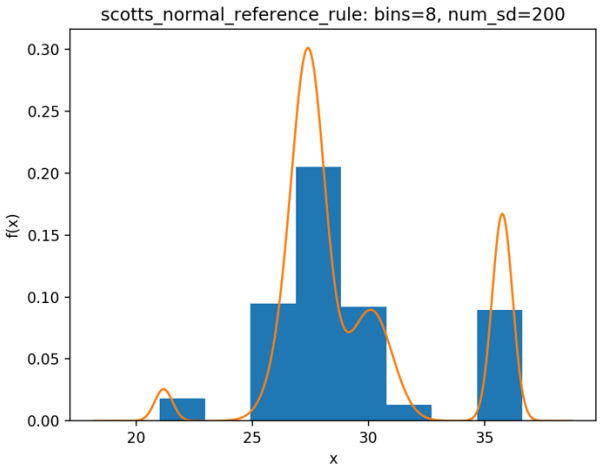
利用这些经验性估计公式，可以得到下列图象

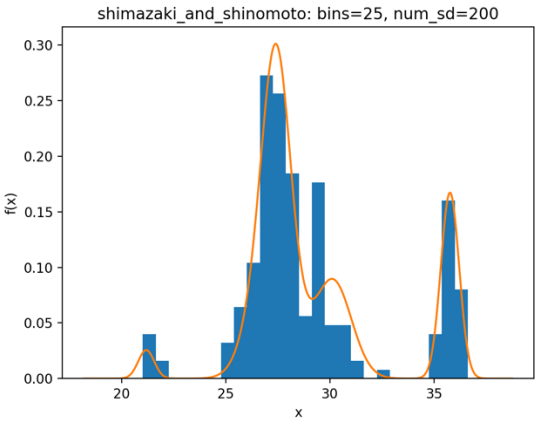
可以看出最后一种方法（Shimazaki and Shinomoto's choice）的求值效果最佳。


square_root_choice: bins=15, num_sd=200
0.30
0.25
0.20
0.15
0.10
0.05
0.00
f(x)
20
25
30
35
x


sturges_formula: bins=9, num_sd=200
0.30
0.25
0.20
0.15
0.10
0.05
0.00
f(x)
20
25
30
35
x

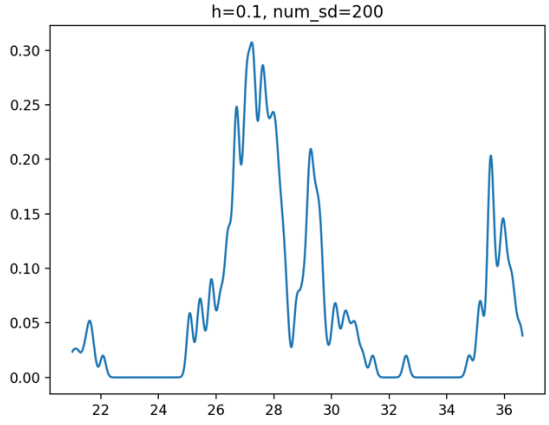

rice_rule: bins=12, num_sd=200
0.30
0.25
0.20
0.15
0.10
0.05
0.00
f(x)
20
25
30
35
x

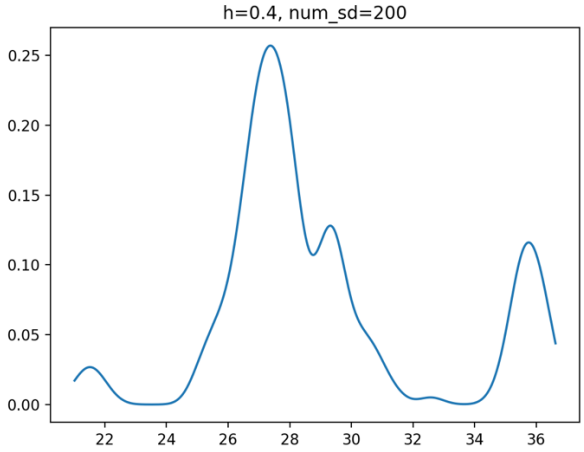

scotts_normal_reference_rule: bins=8, num_sd=200
f(x)
x
0.00
0.05
0.10
0.15
0.20
0.25
0.30
20
25
30
35

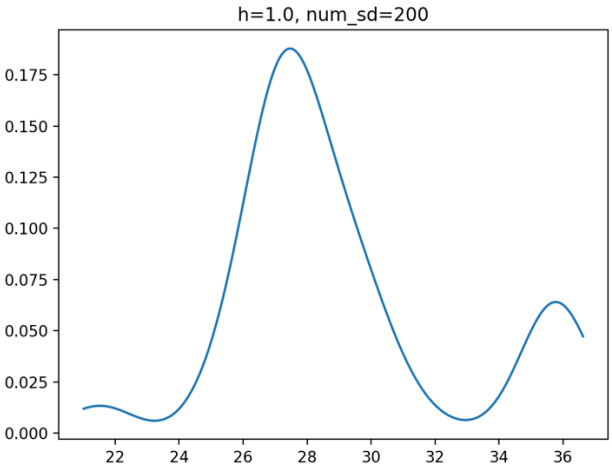

shimazaki_and_shinomoto: bins=25, num_sd=200
f(x)
x
0.00
0.05
0.10
0.15
0.20
0.25
0.30
20
25
30
35

三、　　高斯核密度估计选择最优 h

该核密度估计方法因为采用了高斯核，所以获得的概率密度曲线相对平滑。调节 h 并概率密度曲线与原分布进行比较可以看出，**当 h 比过小的时候，噪声影响较大，而且曲线拟合结果偏向于 undersmooth 的结果。当 h 过大的时候噪声点影响降得很低，但是结果会 oversmooth。目测估计时，选择使曲线平滑程度适中的 h 即可得到较优的结果。**







核密度估计有很多的统计学理论研究结果，Wikipedia 指出 MISE 方法(使均方误差最小)给出了最好的理论结果:

$$h_{AMISE} = \frac{R(K)^{1/5}}{m_2(K)^{1/2}R(f'')^{1/5}n^{1/5}}$$

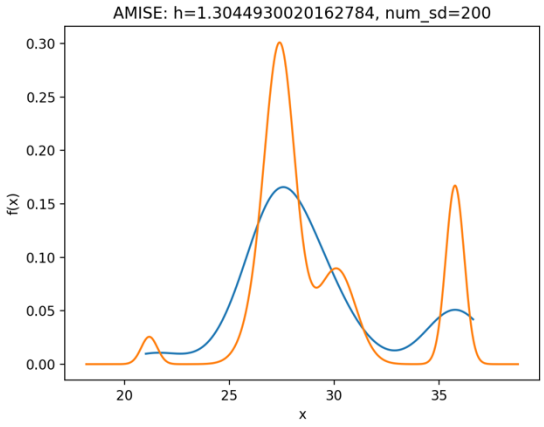
但是由于公式中要求知道原概率分布，所以无法在实际运用中使用，**但是对原函数进行极强的假设（normal distribution）时可以得到很粗糙的结果：**

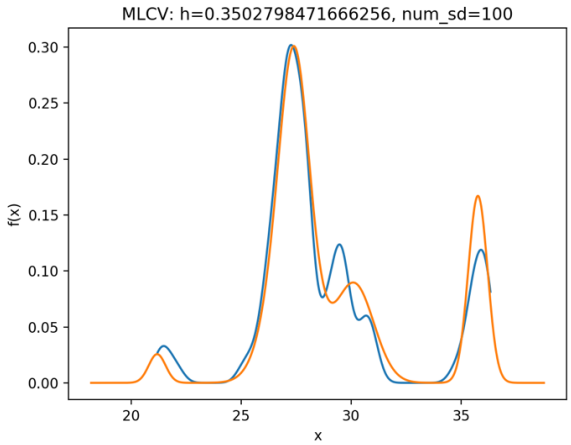
$$\mathbf{h = 1.06\hat{\sigma}n^{-1/5}}$$

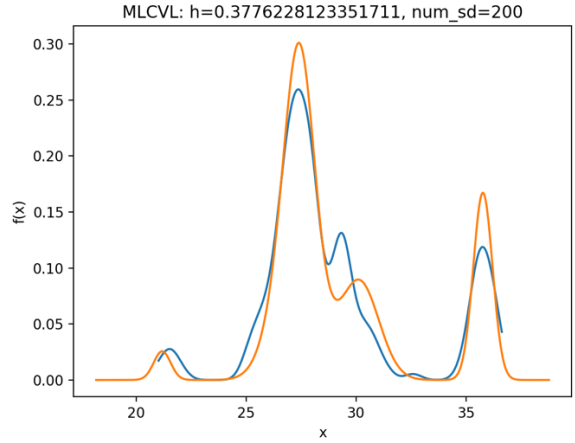
概统中似然的概念中运用了概率密度，而本方法获得的数据刚好也是其公式中所需要的数据，再利用交叉验证的方法可以避免高斯核的 h 趋于零时结果最佳的缺陷，所以实际使用时很多人采取 maximum likelihood cross validation 的方法进行量化，从而得到合适的 h，利用对数化简后获得下列公式

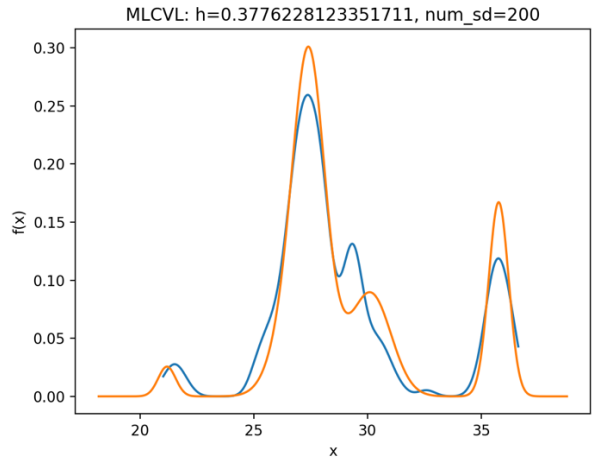
$$\arg min\left(\frac{1}{n}\sum_{i=1}^{n}\log\left[\sum_{i\neq j}K\left(\frac{xj-xi}{h}\right)\right]-\log[(n-1)h]\right), for\ h$$

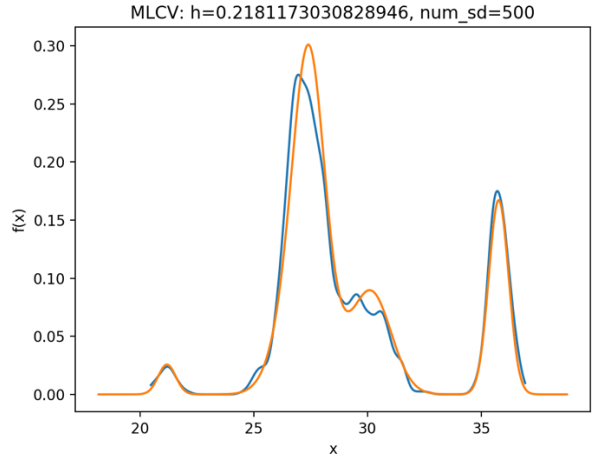
可以得到一下结果











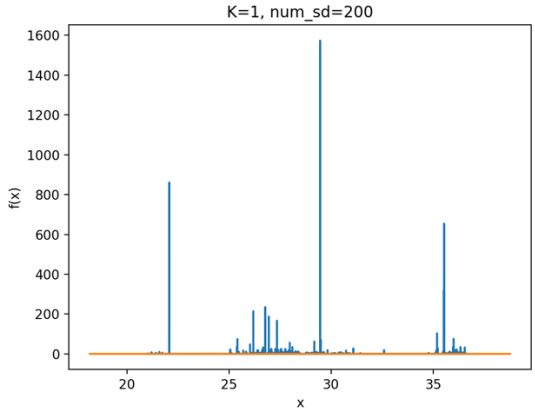
可以看出 MLCV 效果极佳。

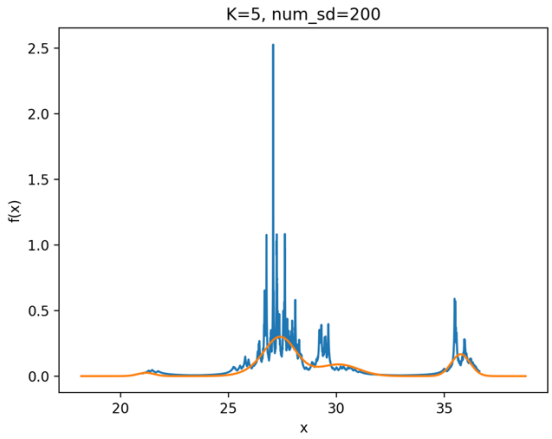
四、 KNN 密度估计探究

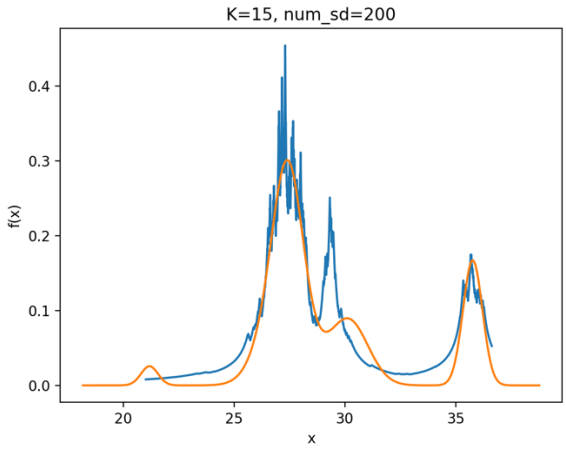
根据 KNN 密度估计公式

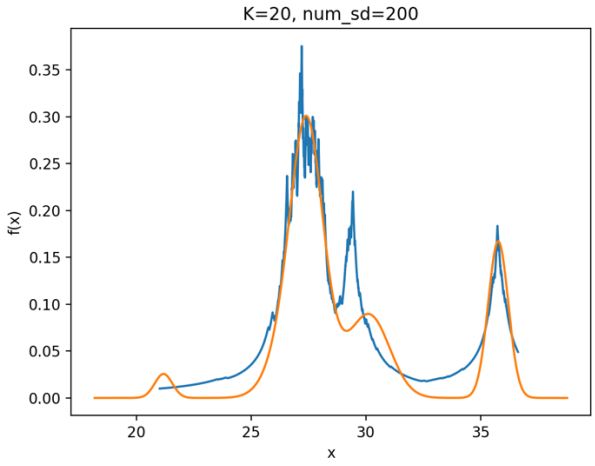
$$\mathrm{p}(x) = \frac{K}{NV}$$

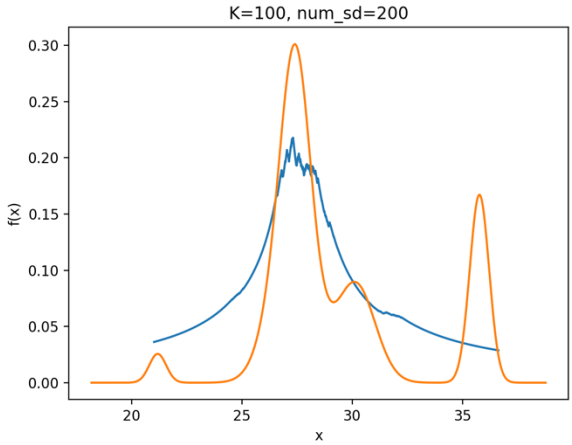
可以直观的感受，当 k 过小的时候 V 受到噪声点影响的概率就越大，图象的起伏就会越大，但是当 k 过大的时候，该概率密度就过包含过多点的信息，而导致图象变得 oversmooth。**下面给出类似 Figure2.26 的图。**











通过上图可以看出，当 k 过小的时候 undersmooth，当 k 过大时又会造成 oversmooth。

下面用数学理论证明 knn 得到的概率密度曲线积分不收敛至 1

假设 x < x_1，可以得$V = 2|x_k - x|$，则可以得到积分公式

$$\int p(x)dx \geq 2\int_{-\infty}^{x_1} \frac{K}{NV}dx = 2\int_{-\infty}^{x_1} \frac{K}{N \times 2|x_k - x|}dx = \frac{K}{N}\ln|x_k - x|\Big|_{-\infty}^{x_1} \to +\infty$$

参考网址

http://176.32.89.45/~hideaki/res/histogram.html
http://cran.irsn.fr/web/packages/kedd/vignettes/kedd.pdf
https://en.wikipedia.org/wiki/Kernel_density_estimation
https://en.wikipedia.org/wiki/Histogram
https://stats.stackexchange.com/questions/21631/what-bandwidth-should-i-use-for-my-kernel-density-estimation