Exercises 2: Some frequentist basics

Kernel density estimation

Linear smoothing

Linear smoothing: one predictor

Let's consider a problem with one predictor and one response, but a potentially more general regression function: $y_i = f(x_i) + \epsilon_i$.

(A) Suppose we want to estimate the value of the regression function $y^\star$ at some new point $x^\star$, denoted $\hat{f}(x^\star)$. Assume for the moment that $f(x)$ is linear, and that y and x have already had their means subtracted, in which case $y_i = \beta x_i + \epsilon_i$.

Stare at your answer from the first problem. Show that for the one-predictor case, your prediction for $y^\star$ may be expressed as a *linear smoother* of the following form:

$$\hat{f}(x^\star) = \sum_{i=1}^{n} w(x_i, x^\star) y_i \,.$$

Which of the following better describes your $w(x_i, x^\star)$? (1) The weight of a previous data point depends on how close x_i is to the sample mean $\bar{x}$. (2) The weight of a previous data point depends up how close x_i is to the value $x^\star$ where we want to predict.[1]

[1] In what situations will this feature be good, and in what will it be bad?

(B) Consider two alternate forms of the weight function $w(x_i, x^\star)$:

$$w_K(x_i, x^\star) = \begin{cases} 1/K, & x_i \text{ one of the } K \text{ closest sample points to } x^\star \\ 0, & \text{otherwise.} \end{cases}$$

$$w_B(x_i, x^\star) = \frac{e^{-(x^\star - x_i)^2/(2B^2)}}{\sum_{j=1}^{n} e^{-(x^\star - x_j)^2/(2B^2)}} \,.$$

Before turning to any software: inspect these functions and write down your intuition for how the resulting smoothers will behave for different values of K and B, compared to the smoother from above. If you knew that the regression function changed slowly as a function of x, what values would you choose, qualitatively speaking? (Also: why include the denominator in S_B?)

(C) Now write your own functions that will fit these smoothers for an arbitrary set of input vectors x and y, and arbitrary choices of (integer) K or (positive real) B.[2] Set up an R script[3] that will simulate noisy data from some nonlinear function $y = f(x) + \text{error}$ of your choice; subtract the sample means from the simulated x and y, and fit the smoother for particular choices of K and B. Plot the prediction functions over a range of values of K and B. Choose a range large enough to yield a correspondingly large range in the qualitative behavior of the prediction functions.

[2] Best of all is to wrap these functions up in another function (perhaps called "smoother") that will let you pass an argument for the kind of smoothing you want done.

[3] Or your favorite language

Choosing the bandwidth parameter

Again, we're in the one-predictor domain where $y_i = f(x_i) + \epsilon_i$. Left unanswered in the previous problem about linear smoothing was the question: how does one choose the tuning constants K or B? Assume for now that the goal is to predict well, not necessarily to recover the truth. (These are related but distinct goals.)

(A) Let $y^\star$ be the (unknown) response at some future point $x^\star$. Let $\hat{f}$ be an estimator of the true function f. Clearly $y^\star = f(x^\star) + \epsilon^\star$. Prove the following decomposition for the expected squared error in prediction:

$$E\left\{\left[y^\star - \hat{f}(x^\star)\right]^2\right\} = \text{var}(\epsilon^\star) + \left[f(x^\star) - E\left\{\hat{f}(x^\star)\right\}\right]^2 + \text{var}\{\hat{f}(x)\}\,.$$

All moments are taken under the sampling distribution for the data, given the true function f. Briefly interpret each of the three summands on the right-hand side. You should be able to explain why this is referred to as the "bias–variance tradeoff."[4]

[4] Why a tradeoff?

(B) "It would be great," you think to yourself, "if I had a fresh data set where I could test my predictions arising from the first, with particular choices of B." Darn right! Let's focus on w_B, with the issues for w_K being substantively the same. Write a function or script that will: (1) accept a "training" data set and a "testing" data set as inputs; (2) fit the w_B smoother to the training data for a range of B values; and (3) return the realized prediction error on the testing data for each value of B. We'll keep score using the average squared error in prediction:

$$L_n(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n^\star}\{y_i^\star - \hat{f}(x_i^\star)\}^2\,,$$

where $(y_i^\star, x_i^\star)$ are the points in the test set, and $\hat{f}$ is some estimate of the regression function arising from the training set.

(C) Imagine a conceptual two-by-two table for the unknown, true state of affairs. The rows of the table are "wiggly function" and "smooth function," and the columns are "highly noisy observations" and "not so noisy." Simulate two independent data sets (one training and one testing data set) for each of the four cells of this table. That is, pick two functions on some closed interval, one smooth and one wiggly. Pick two values of the scale for ϵ_i, one large and one small (relative to f). For each function/scale pair, simulate two data

sets of 250 points each. Apply your method from above. Does your train/test function lead to reasonable choices of B for each cell of the table?

(D) Suppose that the present and future x's are chosen at random from some density $h(x)$, and that the true function is twice differentiable. Let $\sigma^2 = \text{var}(\epsilon)$. Prove that

$$E\left\{\left[y - \hat{f}(x)\right]^2\right\} = \sigma^2 + h^4(\tau^2)^2\left\{\frac{1}{2}f''(x) + \frac{f'(x)\,h'(x)}{h(x)}\right\}^2 + \frac{\sigma^2\eta}{nBh(x)} + o(B^4) + o\left(\frac{1}{nB}\right)$$

for some τ^2 and η that you must calculate.[5]

[5] Remember O and o notation.

In case the form of the right-hand side is insufficiently suggestive: you should use a second-order Taylor expansion. This one is a bit technical. To get you started, here's the flavor of the argument using a first-order Taylor expansion. Expand $f(x_i)$ around the point x where you want to predict:

$$y_i = f(x) + (x - x_i)f'(x) + \epsilon_i\,.$$

Now appeal to the fact that $\hat{f}$ is a linear smoother:

$$\begin{aligned}\hat{f}(x) &= \sum_{i=1}^{n} w_i y_i \\ &= \sum_{i=1}^{n} w_i\left\{f(x) + (x - x_i)f'(x) + \epsilon_i\right\}\,,\end{aligned}$$

abbreviating the weights as w_i. Remember that the residuals are mean zero and uncorrelated with anything. Use this to calculate the individual terms in the decomposition you proved from above.

Once you prove the result, consider some auxiliary questions.

i. Explain, intuitively, why the $h'(x)$ term enters the picture.

ii. Use this expression to derive the optimal asymptotic order (as $n \to \infty$) for B in terms of the sample size n.

iii. Explain what this choice of B implies for the asymptotic behavior of the excess error in prediction,

$$E\left\{\left[y - \hat{f}(x)\right]^2\right\} - \sigma^2\,.$$

It converges to something. What, and how fast?

iv. What barriers exist to the practical usage of this result? Above you could ignore pesky constants that didn't grow or shrink with n. Now think about what's involved in them.