STAT 460/560 Class 15: Non-parametric regression: linear smoothers and basis functions

Ben Bloem-Reddy

Reading: Ch. 21, [Was04], Ch. 5.4-5.5, [Was06].

We'll dig a little deeper today into nonparametric regression.

1. Linear smoothers

At the end of the previous class, we introduced the Nadaraya–Watson kernel estimator (NWKE),

$$\hat{r}(x) = \sum_{i=1}^{n} w_i(x) Y_i \ , \quad \text{with} \quad w_i(x) = \frac{K_h(x, x_i)}{\sum_{j=1}^{n} K_h(x, x_j)} \ . \tag{15.1}$$

The NWKE is an example of a **linear smoother**, which is defined as any estimator $\hat{r}_n$ of r such that for each x, there is a vector $\ell(x) = (\ell_1(x), \dots, \ell_n(x))^\top$ with

$$\hat{r}_n(x) = \sum_{i=1}^{n} \ell_i(x) Y_i \ . \tag{15.2}$$

The name comes from the fact that $\hat{r}_n$ is a linear combination of the observed Y_i's. Note that observations x_i are not part of the definition, but typically they will be used to estimate ℓ.

Denote the vector of fitted values

$$\mathbf{r}_n = (\hat{r}_n(x_1), \dots, \hat{r}_n(x_n))^\top = LY \ , \tag{15.3}$$

where L is a $n \times n$ matrix with $L_{ij} = \ell_j(x_i)$, i.e., the entries of L are the smoothing functions ℓ_j evaluated at the x_i's. We can think of the ith row as the weights assigned to Y_j to calculate $\hat{r}_n(x_i)$. L is called the **smoothing matrix**, with **effective degrees of freedom** equal to $\nu_L = \mathrm{tr}(L)$.

In the case of the NWKE,

$$\ell_i(x) = w_i(x) = \frac{K_h(x, x_i)}{\sum_{j=1}^{n} K_h(x, x_j)} = \frac{K((x - x_i)/h)}{\sum_{j=1}^{n} K((x - x_j)/h)} \ . \tag{15.4}$$

As in the case of kernel density estimation, the bandwidth parameter can be selected by cross-validation (CV). In regression, the **leave-one-out CV** score is defined as

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{r}_{(-i)}(x_i))^2 \ , \tag{15.5}$$

where $\hat{r}_{(-i)}$ denotes the estimated regression function when excluding observation (x_i, Y_i). In the case of linear smoothers that satisfy $\sum_{i=1}^{n} \ell_i(x) = 1$, it's straightforward to see that

$$\hat{r}_{(-i)}(x) = \sum_{j=1}^{n} Y_j \ell_{j,(-i)}(x) \ , \quad \text{with} \quad \ell_{j,(-i)}(x) = \begin{cases} 0 & j = i \\ \frac{\ell_j(x)}{\sum_{j' \neq i} \ell_{j'}(x)} & j \neq i \ . \end{cases} \tag{15.6}$$

That is, we can just renormalize the weights. Using this, one can show that

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{r}_n(x_i)}{1-\ell_i(x_i)}\right)^2 , \tag{15.7}$$

which makes it easy to compute.

Activity 15.1. Show that for linear smoothers satisfying $\sum_{i=1}^n \ell_i(x) = 1$, the leave-one-out CV can be written as (15.7).

The NWKE is also an example of **local regression**, which gives higher weight $\ell(x)$ to points near x. See Ch. 5.4 of [Was06] for more examples, including local polynomial regression, of which the NWKE is the zeroth order version.

2. Regularization and splines

We will see later in the case of ridge regression that we can add a penalization/regularization term to our loss function in order to bias our estimators to have certain properties. We can do something similar in nonparametric regression; here, however, we have to regularize the functions themselves, rather than the parameters. In particular, we might consider a **roughness penalty** (or complexity penalty) $J(r)$, so that we aim to minimize the penalized squared error loss

$$M(\lambda) = \sum_{i=1}^{n}(Y_i - \hat{r}_n(x_i))^2 + \lambda J(r) . \tag{15.8}$$

A common penalty is

$$J(r) = \int (r''(x))^2 \, dx . \tag{15.9}$$

In the limit $\lambda \to \infty$, this will force the second derivative of r to be zero, and we get linear regression. In the other extreme, $\lambda \to 0$ essentially lets $\hat{r}$ interpolate the data (assuming the function class we're using is flexible enough to do so).

A common approach to nonparametric (regularized) regression is to specify a set of **basis functions**, $B_1(x), \ldots, B_p(x)$, and model the regression function as

$$r(x) = \sum_{j=1}^{p}\beta_j B_j(x) . \tag{15.10}$$

Conceptually and computationally, these behave a lot like linear regression models, except that instead of working in a $\mathbb{R}$-valued vector space, formally we're working in a function-valued vector space. We won't worry about that here because practically speaking, things don't end up looking very different.

A popular set of functions for nonparametric regression (popularity justified below) is **cubic splines**. A spline basis is defined on a set of **knots**, which are points in an interval $[a, b]$ with $a \le \xi_1 < \xi_2 < \cdots \xi_k \le b$. A cubic spline is a function r that is cubic polynomial over each interval (ξ_j, ξ_{j+1}), and such that r has continuous first and second derivatives at each of the knots. (This can be generalized to Mth order polynomials.) A **natural spline** is linear beyond the boundary points a, b.

It turns out that the function $\hat{r}_n(x)$ that minimizes $M(\lambda)$ in (15.8) with (15.9) is a natural cubic spline with knots at the data points. This is called a **smoothing spline**.

We just need to construct a basis for them. There are many ways to do so; a computationally efficient basis is the **B-spline** basis. Their definition is somewhat complicated (see Ch. 5.5, [Was06]) but they are simple functions. See Fig. 1. These are smooth bumps that have compact support; which means that $B_j(x)B_{j'}(x)$ is non-zero only on a subset (often small) of (a, b). In practice, this allows the computation required to fit the model to take advantage of sparse/structured matrix computation.

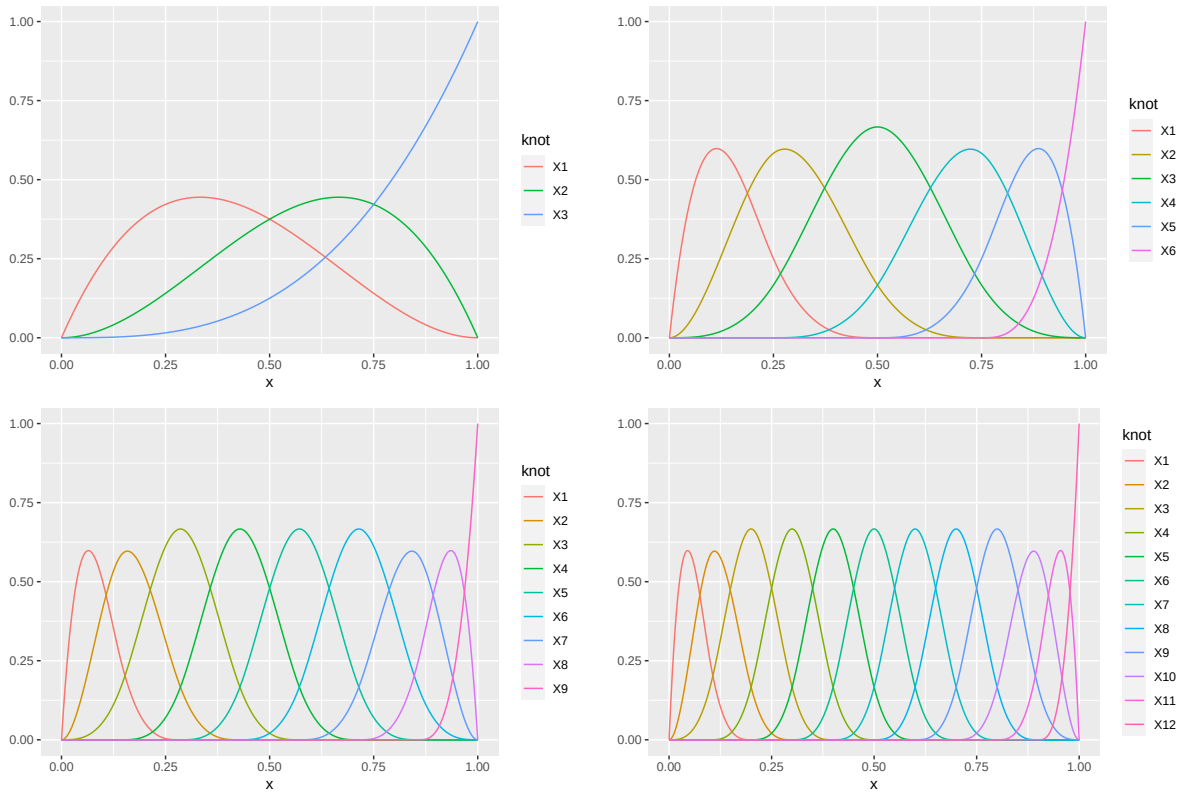

Figure 1: B-spline basis functions on $[0, 1]$ with (clockwise from upper left) $p \in \{3, 6, 9, 12\}$ equally spaced knots.

With this basis, we can write

$$\hat{r}_n(x) = \sum_{j=1}^{p} \hat{\beta}_j B_j(x) \ , \tag{15.11}$$

with $p = n + 4$, and find $\hat{\beta}$ by minimizing

$$\hat{\beta} = \arg\min_{\beta} (Y - B\beta)^\top (Y - B\beta) + \lambda \beta^\top \Omega \beta \ . \tag{15.12}$$

Here, $B_{ij} = B_j(x_i)$ and $\Omega_{jk} = \int B_i''(x) B_j''(x)\ dx$.

Activity 15.2. Show that

$$\hat{\beta} = (B^\top B + \lambda \Omega)^{-1} B^\top Y \ . \tag{15.13}$$

Does this look familiar?

This is a linear smoother. What is the smoothing matrix L that satisfies $(\hat{r}_n(x_1), \ldots, \hat{r}_n(x_n))^\top = LY$?

Solution: We find $\hat{\beta}$ by minimizing (15.12)l Since it is a quadratic function of β, we can take the gradient with respect to β, set it equal to zero, and solve to find the expression for $\hat{\beta}$. If you have ever encountered ridge regression, this is essentially the same form.

References

[Was04] L. Wasserman. *All of Statistics*. Springer New York, 2004.

[Was06] L. Wasserman. *All of Nonparametric Statistics*. Springer New York, 2006.