Nonparametric Regression

Paweł Polak

January 28, 2016

STAT W4413: Nonparametric Statistics - Lecture 5

Nonparametric regression: Objective and Roadmap

Nonparametric regression is also known as "smoothing" or "function learning"

- We observe $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. Here x_i could be a random variable or a random vector. However, y_i is a random variable.
- We believe that the random variables y_i can be written as a function of x_i. More formally, we believe

$$y_i = g(x_i) + \epsilon_i,$$

where

- ϵ_i is assumed to be an iid noise with $\mathbb{E}(\epsilon_i) = 0$ and $var(\epsilon_i) = \sigma^2$
- g is an arbitrary function and we only assume that it is "smooth",[1] i.e., it is continuous and differentiable of certain order.

[1] We can extend the result to piecewise smooth functions as well.

Nonparametric Regression: Terminology

The main goal of the *nonparametric regression* is to estimate the function g from the samples $\{(x_i, y_i)\}_{i=1}^n$ under minimal assumptions, such as "smoothness" of g.

- In this problem the components of x are called covariates or features.
- y is called the response variable.
- The set $\{(x_i, y_i)\}_{i=1}^n$ is known as the training set and
- g is called the regression function.

We will study two major categories of nonparametric regression:

1. Local regression methods: In this category we study "kernel methods" and "locally polynomial regression".
2. Penalization methods: here we study splines.

To simplify and understand this problem in the first few lectures we focus on the case $x_i \in \mathbb{R}$. Later we will extend our results to $x \in \mathbb{R}^p$.

Parametric regression

The problem of curve fitting is an important problem and arises in both parametric and nonparametric settings.

Two famous parametric approaches to deal with the problem of curve fitting are known as:

- "linear regression" and
- "polynomial regression".

Linear regression

In the linear regression, we consider a very simple parametric model for the function g, i.e.,

$$g(x) = \beta_0 + \beta_1 x.$$

Then we first estimate β_0 and β_1 and use their estimates to obtain an estimate of g.
The most popular approach for estimating β_0 and β_1 is through the optimization problem

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg \min_{\beta_0, \beta_1} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \tag{1}$$

that is also known as least square problem for obvious reasons.

Linear regression

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg \min_{\beta_0, \beta_1} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \tag{2}$$

The solution of this optimization problem has a very nice and simple form.

Define $\beta \triangleq [\beta_0, \beta_1]^T$ and $y \triangleq [y_1, y_2, \ldots, y_n]^T$, where T is the transpose operator. Also let the matrix X be defined in the following way:

$$X \triangleq \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \tag{3}$$

Them the solution of the least squares problem is given by

$$\hat{\beta} = (X^T X)^{-1} X^T y.$$

Once we have an estimate of β, we can easily estimate g with

$$\hat{g}(x) = [1, x]\hat{\beta}.$$

Polynomial regression

- In many situations, the linear models are too restrictive and do not fit the data well.

- A straightforward extension is the polynomial regression.
- In the polynomial regression we assume that a polynomial of degree ℓ is a good model for g, i.e., $g(x) \approx \beta_0 + \beta_1 x + \ldots + \beta_\ell x^\ell$.

Then as in the case of linear regression we first estimate $\beta_0, \ldots, \beta_\ell$ and then estimate

$$\hat{g}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \ldots + \hat{\beta}_\ell x^\ell.$$

A popular approach for obtaining an estimate of $\beta_0, \ldots, \beta_\ell$ is

$$(\hat{\beta}_0, \ldots, \hat{\beta}_\ell) = \arg \min_{\beta_0, \ldots, \beta_\ell} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i - \ldots \beta_\ell x_i^\ell)^2. \qquad (4)$$

Polynomial regression

As in the linear regression setting, solving this optimization problem is straightforward. Define $\beta \triangleq [\beta_0, \ldots, \beta_\ell]^T$, and $y \triangleq [y_1, \ldots, y_n]^T$. Also define the matrix X as

$$X \triangleq \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^\ell \\ 1 & x_2 & x_2^2 & \ldots & x_2^\ell \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^\ell \end{bmatrix}. \tag{5}$$

Then the solution of (4) is equal to

$$\hat{\beta} = (X^T X)^{-1} X^T y.$$

Note that the difference between these two estimates is in matrix X. Clearly, the polynomial regression, can capture more nonlinearities. Finally once we have an estimate of β we obtain an estimate of g through

$$\hat{g}(x) = [1, x, x^2, \ldots, x^\ell]\hat{\beta}.$$

Basis expansions

- The two cases described above are special cases of the so called "basis expansion".
- In the "basis expansion", instead of assuming that g can be represented as a polynomial, we assume that it can be written as a linear combination of certain basis functions.
- Calling these basis functions $h_0(x), h_1(x), h_2(x), \ldots, h_\ell(x)$, we assume that $g(x) \approx \beta_0 h_0(x) + \beta_1 h_1(x) + \ldots + \beta_\ell h_\ell(x)$.

Before we describe how we estimate $\beta_0, \ldots, \beta_\ell$, let's review a few examples of such basis functions.

Examples

Example

Piecewise constant regression Let $h_m(x) = \mathbb{I}(x \in [\frac{m}{\ell+1}, \frac{m+1}{\ell+1}])$ for $m = 0, 1, \ldots \ell$. Such basis enables us to approximate g with a piecewise constant function.

Example

Polynomial regression Let $h_m(x) = x^m$. Then the problem is equivalent to polynomial regression.

Examples

Example

Piecewise polynomial regression Consider the following basis functions: $h_m^0(x) = \mathbb{I}(x \in [\frac{m}{\ell+1}, \frac{m+1}{\ell+1}]), h_m^1(x) = x\mathbb{I}(x \in [\frac{m}{\ell+1}, \frac{m+1}{\ell+1}]), \ldots, h_m^k(x) = x^k\mathbb{I}(x \in [\frac{m}{\ell+1}, \frac{m+1}{\ell+1}])$ for $m = 0, 1, \ldots \ell$. These bases enable us to approximate the function g with a piecewise polynomial function. We will get back to this example later in the spline section.

Example

Setting $h_m(x)$ to other nonlinearities such as $\log(x)$ or $\sqrt{x}$ might also be useful for certain applications.

Basis expansion

Once we come up with a set of basis functions $h_m(x)$ for which we believe $g(x) \approx \sum_{i=0}^{\ell} \beta_i h_i(x)$ we can use very similar techniques to find an estimate of g: we first calculate an estimate of β through the minimization problem

$$(\hat{\beta}_0, \ldots, \hat{\beta}_\ell) = \arg \min_{\beta_0,\ldots,\beta_\ell} \sum_{i=1}^{n} (y_i - \beta_0 h_0(x_i) - \beta_1 h_1(x_i) - \ldots - \beta_\ell h_\ell(x_i))^2. \tag{6}$$

and then use $(\hat{\beta}_0, \ldots, \hat{\beta}_\ell)$ to estimate g through $\hat{g}(x) = \sum_{i=0}^{\ell} \hat{\beta}_i h_i(x)$. Furthermore, this optimization can be solved easily. Define $\beta \triangleq [\beta_0, \ldots, \beta_\ell]^T$, and $y \triangleq [y_1, \ldots, y_n]^T$. Also define the matrix X as

$$X \triangleq \begin{bmatrix} h_0(x_1) & h_1(x_1) & h_2(x_1) & \ldots & h_\ell(x_1) \\ h_0(x_2) & h_1(x_2) & h_2(x_2) & \ldots & h_\ell(x_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_0(x_n) & h_1(x_n) & h_2(x_n) & \ldots & h_\ell(x_n) \end{bmatrix}. \tag{7}$$

Then the solution of (6) is again given by $(X^T X)^{-1} X^T y$. Try to prove this for yourself.

Basis expansion vs. nonparametric regression

- One of the main challenges in the basis expansion approach is to come up with proper bases that describe the g function well.
- Unfortunately this is a non-trivial problem.
- Specially when we have more than one feature or covariate.
- In such cases we use the nonparametric approaches that make minimal assumptions on g and are still able to recover it in many cases with acceptable accuracy.
- Specially when the number of observations n is high.

Naive approach

Example

Suppose that we would like to estimate a function g at point x from observations $\{(x_i, y_i)\}_{i=1}^n$.

The simplest approach would be to find x_i that is closest to x, i.e.,

$$x_i = arg \min_{z \in \{x_1, \ldots, x_n\}} |x - z|,$$

from the training set and estimate

$$\hat{g}(x) = y_i.$$

This is called the nearest neighbor estimate of $g(x)$. Since g is smooth (at least continuous), we expect that if x_i is close enough to x then $g(x_i)$ will be also close to $g(x)$ and since $y_i = g(x_i) + \epsilon_i$, then we estimate $g(x)$ with y_i.

Is this a good estimate?

Bias vs. Variance

In many cases the answer is no.

In order to understand why, let's calculate the error of this estimator. We define the error as

$$\mathbb{E}(\hat{g}(x) - g(x))^2.$$

$$\mathbb{E}(\hat{g}(x)-g(x))^2 = \mathbb{E}(y_i-g(x))^2 = \mathbb{E}(g(x_i)+\epsilon_i-g(x))^2 = \sigma^2+\mathbb{E}(g(x_i)-g(x))^2.$$

- σ^2 is called the variance of the estimator (which is due to noise), and
- the second term is called *bias*.

Note that the last expectation is with respect to the randomness in choosing x_i. (You will see some examples in the next homework.)

What is not so good about this estimator?

Bias vs. Variance

$$\mathbb{E}(\hat{g}(x) - g(x))^2 = \sigma^2 + \mathbb{E}(g(x_i) - g(x))^2.$$

- Suppose that we have access to a very large training set.
- Then x_i will be extremely close to x, and so is $g(x_i)$ to $g(x)$ (because of smoothness).

Therefore the dominant term in the error is the variance that remains constant no matter how many samples we have.

Bias vs. Variance

KNN

It turns out that we can do much better. Here is a simple example.

- Suppose that instead of considering only the nearest neighbor of x.
- We keep k nearest neighbors of x. Let's say their indices are $x_{j_1}, \ldots, x_{j_k}$.

Now we estimate $g(x)$ with

$$\hat{g}(x) = \frac{1}{k} \sum_{i=1}^{k} y_{j_i}. \tag{8}$$

Bias vs. Variance

As before, we start calculating the mean square error of this estimator.

$$\begin{aligned}
\mathbb{E}(g(x)-\hat{g}(x))^2 &= \mathbb{E}\left(g(x)-\frac{1}{k}\sum_{i=1}^{k} y_{j_i}\right)^2 \\
&= \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k}\epsilon_{j_i}\right)^2 + \mathbb{E}\left(g(x)-\frac{1}{k}\sum_{i=1}^{k} g(x_{j_i})\right)^2 \\
&= \frac{\sigma^2}{k} + \mathbb{E}\left(g(x)-\frac{1}{k}\sum_{i=1}^{k} g(x_{j_i})\right)^2 . \qquad (9)
\end{aligned}$$

- Again as $n \to \infty$, the bias term goes to zero (if k is not too large).
- Also the remaining variance term is much smaller than the nearest neighbor estimator.

Bias vs. Variance

- Now that it seems we can reduce the variance by considering more samples in the average.
- Let's consider the extreme case where all the data points contribute to our estimate, i.e., $g(x) = \frac{1}{n}\sum_{i=1}^{n} Y_i$.
- This is the k-nearest neighbor estimator with $k = n$.
- It is needless to say that this estimator is not good. Why?

Because in this scenario the bias term that was negligible in the last two examples is not small any more and bias will be the issue of our estimator.

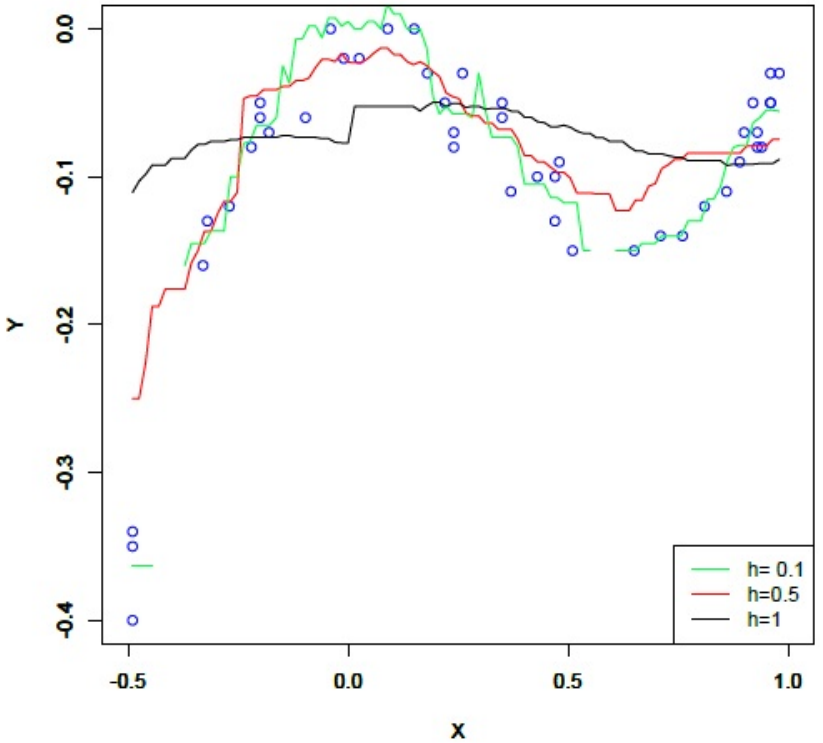
Bias vs. Variance

This example illustrates a few things:

1. It seems that using the samples in the neighborhood of point x for estimating $g(x)$ may lead to good estimates.

2. If the estimator is too local, meaning that it is the average of very few samples (e.g. 1 or 2), then the estimator suffers from high variance. If the estimator is too global then the estimator suffers from a high bias. There is a sweet spot on the neighborhood size for which such algorithms perform the best.

These two ideas are the main bases for Kernel smoothing and locally linear regression that we will see next.



Box kernel regression

Consider the following simple estimator of $g(x)$:

$$\hat{g}(x) = \frac{\sum_{i=1}^{n} \mathbb{I}(|x_i - x| \leq h) y_i}{\sum_{i=1}^{n} \mathbb{I}(|x_i - x| \leq h)} \qquad (10)$$

Lowess method

- This estimator is in spirit very similar to the estimator we already described:
  - The function $\mathbb{I}(|x_i - x| \leq h)$ ensures that only the samples that are in the neighborhood of x contribute to the estimate. If the distance of a sample from x is larger than h, it will not contribute to the estimate.
  - Careful examination of the formula shows that the estimate is essentially the average of the samples in the neighborhood.
- h is called "bandwidth" and controls the bias and variance of the estimator. If h is large (more global estimate) the bias is large and the variance is small. If h is small, then the variance is going to be large and the bias will be small (compare this estimator with k-nearest neighbor and convince yourself that increasing h is in spirit similar to increasing k).
- It is very important to find a good value for h.
- See the last Figure for the impact of h on the performance of the estimator.

We will discuss the practical approaches for finding the optimal value of h. But for now let's study the bias and variance of this estimator more carefully.