STAT/Q SCI 403: Introduction to Resampling Methods **Spring 2018**

Lecture 8: Nonparametric Regression

Instructor: Yen-Chi Chen

8.1 Introduction

Let $(X_1, Y_1), \cdots, (X_n, Y_n)$ be a bivariate random sample. In the regression analysis, we are often interested in the regression function

$$m(x) = \mathbb{E}(Y|X = x).$$

Sometimes, we will write

$$Y_i = m(X_i) + \epsilon_i,$$

where ϵ_i is a mean 0 noise. The simple linear regression model is to assume that $m(x) = \beta_0 + \beta_1 x$, where β_0 and β_1 are the intercept and slope parameter. In this lecture, we will talk about methods that direct estimate the regression function $m(x)$ without imposing any parametric form of $m(x)$.

Given a point x_0, assume that we are interested in the value $m(x_0)$. Here is a simple method to estimate that value. When $m(x_0)$ is smooth, an observation $X_i \approx x_0$ implies $m(X_i) \approx m(x_0)$. Thus, the response value $Y_i = m(X_i) + \epsilon_i \approx m(x_0) + \epsilon_i$. Using this observation, to reduce the noise ϵ_i, we can use the sample average. Thus, an estimator of $m(x_0)$ is to take the average of those responses whose covariate are close to x_0.

To make it more concrete, let $h > 0$ be a threshold. The above procedure suggests to use

$$\widehat{m}_{\mathsf{loc}}(x_0) = \frac{\sum_{i:|X_i - x_0| \le h} Y_i}{n_h(x_0)} = \frac{\sum_{i=1}^n Y_i I(|X_i - x_0| \le h)}{\sum_{i=1}^n I(|X_i - x_0| \le h)}, \tag{8.1}$$

where $n_h(x_0)$ is the number of observations where the covariate $X : |X_i - x_0| \le h$. This estimator, $\widehat{m}_{\mathsf{loc}}$, is called the *local average* estimator. Indeed, to estimate $m(x)$ at any given point x, we are using a local average as an estimator.

The local average estimator can be rewritten as

$$\widehat{m}_{\mathsf{loc}}(x_0) = \frac{\sum_{i=1}^n Y_i I(|X_i - x_0| \le h)}{\sum_{i=1}^n I(|X_i - x_0| \le h)} = \sum_{i=1}^n \frac{I(|X_i - x_0| \le h)}{\sum_{\ell=1}^n I(|X_\ell - x_0| \le h)} \cdot Y_i = \sum_{i=1}^n W_i(x_0) Y_i, \tag{8.2}$$

where

$$W_i(x_0) = \frac{I(|X_i - x_0| \le h)}{\sum_{\ell=1}^n I(|X_\ell - x_0| \le h)} \tag{8.3}$$

is a weight for each observation. Note that $\sum_{i=1}^n W_i(x_0) = 1$ and $W_i(x_0) > 0$ for all $i = 1, \cdots, n$; this implies that $W_i(x_0)$'s are indeed weights. Equation (8.2) shows that the local average estimator can be written as a *weighted average* estimator so the i-th weight $W_i(x_0)$ determines the contribution of response Y_i to the estimator $\widehat{m}_{\mathsf{loc}}(x_0)$.

In constructing the local average estimator, we are placing a hard-thresholding on the neighboring points– those within a distance h are given equal weight but those outside the threshold h will be ignored completely. This hard-thresholding leads to an estimator that is not continuous.

To avoid problem, we consider another construction of the weights. Ideally, we want to give more weights to those observations that are close to x_0 and we want to have a weight that is 'smooth'. The Gaussian function $G(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ seems to be a good candidate. We now use the Gaussian function to construct an estimator. We first construct the weight

$$W_i^G(x_0) = \frac{G\left(\frac{x_0 - X_i}{h}\right)}{\sum_{\ell=1}^n G\left(\frac{x_0 - X_\ell}{h}\right)}.$$

The quantity $h > 0$ is the similar quantity to the threshold in the local average but now it acts as the *smoothing bandwidth* of the Gaussian. After constructing the weight, our new estimator is

$$\widehat{m}_{\mathsf{G}}(x_0) = \sum_{i=1}^n W_i^G(x_0) Y_i = \sum_{i=1}^n \frac{G\left(\frac{x_0 - X_i}{h}\right)}{\sum_{\ell=1}^n G\left(\frac{x_0 - X_\ell}{h}\right)} Y_i = \frac{\sum_{i=1}^n Y_i G\left(\frac{x_0 - X_i}{h}\right)}{\sum_{\ell=1}^n G\left(\frac{x_0 - X_\ell}{h}\right)}. \tag{8.4}$$

This new estimator has a weight that changes more smoothly than the local average and is smooth as we desire.

Observing from equation (8.1) and (8.4), one may notice that these *local* estimators are all of a similar form:

$$\widehat{m}_h(x_0) = \frac{\sum_{i=1}^n Y_i K\left(\frac{x_0 - X_i}{h}\right)}{\sum_{\ell=1}^n K\left(\frac{x_0 - X_\ell}{h}\right)} = \sum_{i=1}^n W_i^K(x_0) Y_i, \quad W_i^K(x_0) = \frac{K\left(\frac{x_0 - X_i}{h}\right)}{\sum_{\ell=1}^n K\left(\frac{x_0 - X_\ell}{h}\right)}, \tag{8.5}$$

where K is some function. When K is a Gaussian, we obtain estimator (8.4); when K is a uniform over $[-1, 1]$, we obtain the local average (8.1). The estimator in equation (8.5) is called the *kernel regression* estimator or Nadaraya-Watson estimator[1]. The function K plays a similar role as the kernel function in the KDE and thus it is also called the *kernel function*. And the quantity $h > 0$ is similar to the smoothing bandwidth in the KDE so it is also called the smoothing bandwidth.

8.2 Cross-Validation

The smoothing bandwidth h has to be chosen to construct our estimator. But in practice, how can we choose it? A good news is that–unlike the density estimation problem, there is a simple approach to choose h: the cross-validation (CV)[2].

Before we discuss the details of CV, we first introduce the *predictive risk*. Let $\widehat{m}_h$ be the kernel regression using the n observations. Let X_{n+1}, Y_{n+1} be a new observation (from the same population). We define the *predictive risk* of our regression estimator as

$$R(h) = \mathbb{E}\left(Y_{n+1} - \widehat{m}_h(X_{n+1})\right)^2. \tag{8.6}$$

Namely, the quantity $R(h)$ is the expected square error of predicting the next observation using the kernel regression.

CV is a collection of approaches that tries to estimate the predictive risk $R(h)$ using a data-splitting approach. A classical version of CV is the leave-one out cross-validation (LOO-CV):

$$\widehat{R}(h) = \frac{1}{n} \sum_{i=1}^n (Y_i - \widehat{m}_{h,-i}(X_i))^2,$$

[1] https://en.wikipedia.org/wiki/Kernel_regression
[2] https://en.wikipedia.org/wiki/Cross-validation_(statistics)

where $\widehat{m}_{h,-i}(X_i)$ is the kernel regression using all observations except i-th observation X_i, Y_i. Namely, LOO-CV leaves each observation out once at a time and use the remaining observations to train the estimator and evaluate the quality of the estimator using the left out observation. The main reasoning of such a procedure is to make sure we do not use the data twice.

Another popular version of CV is the K-fold CV. We randomly split the data into K equal size groups. Each time we leave out one group and use the other K-1 groups to construct our estimator. Then we use the left out group to evaluate the risk. Repeat this procedure many times and take the average as the risk estimator $\widehat{R}(h)$.

K-fold Cross-Validation.

1. Randomly split $\mathcal{D} = \{(X_1, Y_1), \cdots, (X_n, Y_n)\}$ into K groups: $\mathcal{D}_1, \cdots, \mathcal{D}_K$.
2. For ℓ-th group, construct the estimator $\widehat{m}_h^{(\ell)}$ using all the data except ℓ-th group.
3. Evaluate the error by
$$\widehat{R}^{(\ell)}(h) = \frac{1}{n_\ell} \sum_{(X_i, Y_i) \in \mathcal{D}_\ell} (Y_i - \widehat{m}_h^{(\ell)}(X_i))^2$$
4. Compute the average error
$$\widehat{R}(h) = \frac{1}{K} \sum_{\ell=1}^{K} \widehat{R}^{(\ell)}(h).$$
5. Repeat the above 4 steps N times, leading to N average errors
$$\widehat{R}^{*(1)}(h), \cdots, \widehat{R}^{*(N)}(h).$$
6. Estimate $R(h)$ via
$$\widehat{R}^{*}(h) = \frac{1}{N} \sum_{\ell=1}^{N} \widehat{R}^{*(\ell)}(h).$$

The CV provides a simple approach of estimating the predictive errors. To choose the smoothing bandwidth, we pick

$$h^* = \mathsf{argmin}_{h>0} \widehat{R}(h).$$

In practice, we apply the CV to various values of h and choose the one with the minimal predictive risk. Generally, we will plot $\widehat{R}(h)$ versus h and determine if the minimal value makes sense. Sometimes there might be no well-defined minimum value (like a flat region).

Why do we want to split the data into two parts and construct the estimator on one part and evaluate the risk on the other part? The main reason is to obtain a reliable estimate of the predictive risk. If we use the same set of data to construct our estimator and evaluate the errors, the estimated predictive risk will be smaller than the actual predictive risk. To see this, consider the local average estimator with $h \approx 0$. When h is very very small, $\widehat{m}_{\mathsf{loc}}(X_i) = Y_i$ because the neighborhood only contain this single observation. In this case, the estimated predictive risk will be $\sum_{i=1}^{n}(Y_i - Y_i)^2 = 0$. This is related to the so-called *overfitting*[3].

[3] https://en.wikipedia.org/wiki/Overfitting

8.3 Theory

Now we study some statistical properties of the estimator $\widehat{m}_h$. We skip the details of derivations[4].

Bias. The bias of the kernel regression at a point x is

$$\textbf{bias}(\widehat{m}_h(x)) = \frac{h^2}{2}\mu_K \left(m''(x) + 2\frac{m'(x)p'(x)}{p(x)} \right) + o(h^2),$$

where $p(x)$ is the probability density function of the covariates $X_1, \cdots, X_n$ and $\mu_K = \int x^2 K(x)dx$ is the same constant of the kernel function as in the KDE.

The bias has two components: a curvature component $m''(x)$ and a *design* component $\frac{m'(x)p'(x)}{p(x)}$. The curvature component is similar to the one in the KDE; when the regression function curved a lot, kernel smoothing will smooth out the structure, introducing some bias. The second component, also known as the *design bias*, is a new component compare to the bias in the KDE. This component depends on the density of covariate $p(x)$. Note that in some studies, we can choose the values of covariates so the density $p(x)$ is also called the *design* (this is why it is known as the design bias).

Variance. The variance of the estimator is

$$\mathsf{Var}(\widehat{m}_h(x)) = \frac{\sigma^2 \cdot \sigma_K^2}{p(x)} \cdot \frac{1}{nh} + o\left(\frac{1}{nh}\right),$$

where $\sigma^2 = \mathsf{Var}(\epsilon_i)$ is the error of the regression model and $\sigma_K^2 = \int K^2(x)dx$ is a constant of the kernel function (the same as in the KDE). This expression tells us possible sources of variance. First, the variance increases when σ^2 increases. This makes perfect sense because σ^2 is the noise level. When the noise level is large, we expect the estimation error increases. Second, the density of covariate $p(x)$ is inversely related to the variance. This is also very reasonable because when $p(x)$ is large, there tends to be more data points around x, increasing the size of sample that we are averaging from. Last, the convergence rate is $O\left(\frac{1}{nh}\right)$, which is the same as the KDE.

MSE and MISE. Using the expression of bias and variance, the MSE at point x is

$$\textbf{MSE}(\widehat{m}_h(x)) = \frac{h^4}{4}\mu_K^2 \left(m''(x) + 2\frac{m'(x)p'(x)}{p(x)} \right)^2 + \frac{\sigma^2 \cdot \sigma_K^2}{p(x)} \cdot \frac{1}{nh} + o(h^4) + o\left(\frac{1}{nh}\right)$$

and the MISE is

$$\textbf{MISE}(\widehat{m}_h) = \frac{h^4}{4}\mu_K^2 \int \left(m''(x) + 2\frac{m'(x)p'(x)}{p(x)} \right)^2 dx + \frac{\sigma^2 \cdot \sigma_K^2}{nh} \int \frac{1}{p(x)}dx + o(h^4) + o\left(\frac{1}{nh}\right). \tag{8.7}$$

Optimizing the major components in equation (8.7) (the AMISE), we obtain the optimal value of the smoothing bandwidth

$$h_{\mathsf{opt}} = C^* \cdot n^{-1/5},$$

where C^* is a constant depending on p and K.

8.4 Uncertainty and Confidence Intervals

How do we assess the quality of our estimator $\widehat{m}_h(x)$? We can use the bootstrap to do it. In this case, empirical bootstrap, residual bootstrap, and wild bootstrap all can be applied. But note that each of them

[4] if you are interested in the derivation, check http://www.ssc.wisc.edu/~bhansen/718/NonParametrics2.pdf and http://www.maths.manchester.ac.uk/~peterf/MATH38011/NPR%20N-W%20Estimator.pdf

relies on slightly different assumptions. Let $(X_1^*, Y_1^*), \cdots, (X_n^*, Y_n^*)$ be the bootstrap sample. Applying the bootstrap sample to equation (8.5), we obtain a bootstrap kernel regression, denoted as $\widehat{m}_h^*$. Now repeat the bootstrap procedure B times, this yields

$$\widehat{m}_h^{*(1)}, \cdots, \widehat{m}_h^{*(B)},$$

B bootstrap kernel regression estimator. Then we can estimate the variance of $\widehat{m}_h(x)$ by the sample variance

$$\widehat{\mathsf{Var}}_B(\widehat{m}_h(x)) = \frac{1}{B-1}\sum_{\ell=1}^{B}\left(\widehat{m}_h^{*(\ell)}(x) - \bar{\widehat{m}}_{h,B}^*(x)\right), \quad \bar{\widehat{m}}_{h,B}^*(x) = \frac{1}{B}\sum_{\ell=1}^{B}\widehat{m}_h^{*(\ell)}(x).$$

Similarly, we can estimate the MSE as what we did in Lecture 5 and 6. However, when using the bootstrap to estimate the uncertainty, one has to be very careful because when h is either too small or too large, the bootstrap estimate may fail to converge its target.

When we choose $h = O(n^{-1/5})$, the bootstrap estimate of the variance is consistent but the bootstrap estimate of the MSE might not be consistent. The main reason is: it is easier for the bootstrap to estimate the variance than the bias. Thus, when we choose h in such a way, both bias and the variance contribute a lot to the MSE so we cannot ignore the bias. However, in this case, the bootstrap cannot estimate the bias consistently so the estimate of the MSE is not consistent.

Confidence interval. To construct a confidence interval of $m(x)$, we may use the following property of the kernel regression:

$$\sqrt{nh}\left(\widehat{m}_h(x) - \mathbb{E}(\widehat{m}_h(x))\right) \xrightarrow{D} N\left(0, \frac{\sigma^2 \cdot \sigma_K^2}{p(x)}\right)$$
$$\frac{\widehat{m}_h(x) - \mathbb{E}(\widehat{m}_h(x)}{\mathsf{Var}(\widehat{m}_h(x))} \xrightarrow{D} N(0,1).$$

Thus, using a bootstrap variance estimate, we may construct a $1-\alpha$ CI of $m(x)$ as

$$\widehat{m}_h(x) \pm z_{1-\alpha/2} \cdot \sqrt{\widehat{\mathsf{Var}}_B(\widehat{m}_h(x))}.$$

Similar to the CI of the KDE, this CI is for $\mathbb{E}(\widehat{m}_h(x))$ rather than $m(x)$. However, when we undersmooth the data (choose a small h), the bias will be much smaller than the stochastic variation so that the CI will be valid for $m(x)$ as well.

We can also use the quantile method to construct a CI. Let

$$Q_\alpha = \alpha\text{-quantile of } \left\{\widehat{m}_h^{*(1)}(x), \cdots, \widehat{m}_h^{*(B)}(x)\right\}.$$

Then a $1-\alpha$ CI of $m(x)$ is

$$[Q_{\alpha/2},\ Q_{1-\alpha/2}].$$

Here is another quantile approach of constructing a CI of $m(x)$. Let

$$S_\alpha = \alpha\text{-quantile of } \left\{|\widehat{m}_h^{*(1)}(x) - \widehat{m}_h(x)|, \cdots, |\widehat{m}_h^{*(B)}(x) - \widehat{m}_h(x)|\right\}.$$

Then a $1-\alpha$ CI of $m(x)$ can be constructed using

$$\widehat{m}_h(x) \pm S_{1-\alpha}.$$

8.5 ♦ : Advanced Topics

- ♦ **Similarity to KDE.** Many theoretical results of the KDE apply to the nonparametric regression. For instance, we can generalize the MISE into other types of error measurement between $\widehat{m}_h$ and m. We can also use derivatives of $\widehat{m}_h$ as estimators of the corresponding derivatives of m. Moreover, when we have a multivariate covariate, we can use either a radial basis kernel or a product kernel to generalize the kernel regression to multivariate case.

- ♦ **KDE and kernel regression.** The KDE and the kernel regression has a very interesting relationship. Using the given bivariate random sample $(X_1, Y_1), \cdots, (X_n, Y_n)$, we can estimate the joint PDF $p(x, y)$ as

$$\widehat{p}_n(x, y) = \frac{1}{nh^2} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) K\left(\frac{Y_i - y}{h}\right).$$

  This joint density estimator also leads to a marginal density estimator of X:

$$\widehat{p}_n(x) = \int \widehat{p}_n(x, y)dy = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right).$$

  Now recalled that the regression function is the conditional expectation

$$m(x) = \mathbb{E}(Y|X = x) = \int yp(y|x)dy = \int y\frac{p(x, y)}{p(x)}dy = \frac{\int yp(x, y)dy}{p(x)}.$$

  Replacing $p(x, y)$ and $p(x)$ by their corresponding estimators $\widehat{p}_n(x, y)$ and $\widehat{p}_n(x)$, we obtain an estimate of $m(x)$ as

$$\begin{aligned}
\widehat{m}_n(x) &= \frac{\int y\widehat{p}_n(x, y)dy}{\widehat{p}_n(x)} \\
&= \frac{\int y\frac{1}{nh^2}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) K\left(\frac{Y_i - y}{h}\right) dy}{\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)} \\
&= \frac{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) \cdot \int y \cdot K\left(\frac{Y_i - y}{h}\right)\frac{dy}{h}}{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)} \\
&= \frac{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) Y_i}{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)} \\
&= \frac{\sum_{i=1}^{n} Y_i K\left(\frac{X_i - x}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)} \\
&= \widehat{m}_h(x).
\end{aligned}$$

  Note that when $K(x)$ is symmetric, $\int y \cdot K\left(\frac{Y_i - y}{h}\right)\frac{dy}{h} = Y_i$. Namely, we may understand the kernel regression as an estimator inverting the KDE of the joint PDF into a regression estimator.

- ♦ **Other approaches.** Nonparametric regression is a very important topic in statistics. There are many other approaches such as the local polynomial regression[5], spline approach[6], basis approach[7], regression trees[8] ...etc[9].

[5] http://www.maths.manchester.ac.uk/~peterf/MATH38011/NPR%20local%20Linear%20Estimator.pdf

[6] https://en.wikipedia.org/wiki/Smoothing_spline

[7] http://www.biostat.umn.edu/~johnh/pubh8422/notes/Basis_Approaches.pdf and http://www.cs.columbia.edu/~jebara/4771/tutorials/regression.pdf

[8] http://www.stat.cmu.edu/~cshalizi/350-2006/lecture-10.pdf

[9] a brief introduction: http://wwwf.imperial.ac.uk/~bm508/teaching/AppStats/Lecture7.pdf