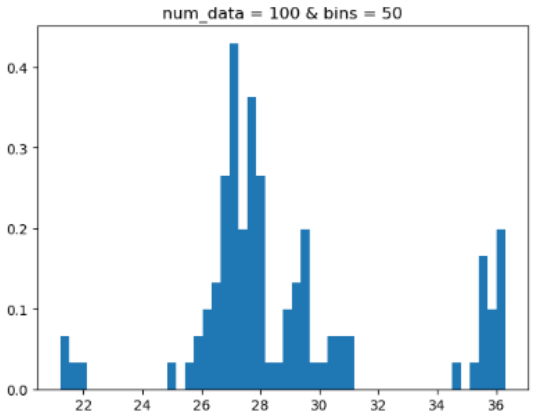
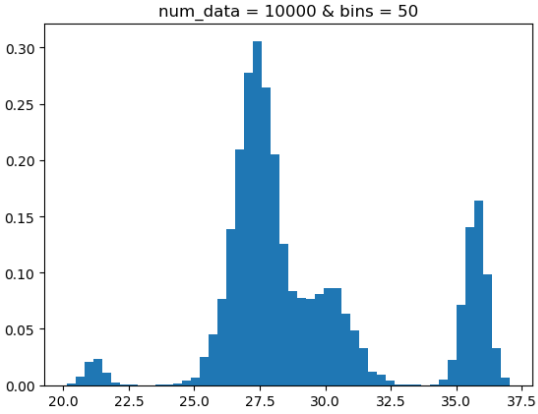
PRML Assignment 1 Report

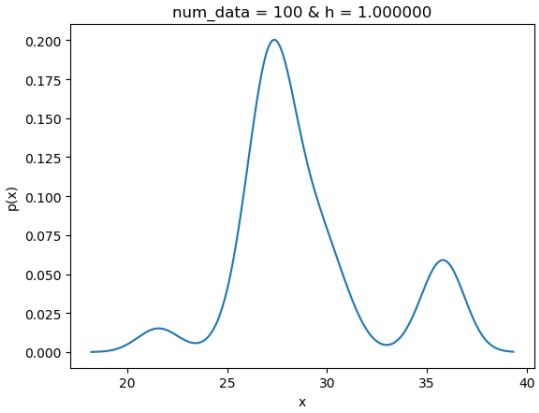
Part 1: num_data 对估计质量的影响

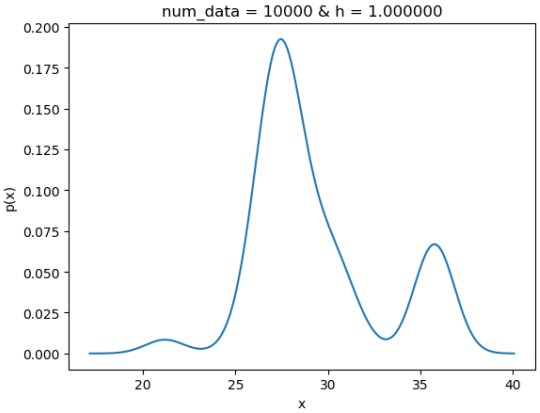
- 对于三种方法，在**固定参数 bins、h、k** 的情况下，通过不同样本数据量的对比，可以得到：当样本数据量较小时，得到的估计结果出现信息丢失，不能正确地反应分布情况；样本数据量越大，包含的信息越多，得到的估计结果越**平滑**，从而估计结果能够接近真正的概率密度函数。
- 针对于**直方图估计**，取参数 **bins = 50**：



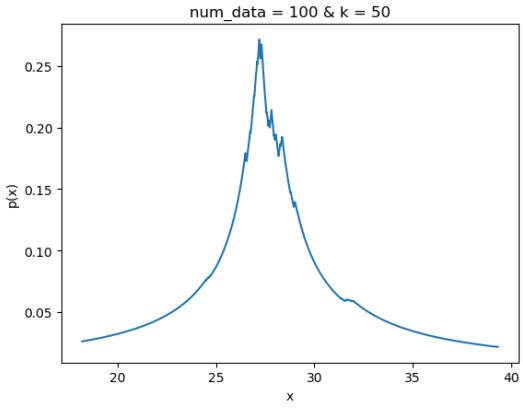


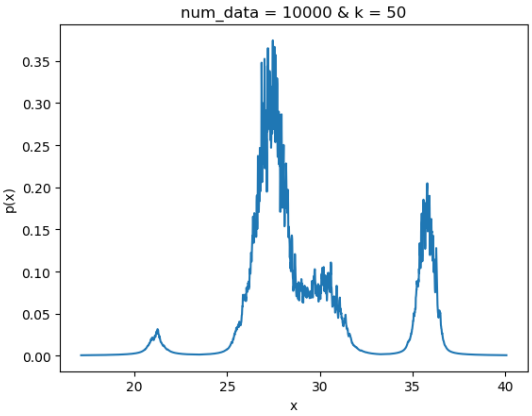
- 针对于**核密度估计**，取参数 h = 1：





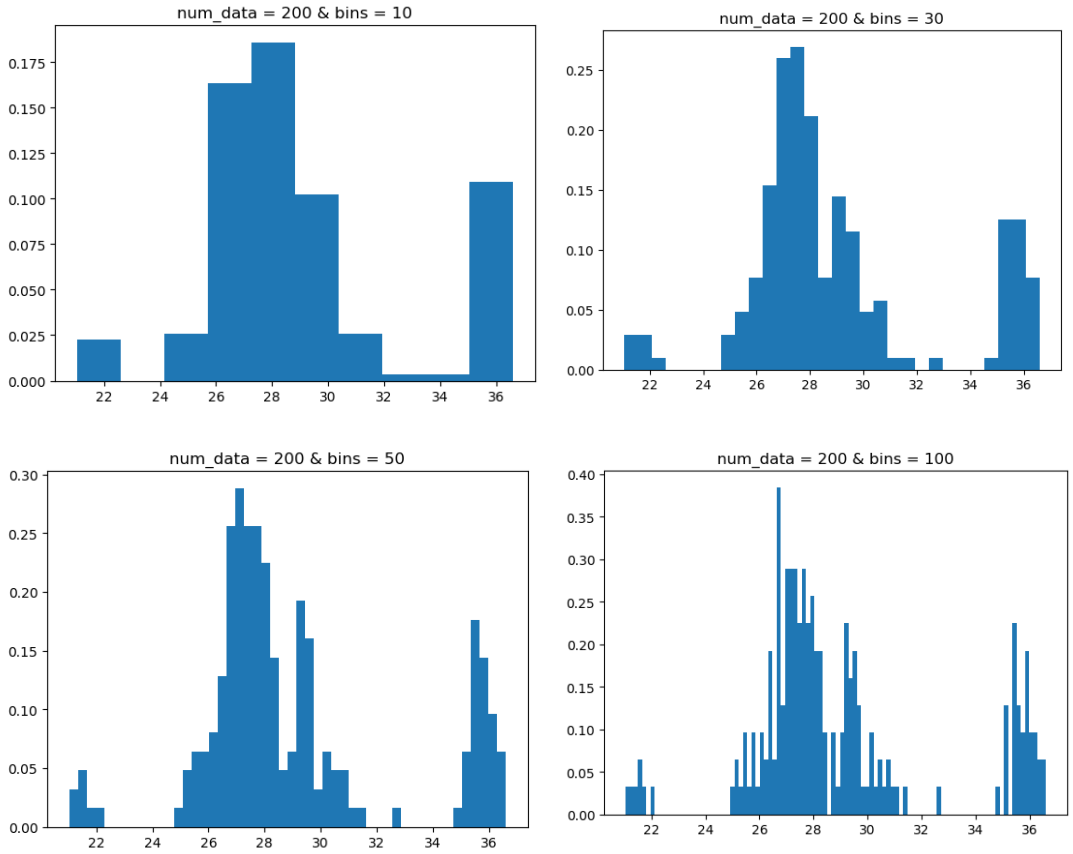
- 针对于**最近邻方法**，取参数 k = 50：





Part 2: 直方图估计

- 取 num_data = 200，分别取 bins = 10、30、50、100，得到直方图如下：



- 固定样本数据数量，根据变化 bins 的取值，可以得到：

  当参数 bins 的取值过小时（例如 bins=10），即每个 bin 内区域很大，会导致最终估计得到的概率密度函数十分粗糙，且丢失了大量信息（如峰值）；当 bins 的取值过大时（例如 bins=100），即每个 bin 内区域很小，这样会导致有些区域内完全没有或样本数量很少，得到的密度函数很不连续。

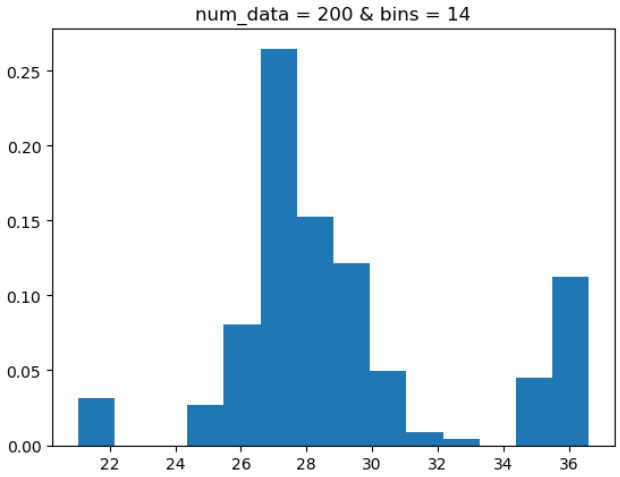
- **参数 bins 取值的选择：**

  为了选择合适的 bins 值，对于一定数量的样本数，应当保证每个区域尽可能小，同时每个区域内有充分多的样本，但每个区域内样本数只占总样本数的很小一部分，这样的条件下能够得到近似合适的 bins 取值。

  对于 bins 值的选取，有 Sturges formula 与 Doane’s formula，其中第一个公式来自于二项分布，并且隐含假设正态分布，因此无法用来选取该情况下 GaussianMixture1D 分布所需的 bins，而第二个公式是对第一个公式的修改，该公式针对于非正态分布数据改进了性能，具体计算公式如下：

$$k = 1 + \log_2(n) + \log_2\left(1 + \frac{|g_1|}{\sigma_{g_1}}\right) \qquad \sigma_{g_1} = \sqrt{\frac{6(n-2)}{(n+1)(n+3)}}$$

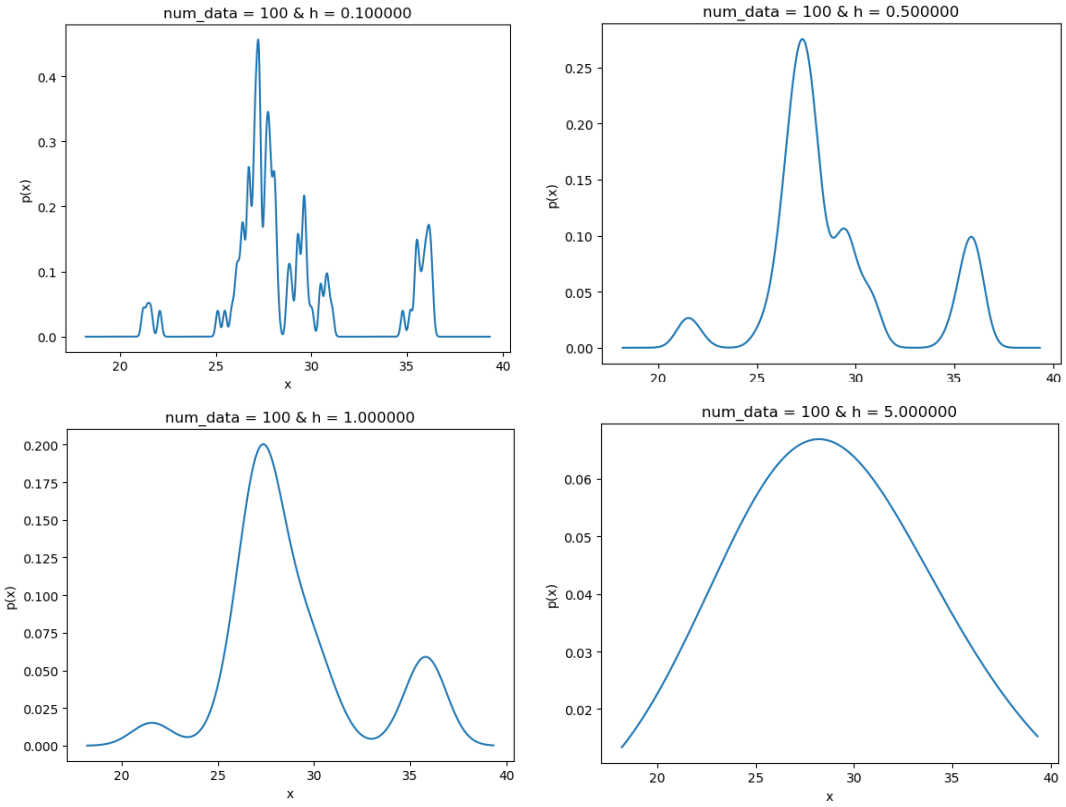
（其中 g1 为三阶偏度，利用 pandas 库 pd.Series(sampled_data).skew()）通过计算得到 g1 = 0.76016483，代入得到 k(bins)=14，对应图像为



以上为使用公式的理论结果，但实际上通过观察发现 bins=14 时并不是最接近概率分布的取值，即该方法对于该类高斯分布下的估计存在着误差。通过不断改变 bins 值，并观察图像，可以得出当 **bins 取值约为 50** 时，得出的估计曲线最接近真实的概率密度分布，当 bins=50 时的曲线见上图。

Part 3: 核密度估计

- 取 num_data = 100，分别取 h = 0.1、0.5、1.0、5.0，得到曲线如下：



- 固定样本数据数量，根据变化 h 的取值，可以得到：

  当参数 h 取值过小时（例如 h=0.1），得到的估计曲线尖锐不够平滑，并且邻域中参与拟合的点过少，很大程度影响了估计的质量；

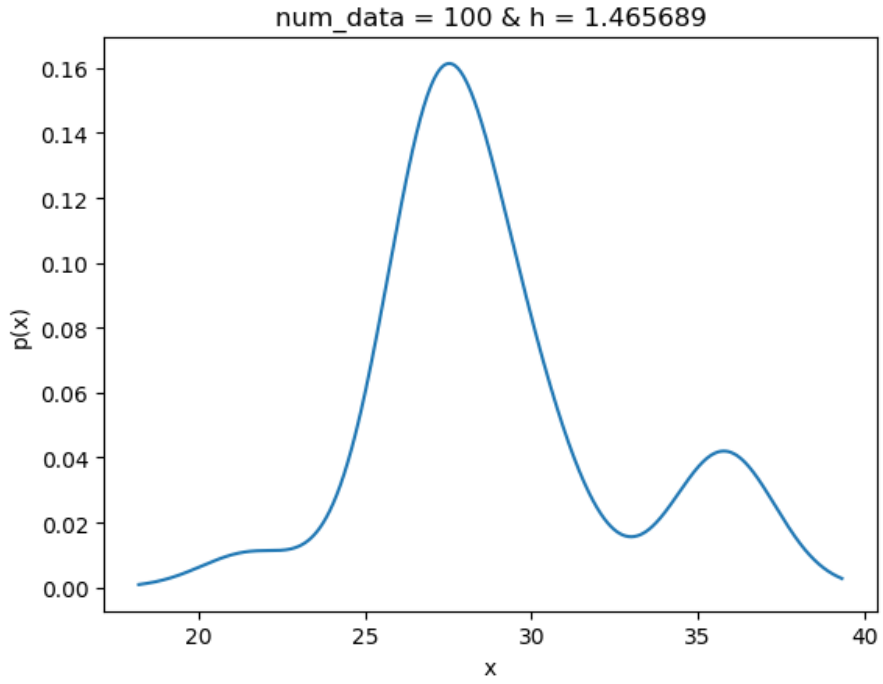
  当参数 h 取值过大时（例如 h=5.0），即带宽 h 选取过大，尽管曲线平滑，但会导致大量信息丢失，估计结果偏离正确的概率密度函数。

- **参数 h 取值的选择：**

  从上方图片看出，不同的带宽 h 得到的估计结果差别很大，因此选择可以使误差最小的 h，即可针对概率密度做出近似正确的估计。可以利用平均积分平方误差 MISE 的大小来衡量 h 的优劣： $\mathrm{MISE}(h) = E\int (\hat{f}_h(x) - f(x))^2 dx.$ 则可知，如果使用高斯核来进行概率密度估计，并且估计的基础密度为高斯分布，则 h 的最佳选择（即最小化平均积分平方误差的带宽）为：

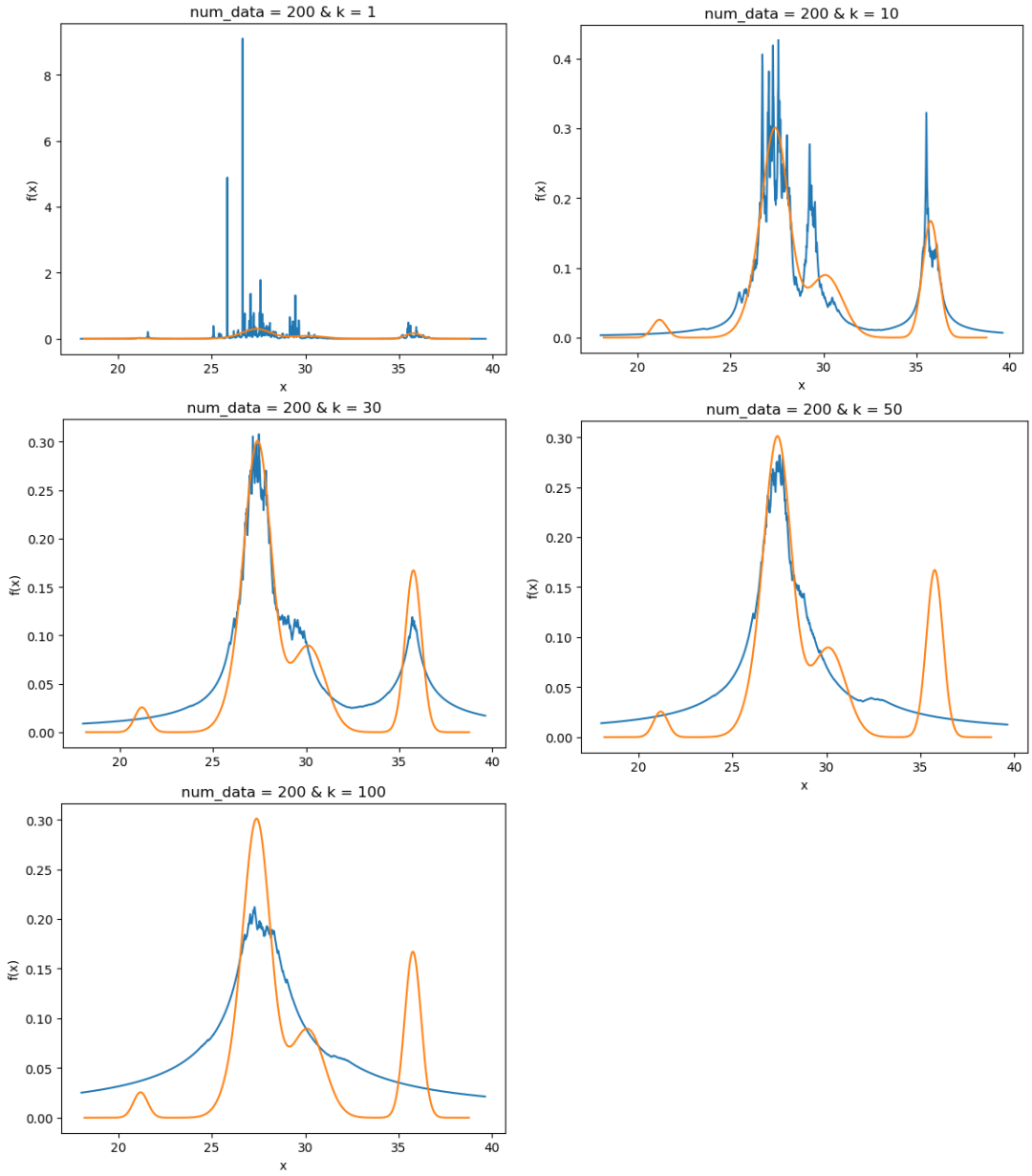
  $h = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}} \approx 1.06\hat{\sigma}n^{-1/5}$ （其中 $\hat{\sigma}$ 为样本数据的标准差）

  代入数据计算，当 n = 100 时，利用 np.std(sampled_data)计算得到标准差约为 3.47325，则 h=1.06*3.47325*$100^{-1/5}$=1.465689，即在该种选取方法下得到的合适的 h 值为 1.466，得到的概率密度估计曲线图如下：



Part 4: 最近邻方法

- 取 num_data = 200，分别取 k = 1、10、30、50，得到曲线如下：
  （加入实际分布曲线进行比较）



- 固定样本数据数量，根据变化 k 的取值，可以得到：
  当参数 k 取值过小时（例如 k=1），密度估计出现过拟合，从图中可以看到，得到的估计结果与真实概率密度分布差距很大；
  当参数 k 取值过大时（例如 k=50 & 100），密度估计出现欠拟合，从图中看出，估计曲线未能够体现第 1、3、4 个峰值，丢失了样本数据的大量特征，无法正确地估计概率密度。

- **Why nearest neighbor methods do not always converge to 1：**

  从最近邻方法的计算公式出发，p(x) = (K/N)/V，其中当样本数量和 K-近邻选定时，K/N 即为常数，而对于 V 的取值为当前 x 与 K-近邻点的距离的两倍，可以看作是一次函数 f(x)。因此，在整个 x 空间上对 p(x)求 x 的积分时，相当于对 constant/f(x)求 x 的积分，结果即为若干个 constant * ln(|f(x)|)的求和。显然，上述积分的值取决于样本点数量以及样本点的实际分布，在大多数情况下，该积分不会收敛于 1。

  以上解释了为什么最近邻方法概率密度估计的积分往往不收敛于 1。

附录：

编写代码过程中，利用 argparse 设置命令行参数，用以满足不同的画图需求：

直方图估计：**python source.py --he --num num_data --bins bins**

核密度估计：**python source.py --kde --num num_data --h h**

最近邻方法：**python source.py --nne --num num_data --k k**

（其中 num_data 为样本数据的数量，bins 为直方图估计所用参数，h 为核密度估计所用参数，而 k 为最近邻方法所用参数）