STATISTICS 452/652: Statistical Learning and Prediction

October 14, 2020

Lecture 10: Other Splines and Smoothers

(Reading: ISLR 7.5–7.6)

1 Goals of lecture

- We have seen how basis functions can be used to create smooth curves called cubic (regression basis) splines
- There are other approaches to creating smooth curves that adapt to the shapes suggested by data
- We will briefly explore two of these
- Once again, these work best with one explanatory variable, and are mostly used for visual appeal rather than for prediction.

2 Smoothing Splines (Regularization and Shrinkage)

- Another way to make a smooth, adaptable function is to use *shrinkage*
- Let $f(X)$ go wherever the data are
  - Minimize $\sum_{i=1}^{n}(y_i - f(x_i))^2$
  - This will surely overfit by chasing errors
- To prevent overfitting, use regularization/penalization to shrink the function back toward the mean

  - Similar idea to LASSO and Ridge

$$Q_{RIDGE,\lambda} = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

$$Q_{LASSO,\lambda} = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

- Specifically penalize the “wiggliness” of the function
  - Penalty accumulates as the slope is allowed to change [1]

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda[\text{Smoothness Penalty}]$$

  - Penalty coefficient is chosen by user to control relative contributions of minimizing sMSE vs. keeping penalty low.

- By “mathemagic,” it turns out that this open-ended formulation is equivalent to a natural spline with a knot at each observation
  - WAY too many parameters to estimate
  - Penalty applies shrinkage (like LASSO and Ridge parameter estimates)

- Results in a smooth trend with smoothness controlled by user.
  - Pretty popular.
  - Usually approximated by using a grid of many knots if n is large.
  - Also, math relationships allow penalty coefficient to be related to DF,
    - Can specify DF to control penalty coefficient indirectly and get a fit with approximately those DF

Example: Smoothing Splines on Covid-19 cases in BC (L10 - Smoothers.R) The `smooth.spline()` function in the main `stats` package does smoothing splines. I have specified 6, 9, and 12 DF, which the function will use as targets for estimating the shrinkage penalty coefficient. See Figure 1.

All of these curves seem to me to be somewhat less prone to extremes and high curvature than the other methods’ curves. In particular, the 12DF curve seems like it is hitting most of the interesting features pretty well.

You can also use CV or GCV to estimate the “optimal” penalty or DF. A word of warning, though: you may not like the results. GCV, in particular, has given me unsatisfying results in several examples. GCV is an approximation to leave-one-out CV, and leaving one out at a time often makes the function think it’s doing great bu following a lot of the short-term patterns, instead of smoothing them over. Regular CV tends to result in much smoother functions. It just depends on whether you want to follow a lot of short cycles or cover them over. See Figure 2.

---

[1] Amounts to penalizing the second derivative, integrates across the range of X

Figure 1: Smoothing splines fit to Covid-19 data

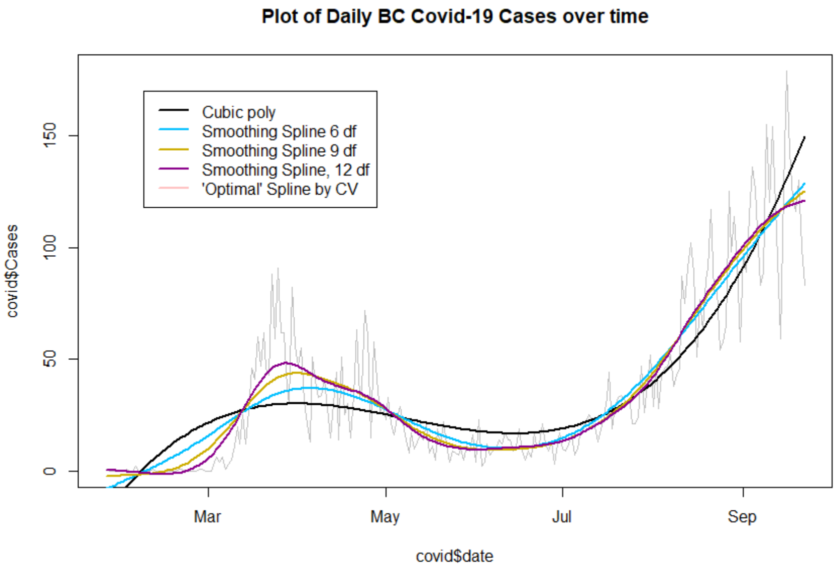
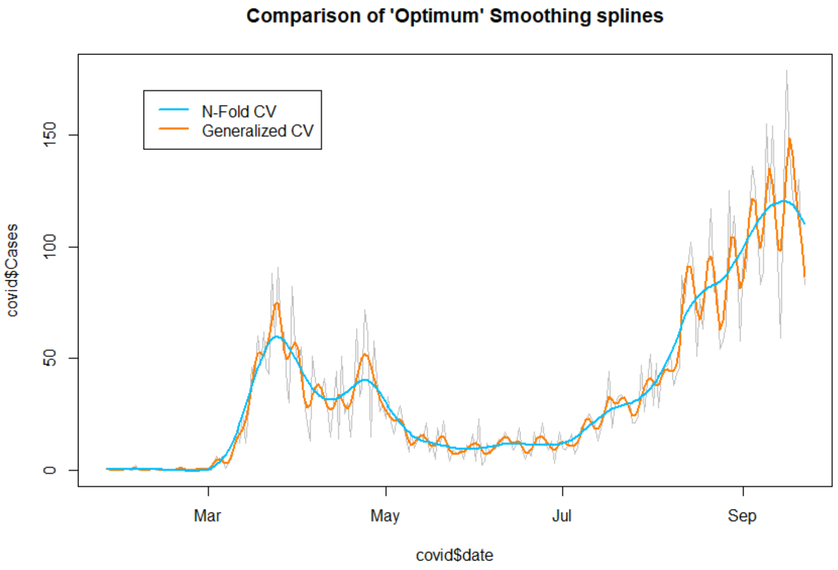


Figure 2: Smoothing splines fit to Covid-19 data



3 Local Polynomial Regression (LOESS)

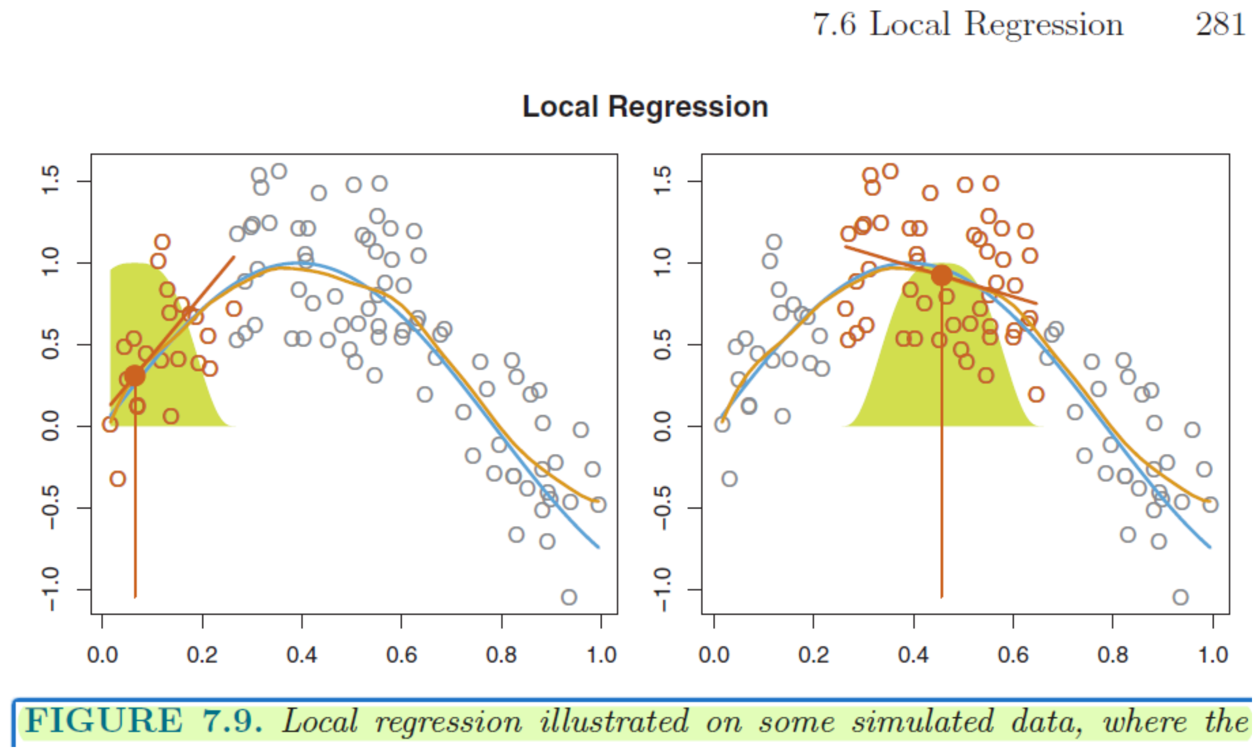
Alternatively, can fit *very* locally (one point at a time!) using a weighting scheme to "count" some points more toward the fit than others.

1. Predict Y at any x_0 using a weighted function of points in the neighbourhood of x_0. See Figure 3.
   (a) Definition of "neighbourhood" is flexible
      i. In R, main function uses "SPAN", which is the fraction of observations that are allowed to contribute to the fit at x_0
         A. User chooses span fraction s, which selects the $k = ns$ nearest points to focus on
      ii. Bias-variance tradeoff
         A. More points = lower variance, higher bias
   (b) Definition of "weighted" is flexible
      i. "Kernel" function applies differential weights depending on distance from x_0 (Figure 3 shows one common kernel shape).
         A. Some kernels put 0 weight on points outside the span
         B. Most weights are higher for x_i points closer to x_0
   (c) Definition of "function" flexible
      i. Usually predict Y at x_0 locally using quadratic; best for local minima and maxima (Figure 3 shows a linear regression function used locally on weighted data.)
2. Result is a "locally weighted scatterplot smoother" (LOWESS or LOESS)
   (a) Predictions typically made on a find grid of target points within range of X.
   (b) Parameters of local regression estimated using weighted least squares

3.1 Notes

1. Usually, the function and the kernel are chosen in advance and the span s is left as a tuning parameter.
   (a) Smoothing is a visual tool. *Frequently done by look of results.*
   (b) Formulas exist for N-fold CV and GCV.
   (c) Also can be chosen based on approximate relationship between span, weights, and DF
2. Boundary issues (see Fig 3)
   (a) Fewer observations near end

Figure 3: Locally weighted regression smoothing using only a fraction of the points with weighted contribution to the fit. Blue curve is truth. Solid orange point is x_0, where prediction is being done. Orange circles are data within span. Green shape is kernel for weighing points. Orange line is the prediction function used locally. Lighter orange curve is complete prediction function.

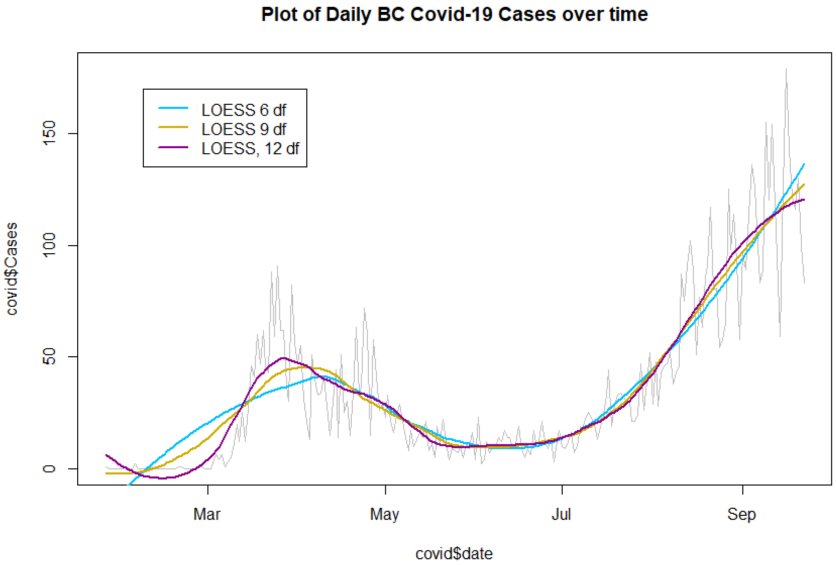


7.6 Local Regression 281

Local Regression

FIGURE 7.9. *Local regression illustrated on some simulated data, where the*

Figure 4: LOESS smoothers fit to Covid-19 data



(b) Only part of kernel in use

(c) Hard to pick up local changes

(d) Quadratic function helps bias but beware variability, especially near ends.

Example: LOESS on Covid-19 cases in BC (L10 - Smoothers.R) The `loess()` function in the main `stats` package does local polynomial LOESS fits. The default is a quadratic polynomial (`degree=2`), and `span=0.75`. You can mess with `span` to control the bias-variance tradeoff, or do this indirectly through approximate DF using `enp.target=` . I have specified 6, 9, and 12 DF, which the function will use as targets for estimating the span. See Figure 4.

All of these curves seem broadly similar to other methods' curves. The 12DF curve seems a little less smooth than others, which can be a property of methods that use sliding windows of data for their calculations.

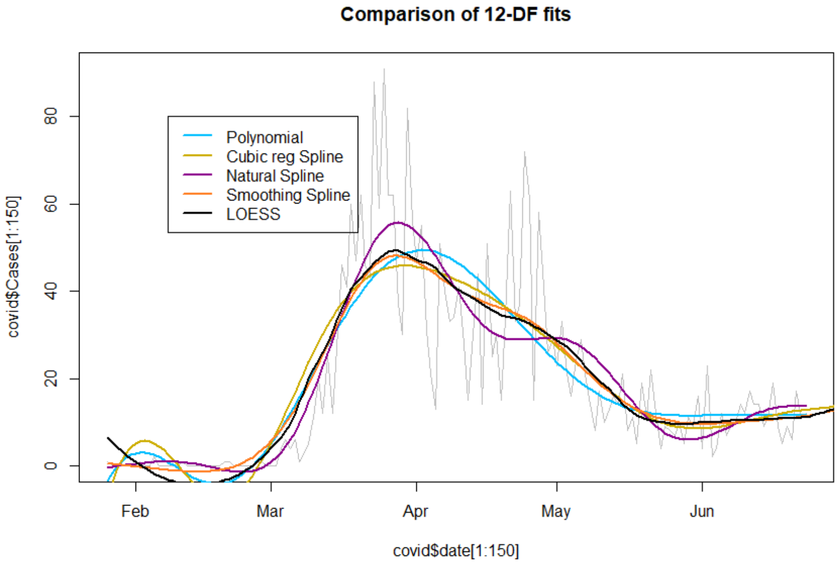
As a final comparison, I show the 12DF fits from each method we have seen: polynomials, basis splines, natural splines, smoothing splines, and LOESS. Since all methods give similar performance in modeling the "second wave"—except for a downturn in some methods at the very end—I focus on first 2/3 of the time line. The comparison is in Figure 5. You can draw your own conclusions about these fits. For my money, I like the natural and smoothing splines' flatness until March, as well as their recognition of the slight second hump (although the natural spline seems to mis-place it to the right of where it should be).

---

4 What to take away from this

- There are several different methods for smoothing mean trends in data

Figure 5: Comparison of 12DFfits to Covid-19 data



- No one method is superior to all others in all cases
  - I tend to prefer natural and smoothing splines
- These methods break down in higher dimensions, so are mainly used as visual tools
- Suggestion: Add a smoother to residual plots from a regression.
  - It is sometimes hard to detect a trend in these plots by eye, so a smoother is great here.
  - Residuals from a good model fit should have a smooth curve that undulates around 0 over the full range of the residual plot vs. X or $\hat{Y}$.
  - Strong movement away from 0 is an indication of a lack of fit of the regression.

5 Exercises

Application

Refer to the Air Quality data described previously, and the analyses we have done with `Ozone` as the response variable, and the five explanatory variables (including the two engineered features).

1. Use smoothing splines to model the relationship between `Ozone` and `Temp`:

   (a) On one graph, plot the data along with fits of smoothing splines with 5, 7, 9, and 20 DF.

      i. **Present the plot. Be sure to add a legend and use different colours for the different functions**

      ii. If you had to choose one model, **which would it be? Why?**

   (b) Use cross-validation and generalized cross-validation to choose the optimal smoothing amount

      i. **How many DF does each method suggest to use?**

      ii. **Show the fits on one plot with the data.**

      iii. **Comment on the quality of each fit.**

2. Repeat part (a) from Exercise 1 using LOESS.

Note that, for these questions, I am less interested in seeing you achieve a perfect solution, and more interested in seeing you learn to appreciate the merits of different fits.