**STATS205: Introduction to Nonparametric Statistics**

Lecturer: Tengyu Ma — Lecture # 1
Scribe: Lan Jiang and Sameer Sundrani — April 2nd, 2021

## 1.1 Overview

In this lecture, we begin our exploration of nonparametric statistics. We first describe the underlying motivation for the field of nonparametric statistics and its general principles. Then, we look at our first examples of such statistics with the nonparametric regression problem, wherein we focus on three different approaches: the regressogram, local averaging, and Nadaraya-Watson kernel estimator.

## 1.2 Overview of nonparametric statistics

The overarching idea of nonparametric statistics is that it does not leverage standard parameterization. While there are not many precise definitions for the field of nonparametric statistics, there are a few core tenets to know.

- Make as few assumptions as possible. For example, do not assume that data extends from linear or quadratic model.
- No fixed set of parameters exists. For example, in nonparametric statistics we will often see across infinite dimensional models, infinite parameters, or circumstances where the dimension $\to \infty$ as the number of data point $n \to \infty$.

Such principles are widely applicable to many areas of statistics and machine learning, such as in nonparametric testing, supervised learning, and unsupervised learning.

However, often and particularly in this class, our data is low dimensional[1], with exceptions like neural networks and some kernel methods. This is important since high dimensional data without many strong parametric assumptions will fundamentally and statistically need many samples (i.e. the data will need exponential dimensions) to estimate anything (density, CDF, etc.), suffering from the “curse of dimensionality”. The lack of high dimensional data without many strong parametric assumptions results in estimate errors at the zero-th or first order.

## 1.3 The nonparametric regression problem

### 1.3.1 Setup

Our first example in nonparametric statistics will be nonparametric regression. In such a problem, we have $n$ pairs of observations

$$(x_1, Y_1), ..., (x_n, Y_n), \tag{1.1}$$

[1] In this course, low dimensions generally refers to the case when data dimension $d = 1, 2, 3$

where each $x_i, Y_i \in \mathbb{R}$ and $x_i$ refers to an input (or covariate) and $Y_i$ refers to an output (or label) or response variable. Furthermore, each $Y_i$ can be written as

$$Y_i = r(x_i) + \xi_i, \tag{1.2}$$

where $r(.)$ is some function and $\xi$ is some noise on the observed output. Here, $r(x_i)$ is defined by

$$r(x_i) := \mathbb{E}[Y_i|x_i] \tag{1.3}$$

and $\xi_i = Y_i - r(x_i)$, with $\mathbb{E}[\xi_i] = 0$.

With these observations now defined, we can view the regression problem under two frameworks: deterministic inputs or random inputs, and we will explore both possibilities in the subsequent sections.

### 1.3.2 Deterministic design and mean squared error

In this view, we treat $x_1, ..., x_n$ as fixed, deterministic inputs with $Y_1, ...Y_n$ being random variables. Our goal then is to estimate or recover $r(x_1), ..., r(x_n)$ as accurately as possible. Pictorially, we can see this in Figure 1.1, where the red circular open dots are the "noisy" set of observations and the blue $r(x)$ is the function where each labeled $r(x_i)$ (in black circular open dots) is what we aim to recover.

Our estimator will therefore be denoted as $\hat{r} : \hat{r}(x_1), ..., \hat{r}(x_n)$. We will evaluate $\hat{r}$ utilizing mean squared error or MSE as

$$\text{MSE}(\hat{r}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{r}(x_i) - r(x_i))^2. \tag{1.4}$$

Because there is randomness from each $Y_i$, we can rewrite this as an expectation utilizing the same form we just described as

$$\text{MSE} = \mathbb{E}[\text{MSE}(\hat{r})] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\hat{r}(x_i) - r(x_i))^2\right]. \tag{1.5}$$

### 1.3.3 The alternative view point: random design

Alternatively, we could have viewed our input as a series of independent and identically distributed random variables $X_1, ..., X_n \overset{\text{iid}}{\sim} P$ (note the upper-case $X_i$ now) where our $Y_i = r(X_i) + \xi_i$. Our interpretation of such an perspective while modeling the problem remains very similar, though, and our estimator is still some $\hat{r}$ evaluated using MSE as $\mathbb{E}_{X\sim P}[(\hat{r}(X) - r(X))^2]$. For the rest of this note, though, we will maintain within the deterministic design paradigm described above.

### 1.3.4 Motivation for nonparametric regression

With our current understanding of the regression problem at hand, one may claim we can solve such examples parametrically by assuming that each $Y_i$ is a linear combination of the input (as in linear regression) or some polynomial combination of the input (as in polynomial regression). However, consider the regression where $r(x)$ is neither linear nor polynomial as in Figure 1.2 where

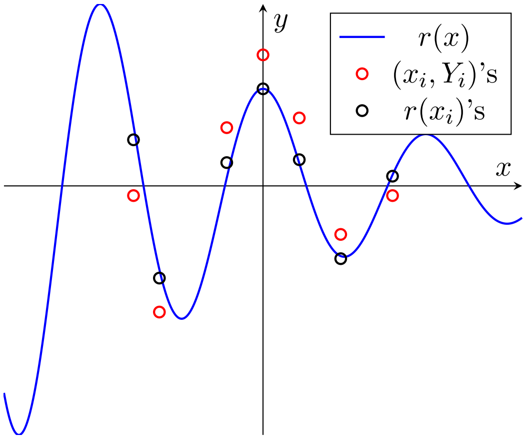

Figure 1.1: Graphical representation of regression problem

$r(x)$ cannot be fitted with any polynomial fit perfectly (assume here that after some $x_0$, $r(x \geq x_0)$ remains fixed at some constant value).

To see why polynomial regression would fail, suppose we fit $r(x)$ with $f(x)$ a $k$-degree polynomial. Suppose $x_1, \cdots, x_{k+1} \geq x_0$ and then $f(x_1) = ... = f(x_{k+1}) = c$ for distinct values of $x_1, ..., x_{k+1}$. Since a $k$-degree polynomial is uniquely determined to its values at $k+1$ different points, the only possible solution would be $f(x) = c$, the constant function. Therefore it is not possible to fit all the points on both the right and left hand side of this curve above. We now see that we cannot simply make an assumption of the behavior of our $r(x)$ in this case, and must turn to new ways of achieving our goal. We turn now to some methodologies of nonparametric regression.

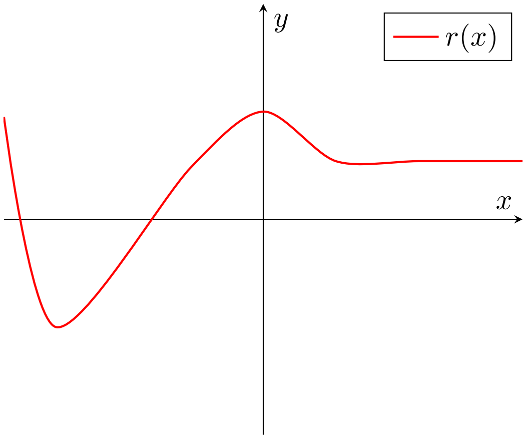

Figure 1.2: nonparametric regression problem on which polynomial regression fail.

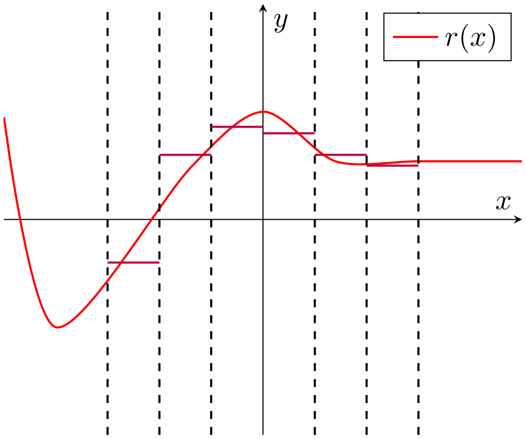

Figure 1.3: Regressogram binning with piecewise constant functions

## 1.4 nonparametric regression methods

### 1.4.1 Regressogram

Our first methodology to approach our modeling problem is known as the regressogram approach. Our algorithm is quite rudimentary: divide our domain of $x$ into some number of bins (assume that our bins are of equal size) as shown in Figure 1.3. For each $(x_i, Y_i)$ that falls within a given bin $B_j$, our estimator is defined as

$$\hat{r} = \frac{1}{|B_j|} \sum_{i \in B_j} Y_i, \tag{1.6}$$

which is, in other words, the average of all $Y_i$ where $x_i \in B_j$. Because each point will fall into some $B_j$ (and for all bins where there is no observation, we won't have a defined prediction), we recover a piece-wise constant function for that $B_j$. We then see that each point within a particular $B_j$ will recover the same $\hat{r}$. While this approach is simple, it is not often used in practice as choosing the binning method and size is quite tricky. Additionally, some bins may not capture many observations while others may not capture enough of a set of observations if there are too many variable regions within a bin.

### 1.4.2 Local averaging

As the original regressogram can easily fail to capture or over-capture a set of observations, we move to a modified version of binning known as local averaging. Here, we instead define bins dynamically i.e. for each observation $(x_i, Y_i)$, we define a bin of some size $h$ in each direction around that value. Each bin is therefore defined in terms of a particular observation such that $B_{x_i} = \{j : |x_j - x_i| \leq h\}$ where our estimator is now

$$\hat{r}(x_i) = \frac{1}{|B_{x_i}|} \sum_{i \in B_{x_i}} Y_i, \tag{1.7}$$

which is, similar to before, an average, but now with the nuance that each bin is defined with respect to each observation. One such bin with locally averaged $\hat{r}(x_i)$ can be seen in Figure 1.4. Note that we are making the assumption that within each $B_{x_i}$, $\hat{r}(x_i)$ is a constant denoted by $a$.

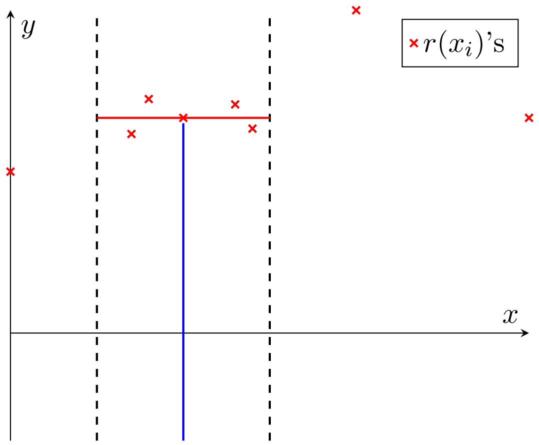

Figure 1.4: Local averaging with one $x_i$, estimating $\hat{r}(x_i)$ as a red line

Following this assumption, we can then derive $\hat{r}(x_i)$ as minimizing the MSE inside each local bin, namely

$$\hat{r}(x_i) = \operatorname*{argmin}_{a} \frac{1}{|B_{x_i}|} \sum_{i \in B_{x_i}} (Y_i - a)^2 = \frac{1}{|B_{x_i}|} \sum_{i \in B_{x_i}} Y_i. \tag{1.8}$$

Setting the first derivative of this function to zero and solving for $a$, we see that the minimum $a$ would be the average value within a bin, as we estimate within this method.

As with the regressogram, we make some assumptions in local averaging that may not be sufficient for our regression. Namely, we are assuming that we can safely ignore all points outside the boundary of each $B_{x_i}$, even if such points are borderline to the bin determined by some $h$. Such a problem motivates our next methodology: soft-weight averaging.

### 1.4.3 Nadaraya-Watson kernel estimator (soft-weight averaging)

As we saw in the previous section on local averaging, we make a potentially detrimental assumption that all points outside the boundary of a given bin should not be considered for our estimator $\hat{r}(x)$. To alleviate this problem, we introduce the concept of soft-weight averaging, where we introduce a weighting for each observation that can be distance dependent. More specifically, we define our new constant estimate $a$ for each observation $(x_i, Y_i)$ over all $n$ observations as follows

$$\operatorname*{argmin}_{a \in \mathbb{R}} \sum_{i=1}^{n} w_i (Y_i - a)^2 = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i}, \tag{1.9}$$

where the minimizer can be found by taking derivative w.r.t. $a$. Notice that if $w_i = \mathbf{1}\{|x_i - x| \leq h\}$, we recover the same local averaging we have previously described. In general, we desire $w_i$ to be smaller as $|x_i - x|$ increases and for any $|x_i - x| > |x_j - x|$ it follows that $w_i < w_j$.

We now need to define a weighting scheme that satisfies our weighting desires, and we will do so using a kernel estimator. Namely, because we desire to have a weighting dependent on the distance a particular observation $x_i$ is from $x$, we will define

$$w_i = f(x_i - x) = K\left(\frac{x_i - x}{h}\right), \tag{1.10}$$

where $K(\cdot)$ is a kernel function and $h$ is the width of our bins.

### 1.4.4 Kernel functions

Before we describe further our possible choices of kernel $K$, we must define more specifically what properties $K$ follows. Formally, we make the following definition.

**Definition 1.1** (Kernel function)**.** $K : \mathbb{R} \to \mathbb{R}$ is called a kernel function if it is non-negative and satisfies the following properties.

1. $\int_{\mathbb{R}} K(t)dt = 1$. $K$ is normalized and scales to 1.
2. $\int_{\mathbb{R}} tK(t)dt = 0$. There must be some kind of symmetry within $K$.
3. $\sigma_t^2 := \int_{\mathbb{R}} t^2 K(t)dt > 0$.

Now, we will discuss four variants of kernels. We start by the boxcar kernel

$$K(t) = \frac{1}{2}\mathbf{1}\left\{|t| \leq 1\right\}. \tag{1.11}$$

The boxcar is named for its box-like shape and can be seen in red in Figure 1.5 overlayed with the Gaussian kernel as well. One then sees that this corresponds exactly to local averaging for some $h$. Next we have the Gaussian kernel

$$K(t) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-t^2}{2}\right), \tag{1.12}$$

Unlike the boxcar, here we have some non-zero weight for some $x_i$ when $|x_i - x| > h$, which tapers off as that difference increases. Other choices of kernels include the Epanechnikov kernel

$$K(t) = \frac{3}{4}(1 - t^2)\mathbf{1}\left\{|t| \leq 1\right\} \tag{1.13}$$

and the tricube kernel

$$K(t) = \frac{70}{81}(1 - |t|^3)\mathbf{1}\left\{|t| \leq 1\right\}. \tag{1.14}$$

In practice, however, the explicit choice of kernel is not that important empirically between the these (excluding boxcar).

### 1.4.5 Choosing the bandwith

Finally, we approach the discussion of choosing our bin width $h$. Larger or smaller values of $h$ can dramatically change our estimates $\hat{r}(x_i)$, so understanding how to do so is critical for our regression. To see why this is true, consider any choice of kernel. Here we see that if we increase $h$, we are simply increasing the bin widths (imagining a larger width in, say, Figure 1.4) and thereby averaging more observations for any particular $x_i$. In other words, a large $h$ allows for greater weightings $w_i$ for farther observations.

In practice, we see that this choice of $h$ is highly dependent on the data. For example, if observations are truly close to one another and have a similar true value of $r(x)$, we will achieve better results using a large $h$. To see this, consider the example set of observations in Figure 1.6,

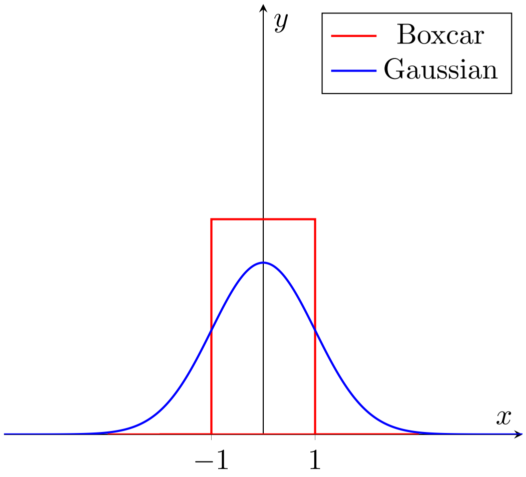

Figure 1.5: Boxcar and Gaussian Kernels

where we see small fluctuations across observations but on average not very many differences and assume that the true value is somewhere along the average of these observations. Here, an $h = 0$ would yield separate constants for each observation and no "denoising" for each observation. On the other hand, choosing $h = \infty$ here would yield the same constant prediction $\hat{r}(x_i)$ for each $x_i$, which we can see as follows:

$$\frac{1}{n}\sum_{i=1}^{n} Y_i = \frac{1}{n}\sum_{i=1}^{n} r(x_i) + \xi_i = \frac{1}{n}\sum_{i=1}^{n} c + \xi_i = c + \frac{1}{n}\sum_{i=1}^{n} \xi_i \approx c \pm \frac{1}{\sqrt{n}} \tag{1.15}$$

where we draw the simplification of the last term via the central limit theorem. [2]

If instead we had chosen a moderate $h$, our estimate for each $x_i$ may have not included all observations and then would be represented using a similar derivation as follows:

$$\hat{r}(x_i) = \frac{1}{|B_{x_i}|}\sum_{j \in B_{x_i}} Y_j = \frac{1}{|B_{x_i}|}\sum_{j \in B_{x_i}} r(x_j) + \xi_j = \frac{1}{|B_{x_i}|}\sum_{j \in B_{x_i}} c + \xi_i = c + \frac{1}{|B_{x_i}|}\sum_{j \in B_{x_i}} \xi_i \approx c \pm \frac{1}{\sqrt{|B_{x_i}|}} \tag{1.16}$$

where we can see here that we could be obtaining a noisier value for our estimate compared to a larger $h$.

Finally, considering another extreme case, where the true $r(x)$ fluctuates a lot such as the $r(x)$ in Figure 1.1, utilizing a large $h$ would yield a poorer result than simply assuming there is no noise $\xi_i$ and simply choosing $h = 0$. In such a case, although we may not be correct in our assumption, we may still obtain a reasonable estimate $\hat{r}(x_i) = Y_i$ (when $h = 0$).

[2] Assuming $\mathrm{Var}(\xi_i) = 1$, the central limit theorem implies $\frac{1}{\sqrt{n}}\sum_{i=1}^{n} \xi_i \xrightarrow{\mathrm{d}} \mathsf{N}(0,1)$.

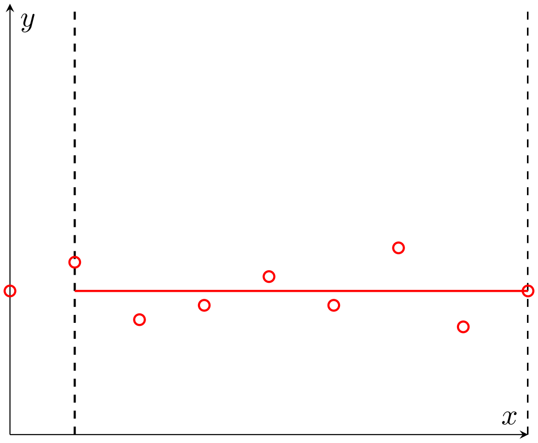

Figure 1.6: Example where choosing a large $h$ would yield a better estimate