STATS 205: Introduction to Nonparametric Statistics

Lecturer: Tengyu Ma — Lecture 1
Scribe: Honglin Yuan — Sept. 27th, 2019

1 Administration

1.1 Course Logistics

Please always refer to the course website `http://web.stanford.edu/class/stats205/` for complete versions of logistics and other information.

1.2 Topics Covered in this Course

The main philosophy of non-parametric statistics is to make as few assumptions as possible (not assuming linaer, polynomial, etc.).

This course will mainly focus on **non-parametric estimation**. We will cover the following topics:

- Nonparametric regression (today)
- Nonparametric density estimation
- Nearest neighbor algorithms
- Estimating CDFs
- Connection to over-parameterized neural networks (advanced topics, tentative)
- Wavelet (advanced topics, tentative)

Another subfield of non-parametric statistics is **non-parametric testing**, which is the non-parametric version of testing. This course will NOT cover non-parametric testing.

2 Non-parametric Regression

In today's lecture we will study non-parametric regression, which is covered in Chapter 5 of the textbook *All of Nonparametric Statistics* (`http://www.stat.cmu.edu/~larry/all-of-nonpar/`).

2.1 General Setup

Suppose we are given n pairs of observations, namely $(x_1, Y_1), (x_2, Y_2), \ldots, (x_n, Y_n)$. We assume both inputs and outputs are one-dimensional, i.e., $x_i \in \mathbb{R}$, $y_i \in \mathbb{R}$. [1].

Remark on Terminology In the textbook, the input variable x is referred to as **covariate** and the output variable (label) Y is referred to as **response variable**. In this lecture we will indistinguishably use these terminology as well as the input/output terminology.

[1] There are fundamental difficulties of dealing with high-dimension with non-parametric statistics

Assume that x_i is fixed (preselected) and Y_i is a random variable dependent on x_i (we use capital characters to emphasize the randomness). Let us write $Y_i = r(x_i) + \xi_i$ where $\mathbb{E}[\xi_i] = 0$ (in fact this can be done by putting $r(x_i) = \mathbb{E}[Y_i]$). The goal is to estimate (recover) $r(x_1), \dots, r(x_n)$. The loss of $\hat{r}$ is defined as follows

Definition 1 (Loss)**.** Given an estimator $\hat{r}$, the loss of $\hat{r}$ is defined as

$$\mathrm{loss}(\hat{r}) := \frac{1}{n} \sum_{i=1}^{n} \left(\hat{r}(x_i) - r(x_i) \right)^2 . \tag{1}$$

It is worth noting that the loss only involves the function value of $\hat{r}$ on the preselected points $\{x_i\}_{i=1}^n$.

An alternative viewpoint is to view $X_1, \dots, X_n$ as i.i.d. random variable with distribution $\mathcal{P}$, and then estimate the error as the expectation over $\mathcal{P}$. Particularly, if $\mathcal{P}$ is uniform over $\{x_1, \dots, x_n\}$, then the fixed design (1) is recovered. Hence (1) is a strictly simpler case.

2.2 Relation to Parametric Regression

- In linear regression, we assume $r(x) = ax + b$ and fit parameters a and b to minimize loss.
- In certain case where the underlying function has a large curvature, using linear regression may not be a good choice. An alternative choice is to use **polynomial regression** — assuming $r(x)$ is a k-degree polynomial, namely $r(x) = a_k x^k + a_{k-1} x^{k-1} + \cdots + a_0$. (for some fixed k).
- However, there are functions that **no** low-degree polynomial can fit. One manifestiation is that if a non-zero degree-k polynomial $f(x)$ satisfies $f(z_1) = \cdots = f(z_{k+1}) = C$ for $k+1$ distinct points $z_1, \dots, z_{k+1}$ and some constant C, then $f \equiv C$, since no non-zero degree-k polynomial has more than $k+1$ roots. Hence we need non-parametric solution for regressions.

In today's lecture, we will cover three methods of non-parametric regression: **Regressogram**, **Local Averaging**, and **Kernel Estimator**. We will also discuss the bias-varaince tradeoff of these algorithms.

2.3 Regressogram

The motivation is to divide samples into equally spaced bins $\{B_j\}$, and use constant functions to fit each bin. Formally, we set $\hat{r}$ as follows:

$$\forall x \in B_j, \quad \hat{r}(x) := \frac{1}{\#\{i : x_i \in B_j\}} \sum_{x_i \in B_j} Y_i. \tag{2}$$

Note that we only care about the bin with positive number of x_i's since $\hat{r}$ is only evaluated at these points, and therefore the assignment of $\hat{r}$ at empty bin does not contribute to the loss.

However, equal binning at stated above may not be a good option when the samples are not equally distributed. We would like to adaptively select the bin based on sample points. This motivates the following Local Averaging Algorithm.

2.4 Local Averaging Algorithm

The motivation is to use specialized bins for each data points. Formally for arbitrary x, $\hat{r}(x)$ is set as follows

$$B_x := \{j : |x_j - x| \leq h\}, \quad n_x := |B_x|, \quad \hat{r}(x) := \frac{1}{n_x} \sum_{j \in B_x} Y_j, \tag{3}$$

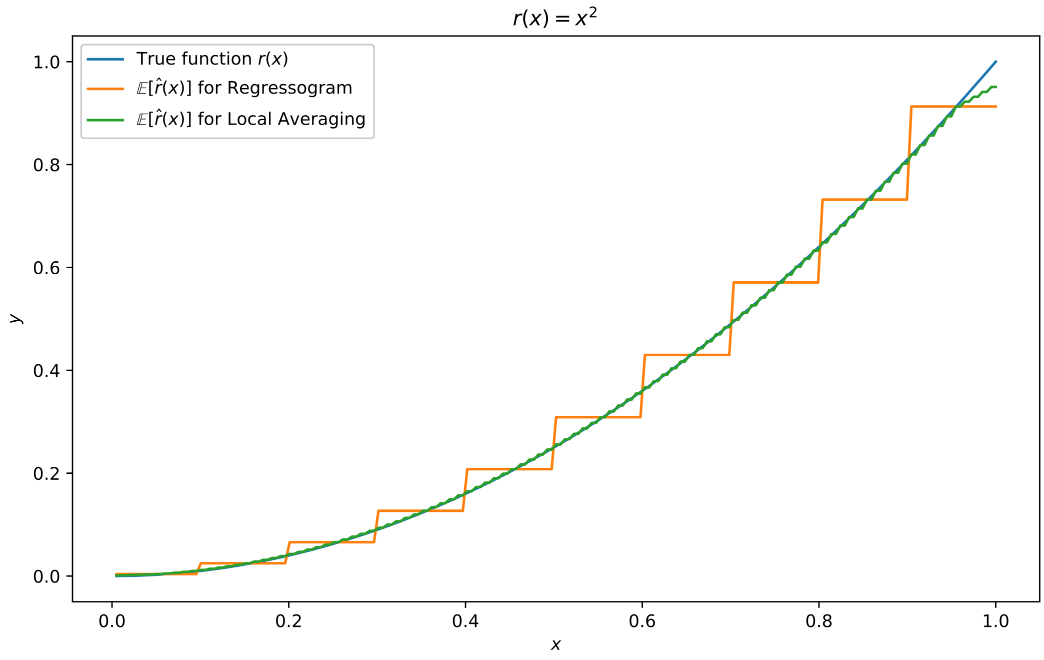
where h is commonly referred to as the **bandwidth**.

In local averaging algorithm, $\hat{r}(x)$ in fact solves the following optimization problem:

$$\hat{r}(x) = \arg\min_a \left(\sum_{j \in B_x} (Y_j - a)^2 \right) = \frac{1}{|B_x|} \sum_{j \in B_x} Y_j. \tag{4}$$

To better understand the potential benefits of adaptive binning introduced by local averaging, let us consider the following example. Suppose the ground truth $r(x) = x^2$, and 100 sample points are equally spaced between $[0, 1]$. We plot the $\mathbb{E}[\hat{r}(x)]$ for both Regressogram (with 10 bins) and Local Averaging (with $h = 0.05$), see Figure 1. We can observe that the $\mathbb{E}[\hat{r}(x)]$ for Local Averaging stays closely with the true function $r(x)$ (except at the boundary), which suggests a smaller bias. On the other hand, the same quantity for Regressogram is stepwise, which has a larger bias.

Figure 1: $r(x) = x^2$



3 Kernel Estimator

In local averaging, B_x introduces a hard truncation of summation, which implies that all points outside the neighborhood will be ignored for the estimation at x. A smoother alternative is to use kernel regression as follows. The motivation of Kernel Estimator is to use full sum with potentially different weights, instead of partial sum. Formally we set

$$\hat{r}(x) \leftarrow \arg\min_a \left(\sum_{j=1}^n w_j (Y_j - a)^2 \right) = \frac{\sum_{j=1}^n w_j Y_j}{\sum_{j=1}^n w_j}, \tag{5}$$

where w_j is a list of preselected weight at x. The principle is to set $w_j > w_{j'}$ if x_j is closer to x than x_j.

3.1 Nadayara-Watson Kernel Estimator

The common practice of setting weights w_j is to let $w_j = K\left(\frac{x-x_j}{h}\right)$, where K is a kernal defined as follows, and h is the bandwidth. This is the **Nadayara-Watson kernel estimator**

$$\hat{r}(x) = \sum_{i=1}^n l_i(x) Y_i, \quad \text{where } l_i(x) = \frac{K(\frac{x-x_i}{h})}{\sum_{j=1}^n K(\frac{x-x_j}{h})}. \tag{6}$$

Note that the larger the bandwidth h is, the fatter the $K(x/h)$ will be.

Definition 2 (Kernel). $K : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a kernel function if

1. $\int_{\mathbb{R}} K(x) \mathrm{d}x = 1$
2. $\int_{\mathbb{R}} x K(x) \mathrm{d}x = 0$ (weak symmetry) or $K(x) = K(-x)$ (stronger symmetry).
3. $\sigma_K^2 := \int_{\mathbb{R}} x^2 K(x) \mathrm{d}x > 0$. ($K(x)$ is not a point-mass at 0)

We list a few examples of kernel which are commonly applied.

Boxcar kernel $K(x) = \frac{1}{2}\mathbf{1}(|x| \leq 1)$. Remark: local averaging is equivalent to using boxcar kernel with bandwidth h.

Gaussian kernel $K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$ (p.d.f. of $N(0,1)$)

Epanechnikov Kernel $K(x) = \frac{3}{4}\left(1 - x^2\right) \mathbf{1}\left(|x| \leq 1\right)$

Triaube kernel $K(x) = \frac{70}{81}\left(1 - |x|^3\right) \mathbf{1}\left(|x| \leq 1\right)$

3.2 The Intuition of Selecting Bandwidth h

In this subsection we will provide intuitions on the selection of h. The goal is to show that **the best bandwidth depends on the data**. We will focus on local averaging as a motivating example.

Recall we assume that $Y_i = r(x_i) + \xi_i$ where $\mathbb{E}[\xi_i] = 0$.

Case 1 Suppose the true function $r(x) \equiv C$ and for simplicity we assume $\xi_i \sim N(0, \sigma^2)$.

- If $h = +\infty$, then $\hat{r}(x_i) = \frac{1}{n}\sum_{j=1}^{n} Y_j = \frac{1}{n}\sum_{j=1}^{n}(r(x_j)+\xi_j) = c + \frac{1}{n}\sum_{j=1}^{n}\xi_j$. Hence $\hat{r}(x_i) \sim N(c, \sigma^2/n)$.
- If $h = 0$, then $\hat{r}(x_i) = Y_i \sim N(c, \sigma^2)$. Effectively we are not doing any estimation.
- In general, for $h \in (0, +\infty)$, the estimation $\hat{r}(x_i) = C + \frac{1}{n_{x_i}}\sum_{j\in B_{x_i}}\xi_j \sim N(C, \sigma^2/n_{x_i})$, hence the larger the bandwidth h is, the larger the $|n_{x_i}|$ is, and the smaller the variance will be.

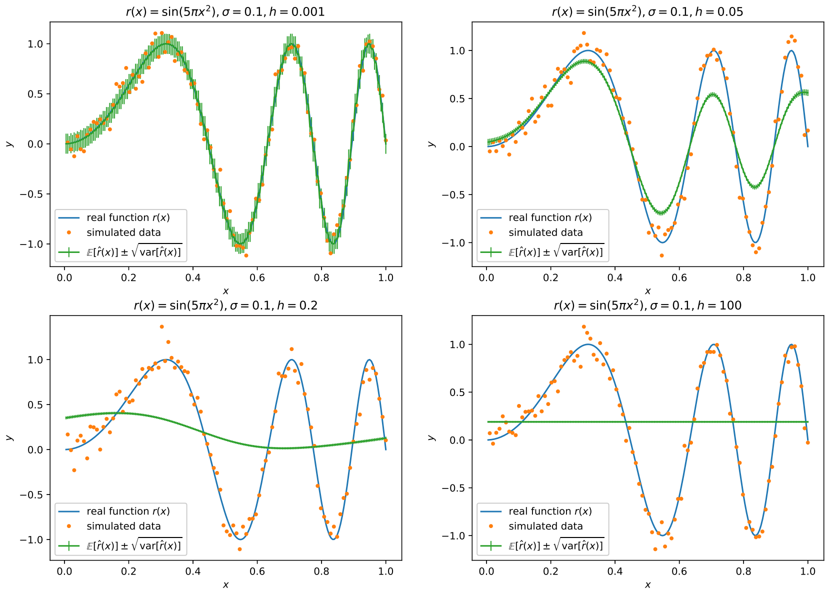
Case 2 The true function $r(x)$ fluctuates a lot, but the noise $\xi_i \equiv 0$. In this case we prefer smaller h. (in fact $h = 0$ gives the estimation with *zero* loss since no denoising is needed)

Case 3 In general, with moderate fluctuation of $r(x)$ and moderate noise, we might prefer some bandwidth in the middle to trade off the local-estimation and denoising.

We provide some concrete intuitions on selecting the bandwidth h.

- The first example is constructed by setting a rough signal $r(x) = \sin(5\pi x^2)$, with relatively small noise $\xi_i \sim N(0, 0.1^2)$. The x_i are 99 equally spaced points between $[0, 1]$. We plot the kernel estimator for Gaussian kernel with $h = 0.001, 0.05, 0.2, 100$, see Figure 2. The $h = 0.001$ case has the best bias-variance tradeoff. The intuition is that with such small noise the need for de-noising is very weak, so smaller h is needed to tradeoff the variance.

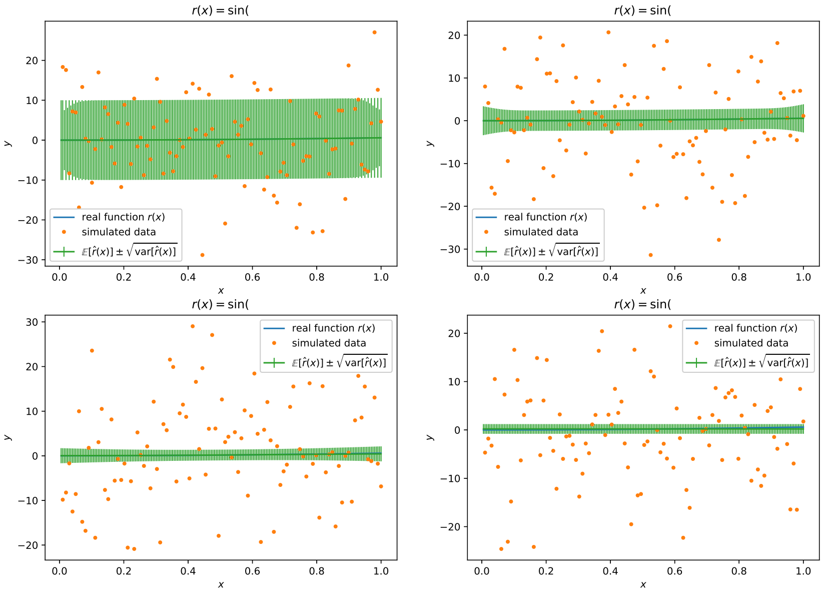
Figure 2: $r(x) = \sin(5\pi x^2), \sigma = 0.1$



- The second example is constructed by setting a flat signal $r(x) = \sin(\frac{1}{5}\pi x^2)$, with relatively large noise $\xi_i \sim N(0, 10^2)$. The x_i are 99 equally spaced points between $[0, 1]$. We plot the kernel estimator for Gaussian kernel with $h = 0.001, 0.05, 0.2, 100$, see Figure 3. The $h = 100$

case has the best bias-variance tradeoff. The intuition is that since r is flat the bias is relative small, so larger h will be preferred for better denoising.

Figure 3: $r(x) = \sin(\frac{1}{5}\pi x^2), \sigma = 10$



3.3 Bias-Variance Tradeoff

In this subsection we will mathematically formalize the above intuition in Subsection 3.2.

Definition 3 (Predictive risk)**.** For $i = 1, \ldots, n$, let Z_i be a new fresh observation at x_i, namely $Z_i = r(x_i) + \xi_i'$. The predictive risk of $\hat{r}$ is defined by

$$\mathrm{risk}(\hat{r}) := \mathbb{E}_{Z_1,\ldots,Z_n}\left[\frac{1}{n}\sum_{i=1}^{n}(Z_i - \hat{r}(x_i))^2\right] \tag{7}$$

We claim that the predictive risk differs with the loss with only a constant.

$$\mathrm{risk}(\hat{r}) = \mathbb{E}_{Z_1,\ldots,Z_n}\left[\frac{1}{n}\sum_{i=1}^{n}(Z_i - \hat{r}(x_i))^2\right] = \frac{1}{n}\sum_{i=1}^{n}\left[(r(x_i) - \hat{r}(x_i))^2 + \mathbf{Var}(\xi_i')\right] = \mathrm{loss}(\hat{r}) + \frac{1}{n}\sum_{i=1}^{n}\mathbf{Var}(\xi_i') \tag{8}$$

Hence it suffices to focus on the loss.

Note that $\mathrm{loss}(\hat{r})$ is still a random variable since $\{Y_i\}_{i=1}^n$ is random. Taking expectation over $\{Y_i\}$,

$$\mathbb{E}_{Y_1,\ldots,Y_n}[\mathrm{loss}(\hat{r})] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y_i}\left[(\hat{r}(x_i) - r(x_i))^2\right] = \frac{1}{n}\sum_{i=1}^{n}\left[(\mathbb{E}\,\hat{r}(x_i) - r(x_i))^2 + \mathbf{Var}\left(\hat{r}(x_i)\right)\right]. \tag{9}$$

The first term $(\mathbb{E}\,\hat{r}(x_i) - r(x_i))^2$ is referred to as **bias**, and the second term $\mathbf{Var}\,(\hat{r}(x_i))$ is referred to as **Variance**. The equation (9) is commonly called the **Bias-Variance Decomposition Formula**.

Let us illustrate the bias-varaince decomposition with a specific example. For kernel regression with weights $w_j = K(\frac{x_i - x_j}{h})$, the estimation $\hat{r}$ at x_i is given by

$$\hat{r}(x_i) = \frac{\sum_{j=1}^n w_j Y_j}{\sum_{j=1}^n w_j} = \left(\frac{1}{\sum_{j=1}^n w_j}\right) \sum_{j=1}^n w_j \left(r(x_j) + \xi_j\right). \tag{10}$$

The mean is given by

$$\mathbb{E}_{Y_1,\ldots,Y_n}\left[\hat{r}(x_i)\right] = \frac{1}{\sum_{j=1}^n w_j} \sum_{j=1}^n w_j r(x_j) \tag{11}$$

Thus the bias term measures the capability of the estimator on the clean data. Using stronger smoothing will increase bias as it behaves worse on the clean data.

Assuming $\mathbf{Var}(\xi_i) \equiv \sigma^2$, the variance term is given by

$$\mathbf{Var}_{Y_1,\ldots,Y_n}[\hat{r}(x_i)] = \mathbf{Var}_{Y_1,\ldots,Y_n}\left[\left(\frac{1}{\sum_{j=1}^n w_j}\right)\left(\sum_{j=1}^n w_j \xi_j\right)\right] = \frac{\sum_{j=1}^n w_j^2}{\left(\sum_{j=1}^n w_j\right)^2}\sigma^2 \tag{12}$$

Particularly for Boxcar kernel, the last term becomes

$$\frac{\sum_{j=1}^n w_j^2}{\left(\sum_{j=1}^n w_j\right)^2}\sigma^2 = \frac{1}{n_{x_i}}\sigma^2. \tag{13}$$

Hence the larger the bandwidth h is, the smaller the variance is.