36-705: Intermediate Statistics **Fall 2017**

Lecture 27: November 6

Lecturer: Siva Balakrishnan

In today's lecture we will discuss the problem of non-parametric regression.

Recall that, broadly in regression our goal is to estimate the regression function

$$r(x) = \mathbb{E}[Y|X = x].$$

Unlike the CDF which we could estimate with no assumptions about the distribution, here we will need *smoothness* assumptions, i.e. we will need to assume that $r(x)$ is a smooth function of x. This allows us to gain statistical strength by averaging near by points.

Suppose we construct an estimate $\widehat{r}(x)$. Then a natural measure of how well we do is the squared loss, except since these are functions this is called the *integrated* squared loss, i.e.:

$$L(\widehat{r}, r) = \int (\widehat{r}(x) - r(x))^2 dx.$$

The risk is then just the expected loss, i.e.:

$$R(\widehat{r}, r) = \mathbb{E}\left(\int (\widehat{r}(x) - r(x))^2 dx\right).$$

As in the case of point estimation we have a bias variance decomposition. First we define the point-wise bias:

$$b(x) = \mathbb{E}(\widehat{r}(x)) - r(x),$$

and the point-wise variance:

$$v(x) = \mathbb{E}(\widehat{r}(x) - \mathbb{E}(\widehat{r}(x)))^2.$$

Now, as before we can verify that:

$$R(\widehat{r}, r) = \int b^2(x)dx + \int v(x)dx.$$

A natural strategy in non-parametric regression is to locally average the data, i.e. our estimate of the regression function at any point will be the average of the Y values in a small neighborhood of the point.

The width of this neighborhood will determine the bias and variance. Too large a neighborhood will result in high bias and low variance (this is called oversmoothing) and too small a neighborhood will result in low bias but large variance (this is known as undersmoothing).

27.1 Optimal Regression Function

Suppose we knew the joint distribution over (X, Y). One could alternatively begin by defining the risk of an estimate $\widehat{r}$ as

$$R(\widehat{r}) = \mathbb{E}(Y - \widehat{r}(X))^2.$$

This risk simply measures the prediction error, i.e. the expected error we make in predicting Y when we use the function $\widehat{r}(X)$. This risk is minimized by the conditional expectation, i.e. we have the following theorem.

Theorem 27.1 *The risk R is minimized by*

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

Proof: Let $g(x)$ be any function of x. Then

$$\begin{aligned}
R(g) &= \mathbb{E}(Y - g(X))^2 = \mathbb{E}(Y - r(X) + r(X) - g(X))^2 \\
&= \mathbb{E}(Y - r(X))^2 + \mathbb{E}(r(X) - g(X))^2 + 2\mathbb{E}((Y - r(X))(r(X) - g(X))) \\
&\geq \mathbb{E}(Y - r(X))^2 + 2\mathbb{E}((Y - r(X))(r(X) - g(X))) \\
&= \mathbb{E}(Y - r(X))^2 + 2\mathbb{E}\mathbb{E}\left((Y - r(X))(r(X) - g(X)) \;\middle|\; X \right) \\
&= \mathbb{E}(Y - r(X))^2 + 2\mathbb{E}\Big((\mathbb{E}(Y|X) - r(X))(r(X) - g(X)) \Big) \\
&= \mathbb{E}(Y - r(X))^2 + 2\mathbb{E}\Big((r(X) - r(X))(r(X) - g(X)) \Big) \\
&= \mathbb{E}(Y - r(X))^2 = R(r).
\end{aligned}$$

■

For what we will do in class it will not matter which definition of risk we use so we will use the one from the previous section for the rest of the lecture.

27.2 Kernel Regression

One of the most basic ways of doing non-parametric regression is called kernel regression. We will analyze kernel regression when we only have one covariate. The general case is not very different.

Here the estimator is defined as:

$$\widehat{r}(x) = \sum_{i=1}^{n} w_i(x) Y_i,$$

where the weights assign more importance to points near x. This is called a kernel regressor when the weights are chosen according to a kernel, i.e. we have weights:

$$w_i(x) = \frac{K\left(\frac{x-X_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right)},$$

where h controls the amount of smoothing. It is called the bandwidth. As a typical common example we have the Gaussian kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2).$$

27.3 Analysis of Kernel Regression

In this section we will provide a simple analysis of kernel regression. We will do so under various (strong) assumptions. We make these assumptions so that we can prove our main result in this lecture.

We assume that

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

where:

1. **Assumptions about the design:** We will assume that x_i is one-dimensional, and equally spaced on $[0, 1]$. In general, we do not need that the design is equally spaced but intutively we do need to ensure that we see some points in the vicinity of each point where r is non-zero.

2. **Assumptions about the regression function:** We will assume that the function $r(x) = \mathbb{E}[Y|X = x]$ is L-Lipschitz, i.e. that there is some constant L such that:

$$\left|\frac{d}{dx} r(x)\right| \leq L.$$

3. **Assumptions about the noise:** We will assume that the noise is i.i.d and that $\mathbb{E}[\epsilon_i] = 0, \mathrm{Var}[\epsilon_i] = \sigma^2$.

4. **Assumptions about the kernel:** We will assume that the kernel is the *spherical kernel*:

$$K(x) = \mathbb{I}(-1 \leq x \leq 1).$$

Again the result we prove holds for a much broader class of kernels but we make this assumption to simplify the proof.

The main result: Suppose that the bandwidth $h \geq 1/n$, then the point-wise bias of the kernel regression estimator:

$$b(x) = \mathbb{E}[\widehat{r}(x)] - r(x) \leq Lh,$$

and the point-wise variance:

$$\mathrm{Var}(\widehat{r}(x)) = \mathbb{E}(\widehat{r}(x) - \mathbb{E}(\widehat{r}(x)))^2 \leq \frac{\sigma^2}{nh}.$$

Before, we prove the theorem let us observe that we can calculate the integrated risk:

$$R(\widehat{r}, r) = \int b^2(x)dx + \int v(x)dx \leq L^2h^2 + \frac{\sigma^2}{nh},$$

a natural strategy would be to choose the bandwidth to minimize this expression. Choosing

$$h = \left(\frac{\sigma^2}{2nL^2}\right)^{1/3},$$

we obtain that

$$R(\widehat{r}, r) \leq \frac{2L^{2/3}\sigma^{4/3}}{n^{2/3}} = 2\left(\frac{L\sigma^2}{n}\right)^{2/3}.$$

This result already reveals something fundamental about non-parametrics. We see that with an optimally chosen bandwidth, the MISE decreases to 0 at rate $n^{-2/3}$. By comparison, most parametric estimators converge at rate n^{-1}. The slower rate of convergence is the price we pay for being nonparametric. The formula for the optimal bandwidth is not useful in practice since it depends on the unknown Lipschitz constant L, and the typically unknown noise variance.

With all of these preliminaries in place let us prove the result:

Bounding the bias: The bias is given by:

$$\begin{aligned} b(x) = |\mathbb{E}\widehat{r}(x) - r(x)| &= \left|\mathbb{E}\left[\sum_{i=1}^n (w_i(X_i)(Y_i - r(x)))\right]\right| \\ &= \left|\sum_{i=1}^n (w_i(X_i)(r(X_i) - r(x)))\right| \leq \sum_{i=1}^n w(X_i)|r(X_i) - r(x)| \\ &\leq Lh\sum_{i=1}^n w(X_i) = Lh. \end{aligned}$$

At a high-level, the kernel regressor aggregates Y values from nearby points, and the bias just captures how much the true function can change over this region.

Bounding the variance: We note first that each weight is upper bounded as:

$$w_i(X_i) \leq \frac{1}{nh}.$$

Returning to the variance we obtain:

$$\begin{aligned} v(x) = \mathbb{E}(\widehat{r}(x) - \mathbb{E}(\widehat{r}(x)))^2 &= \mathbb{E}\left(\sum_{i=1}^n (w_i(X_i)(Y_i - r(X_i)))\right)^2 \\ &= \mathbb{E}\left(\sum_{i=1}^n \epsilon_i w_i(X_i)\right)^2 \\ &= \sum_{i=1}^n w_i(X_i)^2 \mathbb{E}(\epsilon_i^2) = \sigma^2 \sum_{i=1}^n w_i(X_i)^2 \\ &\leq \sigma^2 \max_{i=1}^n w_i(X_i) \sum_{i=1}^n w_i(X_i) \leq \frac{\sigma^2}{nh}. \end{aligned}$$

This completes our analysis.

27.4 The general case

We will be quite loose here just to focus on the main ideas. More generally, suppose that the β^{th} derivative of $r(x)$ is bounded, and we are in d-dimensions. In this case the bias will be roughly:

$$b^2(x) \approx h^{2\beta},$$

and the variance:

$$v(x) \approx \frac{1}{nh^d},$$

and balancing these will lead to the rate of convergence:

$$R(\widehat{r}, r) \approx n^{-2\beta/(2\beta+d)}.$$

This reveals another crucial feature of non-parametrics. In linear regression, the rate of convergence is typically something like:

$$R(\widehat{\beta}, \beta) \approx \frac{d}{n}.$$

In both cases, the situation gets worse as d increases, however in non-parametrics the situation gets *exponentially* worse. This is often colloquially referred to as the *curse of dimensionality.*

27.5 RKHS regression - optional

In machine learning, a somewhat different form of non-parametric regression is popular. This form is also known as kernel regression but we will refer to it as Reproducing Kernel Hilbert Space (RKHS) regression. We will not cover this in much detail (I assume you will see this or something quite similar in an ML class but let me know if not).

The idea roughly is the following. First we need the notion of a positive semi-definite kernel:

A symmetric bivariate function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is positive semidefinite (PSD) if for all integers $n \geq 1$ and elements $\{x_i\}_{i=1}^n$ where each $x_i \in \mathcal{X}$, the $n \times n$ matrix K with elements $K_{ij} := K(x_i, x_j)$ is positive semidefinite.

Here are a few standard examples:

1. Linear kernel: When $\mathcal{X} = \mathbb{R}^d$ then $K(x_i, x_j) = \langle x_i, x_j \rangle = \sum_{u=1}^d x_{iu} x_{ju}$, is the linear kernel and is PSD.
2. Polynomial kernel: Again when $\mathcal{X} = \mathbb{R}^d$ then $K(x_i, x_j) = (\langle x_i, x_j \rangle)^m$, is the homogenous polynomial kernel of degree $m \geq 2$. This kernel is also PSD. The inhomogenous polynomial kernel $K(x_i, x_j) = (1 + \langle x_i, x_j \rangle)^m$ is also PSD.
3. Gaussian kernel: Perhaps the most popular kernel in machine learning is the Gaussian kernel. Here we take $K(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2/(2\sigma^2))$.

The overly simplified description of RKHS regression is the following: given a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ we are going to attempt to estimate the function r by a function r_α which we will assume has the form:

$$r_\alpha(x) = \sum_{i=1}^n \alpha_i K(x_i, x),$$

where we need to estimate the α_is. In order to do this we will minimize a least-squares type objective:

$$\widehat{\alpha} = \arg\min_\alpha \frac{1}{2} \sum_{i=1}^n (y_i - r_\alpha(x_i))^2 + \lambda \text{Pen}(r_\alpha).$$

The penalty we will use is something called an RKHS norm penalization and it takes the form:

$$\text{Pen}(r_\alpha) = \alpha^T K \alpha,$$

where K is the gram matrix, i.e. $K_{ij} = K(x_i, x_j)$. This penalty encourages the function to be smooth but this is not easy to see without going into more detail. Observe that we can

write:

$$\begin{bmatrix} r_\alpha(x_1) \\ r_\alpha(x_2) \\ \vdots \\ r_\alpha(x_n) \end{bmatrix} = K\alpha,$$

so the RKHS regression objective simplifies to:

$$\widehat{\alpha} = \arg\min_\alpha \frac{1}{2}\|y - K\alpha\|_2^2 + \lambda \alpha^T K \alpha,$$

which we can solve in closed form (just by taking derivatives and setting to zero) as:

$$\widehat{\alpha} = (K + \lambda I)^{-1} y.$$

Our estimated regression function then takes the form:

$$r_{\widehat{\alpha}}(x) = \sum_{i=1}^n \widehat{\alpha}_i K(x_i, x).$$

Superficially there are similarities between RKHS regression and kernel regression. They both produce a function whose value at a given point is a weighted combination of the y values at other points. In kernel regression the weights are easy to interpret, while in RKHS regression the weights are the solution to a least squares problem and are not directly interpretable.

From a practical standpoint, RKHS regression typically has two tuning parameters: the penalty parameter λ and usually some RKHS parameter (for instance the RKHS kernel bandwidth for a Gaussian kernel).

There are two types of problems one could ponder: (1) we want to fit a function that is Lipschitz or Holder smooth (as we analyzed in the first half): in this case, it is perhaps natural to use kernel regression and somewhat more artificial to use RKHS regression (2) we want to fit a function in a particular RKHS, in this case it is perhaps more natural to use RKHS regression.

From a theoretical standpoint, RKHS regression is usually analyzed using variants of the Rademacher complexity results we saw earlier in the course, i.e. they are not directly analyzed in terms of the bias and variance (this is because the RKHS regression procedure is naturally viewed as ERM over an RKHS). This means that the rates of convergence are typically specified in terms of properties of the RKHS and the data-generating distribution, i.e. a typical measure of complexity of an RKHS is the decay-rate of eigenvalues of the kernel gram matrix. This is quite unlike kernel regression where the function class is something simple (Lipschitz functions), and the measure of complexity is just the smoothness of the function.

RKHS regression is not the only alternative to kernel regression. Often you will see methods like k-NN regression (where you predict at a point by averaging the y values of the k-closest points), local polynomial regression (where you chop up the domain and fit (low-degree) polynomials in each piece of the domain) and orthogonal series estimators or projection estimators (where you expand the regression function in a orthogonal basis – say of sine/cosine type functions – and then estimate the coefficients in this basis).