Nonparametric Regression

Paweł Polak

February 2, 2016

STAT W4413: Nonparametric Statistics - Lecture 6

Box kernel regression

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

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

- 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.

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.

Box kernel regression

Let's characterize the mean square error of this estimator in terms of h. Assume that:

- g is a differentiable function with $|g'(x)| < C$, where C is a constant.
- the training set has equispaced samples, i.e., $x_i = \frac{i}{n}$ and $y_i = g(x_i) + \epsilon_i$. (The other case where $x_i s$ are random leads to similar results.)

Call the $x_i s$ that are in the neighborhood of x, $x_{j_1}, \dots, x_{j_m}$.

- It is straightforward to confirm that m, the number of points that fall in this neighborhood, satisfies $\lceil 2nh \rceil - 1 \leq m \leq \lceil 2nh \rceil + 1$. Why? 笔记本

Hence, we have

$$\mathbb{E}(g(x) - \hat{g}(x))^2 = \mathbb{E}(g(x) - \frac{1}{m}\sum_{p=1}^{m} Y_{j_p})^2 = \frac{\sigma^2}{m} + (g(x) - \frac{1}{m}\sum_{p=1}^{m} g(x_{j_p}))^2. \qquad (2)$$

- Note that $|x_{j_p} - x| \leq h$. Therefore, according to the "mean value theorem"

可以尝试更高阶的泰勒展开

$$|g(x_{j_p}) - g(x)| = |g'(x')||x - x_{j_p}| \leq Ch,$$

中值定理

where x' is in the interval between x and x_{j_p}. Therefore, MSE is upper bounded by

$$\mathbb{E}(g(x) - \hat{g}(x))^2 = \frac{\sigma^2}{m} + (g(x) - \frac{1}{m}\sum_{p=1}^{m} g(x_{j_p}))^2 \leq \frac{\sigma^2}{m} + C^2h^2. \qquad (3)$$

- Note that if $nh \to 1$, then $m \approx 2nh$. Therefore, it is clear from this equation that as h decreases the bias decreases and the variance increases. What is the best value of h? In order to calculate the best value of h we can take the derivative and set it to zero. If we do so, we observe that...

Box kernel regression: optimal h

$$h^* = \left(\frac{\sigma^2}{4nC^2} \right)^{\frac{1}{3}}. \tag{4}$$

If we plug (4) in (3) we obtain the following upper bound for the mean square error:

$$\mathbb{E}(g(x) - \hat{g}(x))^2 \leq C' \frac{\sigma^{4/3}}{n^{2/3}}, \tag{5}$$

-2/3收敛阶数

where C' is a constant that only depends on C, and does not depend on σ or n.

Box kernel regression: Err_{Tot}

- So far, we have considered the error of estimating a function g at only one point x.
- In many situations our goal is to estimate the entire function.

Does this change our argument?

Not really.

Suppose that we define the error on the entire interval in the following way:

$$Err_{Tot} = \int_0^1 \mathbb{E}(g(x) - \hat{g}(x))^2 dx.$$

- Since the upper bound we calculated on $\mathbb{E}(g(x) - \hat{g}(x))^2$ is independent of x, then we can bound Err_{Tot} with $C' \frac{\sigma^{4/3}}{n^{2/3}}$ as well.

- You can consider this error either as an error in estimating the function at only one point x or the error in estimating the entire function.

Box kernel regression

- How do we interpret the error?
- Usually people are interested in large sample size regime, n very large, and consider the rate at which the error goes to zero in terms of n.

n^(-q) q阶收敛速度

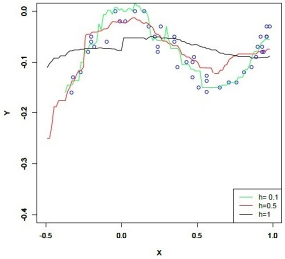
- The faster the decay is the better the estimator will be.
- According to this criterion, we may ask whether this estimator is the best estimator in the nonparametric regime?
- We will get back to this point and will discuss some ways to improve the performance of such estimators.

Box kernel regression

- The estimator we have introduced in (1) is an instance of Kernel regression known as *Box kernel regression*.
- One of the major drawbacks of the box kernel regression that has limited its applications in practice, is:

The estimate from box kernel regression is very rough even when we choose a good value for the bandwidth.

- The reason is that the $\boxed{\text{Kernel } \mathbb{I}(|x_i - x| \leq h)}$ is non-differentiable.
- Can you see why this leads to rough estimates?



- In the next slides we describe how we resolve this issue...

Smooth Kernel regression: Gaussian kernel

- The non-differentiablity of the box kernel leads to a roughness of the final estimate.
- Since we know that function g is smooth, this rough estimate does not seem to be the right function.

Now suppose we modify (1) to the following estimate:

$$\hat{g}(x) = \frac{\sum_{i=1}^{n} K(\frac{|x_i - x|}{h}) y_i}{\sum_{i=1}^{n} K(\frac{|x_i - x|}{h})} \tag{6}$$

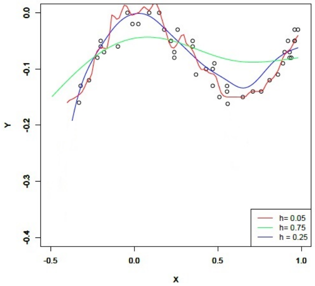
where $K(x) = e^{-\frac{x^2}{2}}$. This is called "Gaussian" kernel.

- The main advantage of Gaussian Kernel over the box Kernel is that the samples that are farther from point x, contribute less in the final estimate, i.e., their weights become smaller and so does their contribution.
- This decrease in the weight happens in a smooth way since the Gaussian Kernel is smooth and unlike the box kernel estimator there is no abrupt change in the weights.

Smooth Kernel regression: Gaussian kernel

- Figure below exhibits the performance of this estimator.
- If you compare this figure with the previous one, you see that Gaussian estimate is much smoother than the estimate we obtained from the box Kernel.
- Notice that the effect of h is the same as the effect of h in the box estimator.
- If h is large then our estimator is going to oversmooth data (high bias). If it is small, then our estimator is going to undersmooth the data (passes through all the points). This corresponds to high variance.

So, again it is important to choose the right choice of h and we will discuss practical ways to do this later.



Smooth Kernel regression: other kernels

Instead of Gaussian kernel one can use other smooth kernels as well. For instance two well-known kernels that are used in practice are

1. Epanechnikov: $K(x) = 0.75(1 - x^2)\mathbb{I}(|x| \leq 1)$.
2. Tricube Kernel: $K(x) = \frac{70}{81}(1 - |x|^3)^3\mathbb{I}(|x| \leq 1)$.

- One advantage of these two kernels over the Gaussian Kernel is that with Gaussian Kernel even the samples that are far from x have some contribution in the estimate (even though it is very small).
- But with these two kernels the contribution of far apart samples is going to be zero. Why?
- Nevertheless, from the practical perspective you don't see much difference in the performance of these three kernels.
- A parameter that has a major impact on the performance is going to be the bandwidth h and we will discuss how we should tune this parameter in practice later in the lecture.

Smooth kernels vs. box kernel

- Does Gaussian kernel (or some other kernel) improve the mean square error or not?

  No, no se nada of true f

- For the box kernel, we provide an upper bound on the mean square error. This is due to the fact that the exact nature of the function is not known.
- Characterizing the exact value of mean square error (with correct constants) is a very challenging task and there has not been much progress in that direction.
- Therefore, what we usually do for comparing different estimation algorithms is to assume that n is very large, and measure the decay rate of mean square error in terms of n.
- For instance, in the case of the box Kernel we realized that the mean square error decays as $n^{-2/3}$.
- If we can find an estimator that decays faster (e.g. as n^{-1}) we would call it a better estimator.

Does the Gaussian Kernel estimator (or Epanechnikov or Tricube Kernel) provide a better rate of decay for MSE than the box kernel?

The answer to this question is "NO". So, from the theoretical perspective they are more or less similar to the box estimator. But since they provide a smooth estimate we prefer them in practice. For more information refer to the book "Minimax Theory of Image Reconstruction" by Tsybakov.

Equispaced grid

So far we let X_i to be on a equispaced grid. This simplifies our theoretical analysis of the MSE.

- in many cases this assumption is not true. A better model is to assume that X_i's are randomly drawn from a pdf $f(x)$.
- This is known as $\boxed{\text{random design framework.}}$
- It turns out that if $f(x)$ is uniform, then we can prove an upper bound which is very similar to the upper bound we provided for equispaced grid.
- However, if it is not uniform, then the variance of the estimator depends on $1/f(x)$.

Instead of proving this result we intuitively argue why this is the case.

- In fact if the size of the interval is $2h$ and h is small then the probability that a point contributes to the estimate is going to be approximately $2f(x)h$.
- If n is large, then on average the number of points that will contribute in the estimate is going to be $2nf(x)h$.
- Since the variance is inversely proportional to the number of points that contribute to the average, we obtain $\frac{\sigma^2}{2f(x)hn}$ for the variance.

More data points, larger n, better estimation

Locally linear or polynomial regression

- We discussed that as the number of samples we have goes to infinity the mean square error of kernel regression goes to zero with the rate $n^{2/3}$.
- The question we would like to address now is *whether one can improve this rate by using a different estimator*.

A new look at kernel regression

The kernel regression method we discussed in the last slides can be cast as a result of a simple optimization problem. Consider estimating function g at point x. We do so by solving the following optimization problem:

$$\mathop{arg\,min}_{\beta_0} \sum_{i=1}^{n} K\left(\frac{|x-x_i|}{h}\right)(y_i-\beta_0)^2. \quad =:Q \tag{7}$$

What is this optimization problem about? We assume that our function value at x is equal to β_0. β_0 should be close to y_i if x_i is close to x. By assigning higher weights to the samples that are closer to x, we ensure that β_0 is a good estimate of $g(x)$.

The solution to this optimization problem is given by

$$\beta_0^* = \frac{\sum_{i=1}^{n} K\left(\frac{|x_i-x|}{h}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{|x_i-x|}{h}\right)},$$

Solve for β:
dQ/dβ=0

which is essentially the same as the result of Kernel regression.

- If we set K to a constant, i.e., we ignore the existence of the kernel, then the formulation in (7) corresponds to estimating the function with a constant function.
- However, the existence of Kernel in the formulation, gives more weight to the points that are closer to x.
- This means that the Kernel method is assuming that the value of $g(x)$ is the same as the value of $g(x_i)$ when x_i's are close to x.
- With this new intuition of the Kernel regression, we can easily come up with new methods to improve the Kernel regression. How can we do it?

Locally linear/polynomial regression

- If we look at the Taylor expansion of $g(x)$ we immediately notice a new way to improve the Kernel estimate.
- Instead of assuming that $g(x_i) \approx g(x)$ in the neighborhood of x, we can assume that $g(x_i) \approx g(x) + g'(x)(x_i - x)$.
- This is a linear approximation of g in the neighborhood of x.
- If we make this assumption, then the error of this approximation is at the order of $|g''(c)||x - x_i|^2/2$ for some $c \in [x, x_i]$.
- This is the result of the extension of mean value theorem.
- When $|x - x_i|$ is small, $g''(x)|x - x_i|^2/2$ is much smaller than $|x - x_i|$. The main question is how we can incorporate this higher order model in the formulation of the Kernel regression.
- Formulation (7) is our key. In fact, we can simply modify that to

$$\boxed{arg \min_{\beta_0, \beta_1} \sum_{i=1}^{n} K\left(\frac{|x - x_i|}{h}\right)(y_i - \beta_0 - \beta_1 x_i)^2.} \tag{8}$$

Once we have access to β_0 and β_1 we can estimate $\hat{g}$ by $\hat{g}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$. We will analyze the performance of this algorithm in the next section and show how/when it outperforms the kernel regression technique.

Locally linear/polynomial regression

But before doing so let us also mention an obvious extension of this result, that is, locally polynomial regression.

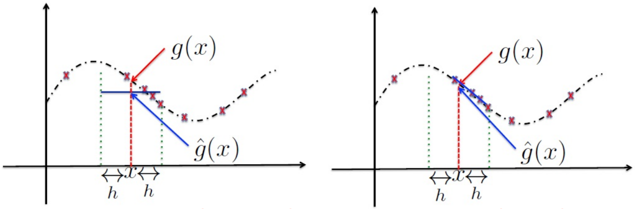
- Again from the Taylor expansion we can claim that we can obtain an even better estimate if we go to higher order polynomial. For instance we can also consider $g(x_i) \approx g(x) + g'(x)(x_i - x) + \frac{1}{2}g''(x)(x_i - x)^2$.
- The error will then be at the order of $|x - x_i|^3$.
- In general the formulation of the locally polynomial regression for a polynomial of order M is equivalent to

$$arg \min_{\beta_0,\beta_1,\ldots,\beta_M} \sum_{i=1}^{n} K\left(\frac{|x - x_i|}{h}\right)\left(y_i - \sum_{j=1}^{M} \beta_j x_i^j\right)^2. \tag{9}$$

Once we have the estimates $\hat{\beta}_i$, we can estimate $\hat{g}(x) = \sum_{j=0}^{M} \hat{\beta}_j x^j$.

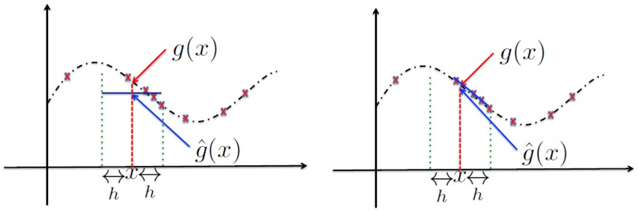
Bias, variance trade-off

- Suppose that we observed $(X_i, Y_i)_{i=1}^n$ with $Y_i = g(X_i)$.
- Note that there is no noise on our observation.
- Skipping the noise lets us ignore the variance of the estimator and discuss its bias only.
- We would like to show that often using higher order polynomials in locally polynomial regression reduces the bias of the estimate.
- To make the discussion even simpler we consider only the box kernel.
- Figure below compares the bias of locally linear regression with Kernel regression.



简单函数（阶梯型）->辛普森函数（梯形）

Bias, variance trade-off



- As is clear from this figure the linear model lets us have a better fit to the data and hence has smaller bias.
- From this figure it seems that as we increase the degree of the polynomial we obtain a better fit to the curve and hence we expect the estimator to have smaller bias.
- This intuition is essentially coming from the Taylor series expansion of the function g.
- According to the Taylor series expansion as we consider a polynomial with higher degree we get a better approximation of the function.
- However, remember that if at certain degree M the function is not differentiable, we cannot expect the polynomials of order $M + 1$ and above to give us any good estimate.
- In practice as we will see later, very high degree polynomials are not very useful and we usually use polynomials of degree one or two. We will compare the biases in the next slides.

Bias, variance trade-off

What about the variance of the estimator?

- Clearly, in the last Figure we skipped the noise.
- Therefore, we do not see the impact of the variance.
- However, it turns out that as we increase the degree of the polynomial the variance increases.
- This is due to the fact that as we increase the complexity of the model, it will have a higher risk to overfit to the noise.
- The accurate characterization of this issue is out of scope of this course. But interested readers are referred to the book by Tsybakov and also Problems 6.2 and 6.3 from the book by Hastie, Tibshirani, and Friedman.

Large sample size analysis

In the last slides, we argued that heuristically locally linear and locally polynomial regression algorithms can outperform the kernel regression methods. However, such discussion raises several major questions.

- For instance in what situations locally polynomial/linear algorithms are better than kernel regression?
- What is the optimal degree we should use for polynomials.
- We will now settle these two major questions.
- Functions that we usually deal with in practice are not infinitely many times differentiable.
- While they are smooth, their first or second derivatives are not smooth any more.
- Hence, a more realistic scenario would be to consider functions that are $M+1$ times differentiable and their $(M+1)^{th}$ derivative is bounded meaning $|f^{(M+1)}(x)| \leq C$ for every x.
- It turns out that in such setting the best choice for the degree of the polynomial is M.

Large sample size analysis

Furthermore, we have the following theorem

Theorem

The MSE of locally polynomial regression with box kernel and degree M is upper bounded by

$$MSE_x \leq \frac{C_1\sigma^2}{nh} + C_2 h^{2M+2}, \qquad (10)$$

where C_1 and C_2 are two constants that are independent of h, n, σ.

Note that the constants C_1 and C_2 may depend on M. If you are interested in the proof of this theorem check it in the book by Tsybakov.

Large sample size analysis

As we expected before since we are using a better model the bias of the estimator has decreased. This means that the overall mean square error has decreased.

- Let's do what we did for the kernel regression; calculate the optimal choice of h and compare the decay of the MSE in terms of n in Kernel and locally polynomial regression.

The optimal choice of h is given by

$$h^* = \left(\frac{C_1}{C_2(2M+2)} \right)^{\frac{1}{2M+3}} \frac{\sigma^{2/(2M+3)}}{n^{1/(2M+3)}}.$$

If we plug h^* in the MSE formula of (10) we obtain

$$MSE_x = C' n^{-\frac{2M+2}{2M+3}},$$

where C' is a constant that does not depend on M.

If we compare this rate with the rate of decay of the Kernel regression, we see that the rate at which this estimator goes to zero is higher.
impact on the constant C_1 in Theorem 1 and usually C_1 increases as we increase M. This does not cause any problem when we have many samples available. However, when the number of samples is small even increasing M may cause overfitting.

Next time: Selecting bandwidth h and degree of polynomial M