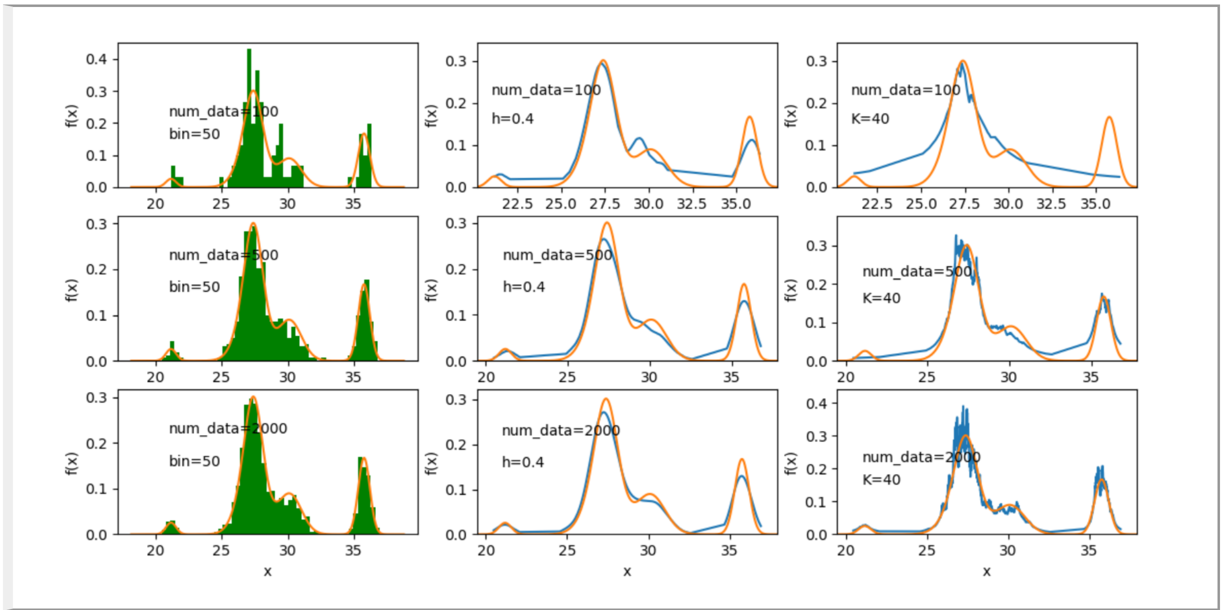
PRML Assignment-1 Report

RUN

```
from source import draw_histogram_estimation
import matplotlib.pyplot as plt
draw_histogram_estimation(200, 50)
plt.show()
```

Question 1

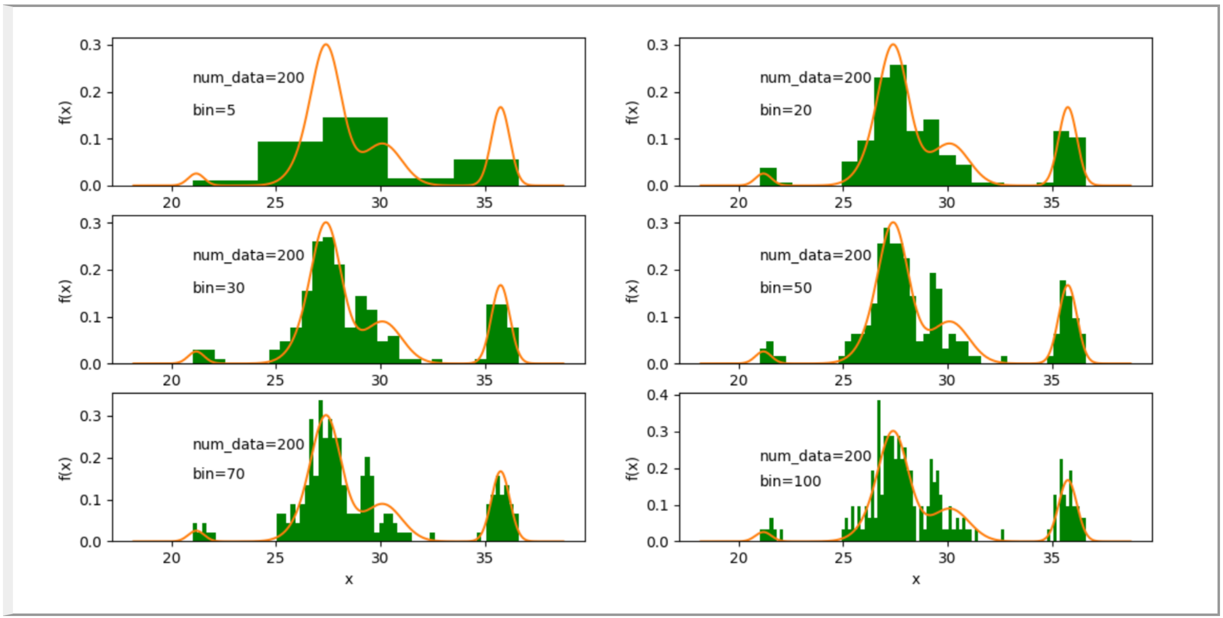
- The following picture is the comparison of three algorithms at different number of data used (constant parameter).The orange curve is the figure of true distribution.



- If I didn't print this picture, I may think the more number of data is used, the better result I can get, but it seems not totally right.
- For the **histogram estimation**, the guess is right. When `num_data = 2000` , the green histogram and the orange curve are **almost completely cioncident**; For the **kernel density estimation**, it seems **not much changes** when number of data used changes (maybe more **dependent the parameter h**); For the **nearest neighbor estimation**, althouth the two curves are more ciondicent, the blue one **looks very spiky** (maybe **related to parameter K**).

Qustion 2

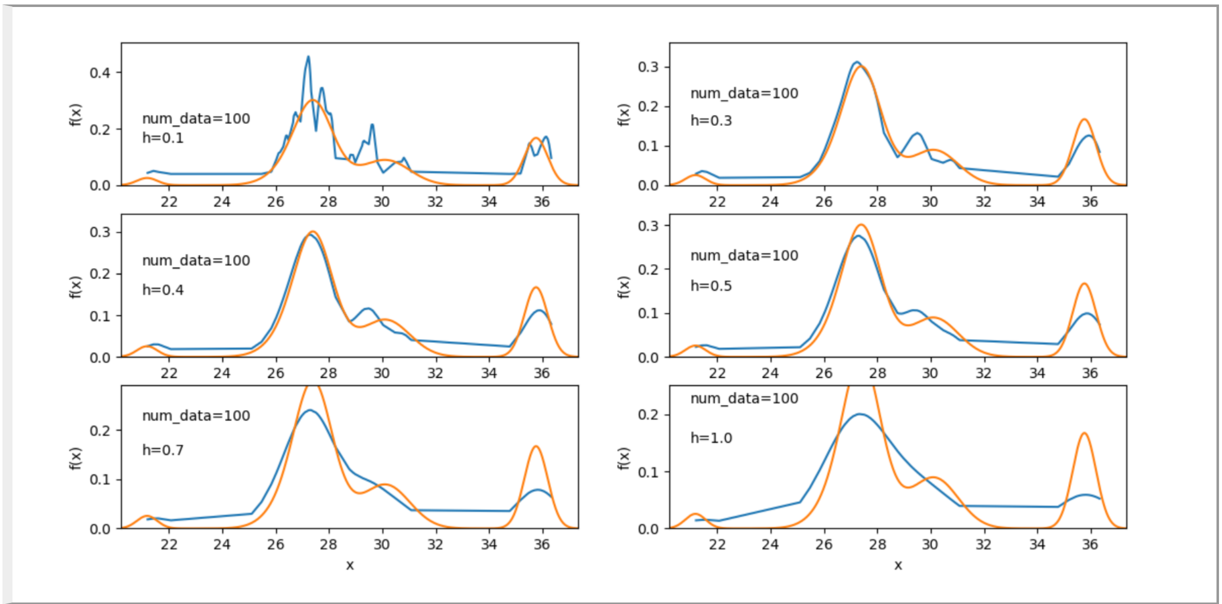
- The following picture is the comparison of histogram estimation at different parameter `bins` (constant number of data used).The orange curve is the figure of true distribution.



- From the picture above, we can see that when bins are **too large**, the resulting density model is **spiky**, with a lot of structure that is **not present** in the underlying distribution that generated the data set. If bins are **small** then the result is a model that **too smooth** and that consequently **fails to capture** the bimodal property of the orange curve.
- The best results are obtained for some **intermediate value** of bins. So I first chose a small one, about 10, found too small, then try 100, too large, then half of (10 + 100), then try the nearby numbers, last get **property `bins` is about 50**.
- According to the Wikipedia, depending on the actual data distribution, experimentation is needed to determine `bins` , but there also are some usefully rules, such as **Square-root choice**, **Sturges' formula**.

Question 3

- The following picture is the comparison of kernel density estimation at different parameter `h` (constant number of data used).The orange curve is the figure of true distribution.



- We can see that `h` acts as a **smoothing parameter** and that if it is set **too small**, the result is a very **noisy** density model, whereas if it is set **too large**, then the bimodal nature of the underlying distribution from which the data is generated is **washed out**.
- The best density model is obtained for some **intermediate value** of `h` . The method I chose `h` is like the above I chose `bins` . From 0.1 to 1, I used binary find and get the **property `h` is about 0.4**.

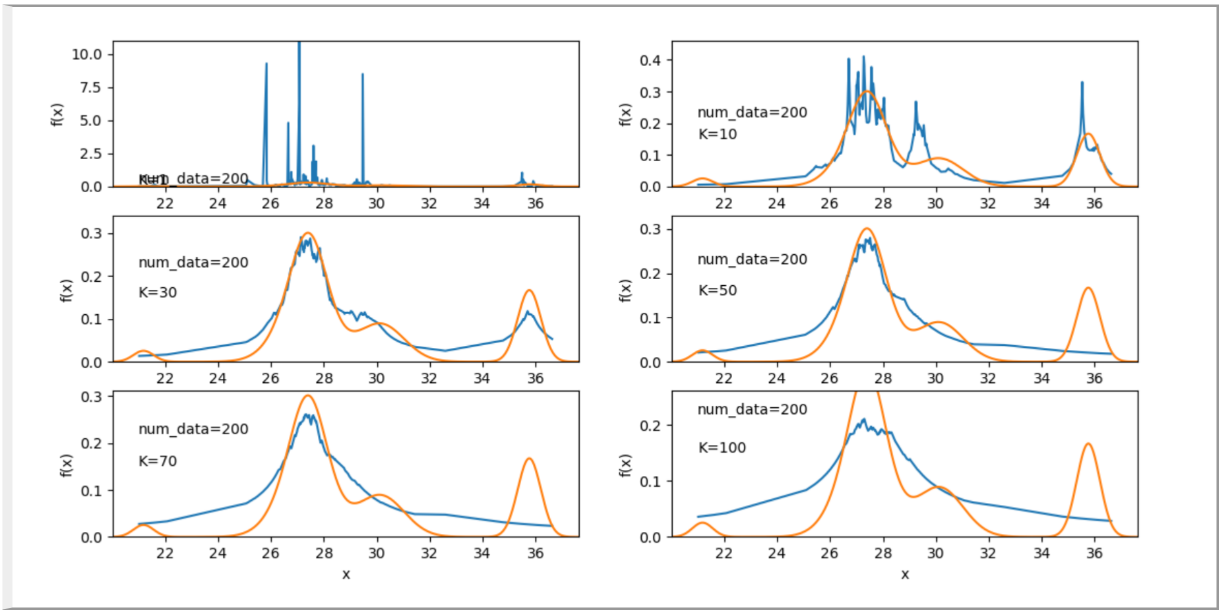
- According to the Wikipedia, there is a rule called Silverman's rule of thumb to estimate `h`.

$$h = (\frac{4\sigma^5}{3n})^{\frac{1}{5}}$$

  Get `h` is 1.465. However, at this value, the model didn't perform well.

Question 4

- The following picture is the comparison of nearest neighbor estimation at different parameter `h` (constant number of data used).The orange curve is the figure of true distribution.



- The model produced bu KNN is not a true density model because the **integral over all space diverges**.
- For the KNN, p(x) = K / (N * V) = (K / N) * 1/V, x takes any real number. Obviously, K/N is a constant c, but V **linearly depends on** the distance from x to its Kth nearest neighbor. Therefore the integration is

  $\int \frac{c}{kx + b}$, and it won't converge to 1.