Smoothing Splines

Advanced Methods for Data Analysis (36-402/36-608)

Spring 2014

1 Splines, regression splines

1.1 Splines

- Smoothing splines, like kernel regression and k-nearest-neigbors regression, provide a flexible way of estimating the underlying regression function $r(x) = \mathbb{E}(Y|X = x)$. Though they can be defined for higher dimensions, we'll assume for simplicity throughout that $X \in \mathbb{R}$, i.e., there is only one predictor variable

- Before introducing smoothing splines, however, we first have to understand what a *spline* is. In words, a kth order spline is a piecewise polynomial function of degree k, that is continuous and has continuous derivatives of orders $1, \ldots k-1$, at its knot points

- Formally, a function $f : \mathbb{R} \to \mathbb{R}$ is a kth order spline with knot points at $t_1 < \ldots < t_m$, if

  - f is a polynomial of degree k on each of the intervals $(-\infty, t_1], [t_1, t_2], \ldots [t_m, \infty)$, and
  - $f^{(j)}$, the jth derivative of f, is continuous at $t_1, \ldots t_m$, for each $j = 0, 1, \ldots k-1$.

  Splines have some very special properties are have been a topic of interest among statisticians and mathematicians for a long time

- The most common case considered is $k = 3$, i.e., that of cubic splines. These are piecewise cubic functions that are continuous, and have continuous first, and second derivatives. Note that the continuity in all of their lower order derivatives makes splines very smooth. A bit of statistical folklore: it is said that a cubic spline is so smooth, that one cannot detect the locations of its knots by eye!

- How can we parametrize the set of a splines with knots at a given set of points $t_1, \ldots t_m$? The most natural way is to use the *truncated power basis*, $g_1, \ldots g_{m+k+1}$, defined as

$$g_1(x) = 1,\ g_2(x) = x,\ \ldots\ g_{k+1}(x) = x^k,$$
$$g_{k+1+j}(x) = (x - t_j)_+^k, \quad j = 1, \ldots m.$$

  Here x_+ denotes the positive part of x, i.e., $x_+ = \max\{x, 0\}$

- While these basis functions are natural, a much better computational choice, both for speed and numerical accuracy, is the *B-spline* basis. This was a major development in spline theory and is now pretty much the standard in software; we won't cover these, but it doesn't hurt to be aware of them

1.2 Regression splines

- So, what can you do with splines? Well, for one, we can perform regression on them! In other words, given samples (x_i, y_i), $i = 1, \ldots n$, we can consider estimating the regression function $r(x) = \mathbb{E}(Y|X = x)$ by fitting a kth order spline with knots at some prespecified locations $t_1, \ldots t_m$

- This means considering functions of the form $\sum_{j=1}^{m+k+1} \beta_j g_j$, where $\beta_1, \ldots \beta_{m+k+1}$ are coefficients and $g_1, \ldots g_{m+k+1}$, are the truncated power basis functions for kth order splines over the knots $t_1, \ldots t_m$

- The coefficients $\beta_1, \ldots \beta_{m+k+1}$ above are just estimated by least squares. That is, we first find $\hat{\beta}_1, \ldots \hat{\beta}_{m+k+1}$ to minimize the criterion

$$\sum_{i=1}^{n} \Big(y_i - \sum_{j=1}^{m} \beta_j g_j(x_i) \Big)^2, \tag{1}$$

  and then define the *regression spline*

$$\hat{r}(x) = \sum_{j=1}^{m+k+1} \hat{\beta}_j g_j(x)$$

- The expression in (1) looks more familiar after a change in notation. Write $y = (y_1, \ldots y_n) \in \mathbb{R}^n$, and define the basis matrix $G \in \mathbb{R}^{n \times (m+k+1)}$ by

$$G_{ij} = g_j(x_i), \quad i = 1, \ldots n, \; j = 1, \ldots m + k + 1.$$

  (I.e., the jth column G gives the evaluations of g_j over the points $x_1, \ldots x_n$.) Then we can rewrite the criterion in (1) as

$$\|y - G\beta\|_2^2. \tag{2}$$

  Of course, from what we know about linear regression, the optimal coefficients are

$$\hat{\beta} = (G^T G)^{-1} G^T y$$

- Regression splines are linear smoothers. To see this, denote $g(x) = (g_1(x), \ldots g_{m+k+1}(x))$, and then the regression spline estimate at x is

$$\hat{r}(x) = g(x)^T \hat{\beta} = g(x)^T (G^T G)^{-1} G^T y,$$

  a weighted combination of y_i, $i = 1, \ldots n$ (where the weights are given by the components of $G(G^T G)^{-1} g(x)$)

- Regression splines are a classic tool, and can work well provided we choose good knot points $t_1, \ldots t_m$; but in general choosing knots is a tricky business. This is the beauty behind smoothing splines—with them, we don't have to choose knots! Before discussing them, we have to take a little detour, though, to learn that they operate on a slightly different kind of piecewise polynomial

1.3 Natural splines

- One problem with regression splines is that the estimates tend to display erractic behavior, i.e., they have high variance, at the boundaries of the domain of $x_1, \ldots x_n$. This gets worse as the order k gets larger

- A way to remedy this problem is to force the piecewise polynomial function to have a lower degree to the left of the leftmost knot, and to the right of the rightmost knot—this is exactly what *natural splines* do. A natural spline of order k, with knots at $t_1 < \ldots < t_m$, is a piecewise polynomial function f such that

  - f is a polynomial of degree k on each of $[t_1, t_2], \ldots [t_{m-1}, t_m]$,
  - f is a polynomial of degree $(k-1)/2$ on $(-\infty, t_1]$ and $[t_m, \infty)$,
  - f is continuous and has continuous derivatives of orders $1, \ldots k-1$ at its knots $t_1, \ldots t_m$.

  It is implicit here that natural splines are only defined for odd orders k. The most common case: $k = 3$, i.e., cubic natural splines, which are linear beyond the boundaries

- Note that there is a variant of the truncated power basis for natural splines (and a variant of the B-spline basis for natural splines). This time, though, we only need m basis functions, $g_1, \ldots g_m$, to span the space of kth order natural splines with knots at $t_1, \ldots t_m$

1.4 Smoothing splines

- Smoothing splines are an interesting creature: these estimators perform (what we will come to know as) a regularized regression over the natural spline basis, placing knots at all points $x_1, \ldots x_n$. Smoothing splines circumvent the problem of knot selection (as they just use the inputs as knots), and simultaneously, they control for overfitting by shrinking the coefficients of the estimated function (in its basis expansion)

- We will focus on cubic smoothing splines (though they can be defined for any odd polynomial order). We consider functions of the form $\sum_{j=1}^n \beta_j g_j$, where $g_1, \ldots g_n$ are the truncated power basis functions for natural cubic splines with knots at $x_1, \ldots x_n$. Specifically, the coefficients are chosen to minimize

$$\|y - G\beta\|_2^2 + \lambda \beta^T \Omega \beta, \tag{3}$$

  where $G \in \mathbb{R}^{n \times n}$ is the basis matrix defined as

$$G_{ij} = g_j(x_i), \quad i, j = 1, \ldots n,$$

  and $\Omega \in \mathbb{R}^{n \times n}$ is the penalty matrix defined as

$$\Omega_{ij} = \int g_i''(t) g_j''(t)\, dt, \quad i, j = 1, \ldots n.$$

  Given the optimal coefficients $\hat{\beta}$ minimizing (3), the *smoothing spline* estimate at x is defined as

$$\hat{r}(x) = \sum_{j=1}^n \hat{\beta}_j g_j(x)$$

- The exact form of the penalty matrix Ω is actually not so important. What you should pay attention to is that there is an extra term $\lambda \beta^T \Omega \beta$ in (3) compared to the usual criterion (2) for regression splines; this is called a *regularization* term, and it has the effect of shrinking the components of the solution $\hat{\beta}$ towards zero. The parameter $\lambda \geq 0$ is a tuning parameter, often called the smoothing parameter, and the higher the value of λ, the more shrinkage

- Recall that each computed coefficient $\hat{\beta}_j$ corresponds to a particular basis function g_j. The term $\beta^T \Omega \beta$ in (3) imparts more shrinkage on the coefficients $\hat{\beta}_j$ that correspond to wigglier functions g_j. Hence, as we increase λ, we are shrinking away the wiggler basis functions

- Similar to least squares regression, it may (should) not surprise you that the coefficients $\hat{\beta}$ minimizing (3) are

$$\hat{\beta} = (G^T G + \lambda\Omega)^{-1} G^T y.$$

Again, then smoothing splines are seen to be linear smoothers. With $g(x) = (g(x_1), \ldots g(x_n))$, we have

$$\hat{r}(x) = g(x)^T \hat{\beta} = g(x)^T (G^T G + \lambda\Omega)^{-1} G^T y,$$

which is linear combination of the points y_i, $i = 1, \ldots n$

- What makes smoothing splines even more interesting is that they can be alternatively motivated directly from a functional minimization perspective. Consider minimizing, over all functions f,

$$\sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2 + \lambda \int \big(f''(x)\big)^2 \, dx. \tag{4}$$

This criterion trades off the least squares error of f over (x_i, y_i), $i = 1, \ldots n$, with a regularization term that grows large when the second derivative of f is wiggly. Remarkably, it so happens that there is a unique function minimizing this criterion, and further, this function is exactly the cubic smoothing spline estimator $\hat{r}$ defined above!

- A practical note: smoothing splines often deliver similar fits to those from kernel regression. However, they are in a sense simpler. Yes, both have a tuning parameter—the bandwidth h for kernel regression, and the smoothing parameter λ for smoothing splines—which we would typically need to choose by cross-validation. But that's it for smoothing splines, i.e., we don't require a choice of kernel. Also, it should be noted that smoothing splines are generally much more computationally efficient (this will be true when you use software that employs the B-spline basis, which is the case in R)