

# STATS 205: Introduction to Nonparametric Statistics

Lecturer: Tengyu Ma  
Scribe: Honglin Yuan

Lecture 1 Part 1  
Sept. 27th, 2019

---

## 1 Administration

### 1.1 Course Logistics

Please read through the course website <http://web.stanford.edu/class/stats205/> for complete versions. Here are some highlights.

- We will use Piazza for Q&A.
- Prerequisites: STATS 200, 203, 217, Linear Algebra, Calculus, either R or Python Programming.
- Homeworks: There are four homeworks. The assignments can be done individually or in pairs.
- Projects: The projects can be done in pairs. For example you can implement algorithms taught on class and apply on new dataset.
- Scribe notes: This course will cover more on non-parametric estimation, instead of non-parametric test, which were offered in previous years. Scribing notes will be helpful for students who miss the class. Please sign up before Sept 30th.
- Late policy: Please refer to the course website.

### 1.2 Topics Covered in this Course

The main philosophy of non-parametric statistics is to make as few assumptions as possible (not assuming linear, 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  $(x_1, Y_1), (x_2, Y_2), \dots, (x_n, Y_n)$ . We assume both inputs and outputs are one-dimensional, i.e.,  $x_i \in \mathbb{R}, y_i \in \mathbb{R}$ .<sup>1</sup>

**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.

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

$$\text{loss}(\hat{r}) := \frac{1}{n} \sum_{i=1}^n (\hat{r}(x_i) - r(x_i))^2. \quad (1)$$

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  $r(x) = a_k x^k + a_{k-1} x^{k-1} + \dots + a_0$ . (for some fixed  $k$ ).
- However, there are functions that **no** low-degree polynomial can fit. One manifestation is that if a non-zero degree- $k$  polynomial  $f(x)$  satisfies  $f(z_1) = \dots = f(z_{k+1}) = C$  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 Regression**. We will also discuss the bias-variance tradeoff of these algorithms.

---

<sup>1</sup>There are fundamental difficulties of dealing with high-dimension with non-parametric statistics

## 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:

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

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

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

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

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

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

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.

## 3 Kernel Regression

The motivation of Kernel Regression is to use full sum with potentially different weights, instead of partial sum. Formally we set

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

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  $x_{j'}$ . The common practice is to set  $w_j = K\left(\frac{x - x_j}{h}\right)$ , where  $K$  is a kernel defined as follows, and  $h$  is the bandwidth. Note that the larger the bandwidth  $h$  is, the fatter the  $K(x/h)$  will be.

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

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

Note that the definition is actually very weak – it does not even require  $K$  to be concentrated 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**  $K(x) = \frac{3}{4} (1 - x^2) \mathbf{1}(|x| \leq 1)$

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

### 3.1 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.

### 3.2 Bias-Variance Tradeoff

In this subsection we will mathematically formalize the above intuition in Subsection 3.1.

**Definition 3** (Predictive risk). For  $i = 1, \dots, n$ , 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

$$\text{risk}(\hat{r}) := \mathbb{E}_{Z_1, \dots, Z_n} \left[ \frac{1}{n} \sum_{i=1}^n (Z_i - \hat{r}(x_i))^2 \right] \quad (6)$$

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

$$\text{risk}(\hat{r}) = \mathbb{E}_{Z_1, \dots, Z_n} \left[ \frac{1}{n} \sum_{i=1}^n (Z_i - \hat{r}(x_i))^2 \right] = \frac{1}{n} \sum_{i=1}^n \left[ (r(x_i) - \hat{r}(x_i))^2 + \mathbf{Var}(\xi'_i) \right] = \text{loss}(\hat{r}) + \frac{1}{n} \sum_{i=1}^n \mathbf{Var}(\xi'_i) \quad (7)$$

Hence it suffices to focus on the loss.

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

$$\mathbb{E}_{Y_1, \dots, Y_n} [\text{loss}(\hat{r})] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{Y_i} \left[ (\hat{r}(x_i) - r(x_i))^2 \right] = \frac{1}{n} \sum_{i=1}^n \left[ (\mathbb{E} \hat{r}(x_i) - r(x_i))^2 + \mathbf{Var}(\hat{r}(x_i)) \right]. \quad (8)$$

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 (8) is commonly called the **Bias-Variance Decomposition Formula**.

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

$$\hat{r}(x_i) = \frac{\sum_{j=1}^n w_j Y_j}{\sum_{j=1}^n w_j} = \left( \frac{1}{\sum_{j=1}^n w_j} \right) \sum_{j=1}^n w_j (r(x_j) + \xi_j). \quad (9)$$

The mean is given by

$$\mathbb{E}_{Y_1, \dots, Y_n} [\hat{r}(x_i)] = \frac{1}{\sum_{j=1}^n w_j} \sum_{j=1}^n w_j r(x_j) \quad (10)$$

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

$$\mathbf{Var}_{Y_1, \dots, Y_n} [\hat{r}(x_i)] = \mathbf{Var}_{Y_1, \dots, Y_n} \left[ \left( \frac{1}{\sum_{j=1}^n w_j} \right) \left( \sum_{j=1}^n w_j \xi_j \right) \right] = \frac{\sum_{j=1}^n w_j^2}{\left( \sum_{j=1}^n w_j \right)^2} \sigma^2 \quad (11)$$

Particularly for Boxcar kernel, the last term becomes

$$\frac{\sum_{j=1}^n w_j^2}{\left( \sum_{j=1}^n w_j \right)^2} \sigma^2 = \frac{1}{n_{x_i}} \sigma^2. \quad (12)$$

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