3.11.1 → Basis functions

→ overview → Recall the model equation for MLR:

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

→ In limiting ourselves to one covariate x, consider the case where the predictor x_j is some function of x denoted $b_j(x)$ for $j = 1, \ldots, p$. This means we want to transform the variable x using p different functions & using the resulting variables as predictors. The functions $b_1(\cdot), \ldots, b_p(\cdot)$ are called basis functions. As a consequence, the model equation is

$$Y = \beta_0 + \beta_1 b_1(x) + \cdots + \beta_p b_p(x) + \varepsilon$$

→ lets consider 3 types of regressions & their particular choice of basis functions:

→ Polynomial regression
→ Step function approach
→ piecewise polynomial regression

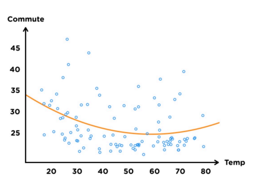
→ Polynomial regression

→ To model the response using x w/ d^{th} order/degree polynomial, the model equation is

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_d x^d + \varepsilon$$

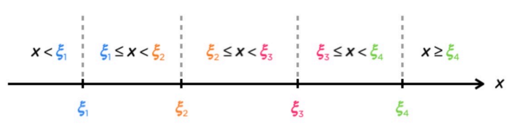
→ Then the basis functions are $b_j(x) = x^j$, $j = 1, \ldots, d$

→ Example → $\hat{y} = 38.2 - 0.46x + 0.004x^2$



→ Step function approach

→ Start by dividing the range of x into $k+1$ intervals or bins. Hence, there are k cutpoints or knots denoted as $\xi_1, \xi_2, \ldots, \xi_k$ throughout the range of x. To illustrate, the following diagram shows how four knots create 5 intervals



→ The basis functions here are dummy variables that indicate the interval where an x value is found. Specifically,

$$b_j(x) = \begin{cases} I(\xi_j \le x < \xi_{j+1}) & , j = 1, \ldots, k-1 \\ I(x \ge \xi_k) & , j = k \end{cases}$$

→ where $I(\cdot)$ is the indicator function. In effect, we have represented the covariate x as a factor w/ $k+1$ levels. These dummy variables are called step functions. Notice that the interval $x < \xi_1$ does not have a dummy variable, as it is the baseline category by default. As a result,

→ β_0 = mean response for $x \in (-\infty, \xi_1)$
→ $\beta_0 + \beta_j$ = mean response for $x \in [\xi_j, \xi_{j+1})$, where $j = 1, \ldots, k-1$
→ $\beta_0 + \beta_k$ = mean response for $x \in [\xi_k, \infty)$

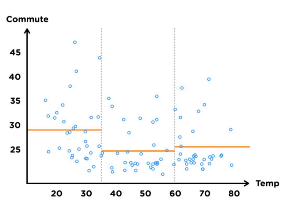
→ while we may arbitrarily choose knot values, it is common to use sample percentiles of x spread uniformly, depending on the # of knots desired. To illustrate, lets divide the range of Temp into three intervals, thus requiring two knots. Then, it makes sense for

ξ_1 = 33rd sample percentile of temp &
ξ_2 = 66th sample percentile of temp

→ using the empirical percentile approach, we get $\xi_1 = 35.1$ & $\xi_2 = 59.9$ & the following line:

$$y = 29.01 - 4.34 \cdot I(35.1 \le x < 59.9) - 3.49 \cdot I(x \ge 59.9)$$

$$= \begin{cases} 29.01 & x < 35.1 \\ 24.68 & 35.1 \le x < 59.87 \\ 25.53 & x \ge 59.87 \end{cases}$$



→ Piecewise polynomial regression

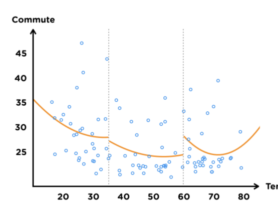
→ Essentially we have step functions as dummy variables & they interact w/ polynomial functions of degree d. The result is a piecewise polynomial regression, & it is more intuitive to write the model equation as:

$$Y = \begin{cases} \beta_{0,1} + \beta_{1,1} x + \cdots + \beta_{d,1} x^d + \varepsilon & , x < \xi_1 \\ \beta_{0,2} + \beta_{1,2} x + \cdots + \beta_{d,2} x^d + \varepsilon & , \xi_1 \le x < \xi_2 \\ \vdots & \\ \beta_{0,k+1} + \beta_{1,k+1} x + \cdots + \beta_{d,k+1} x^d + \varepsilon & , x \ge \xi_k \end{cases}$$

→ Intuitively, b/c interacting w/ dummy variables allows the baseline coefficients to be "updated" for the associated category, this setup creates a d^{th} degree polynomial per category. In this instance, the categories are the different intervals of x.

→ when $d = 0$, the regression simplifies to the step function approach; it is also called piecewise constant regression for this reason.

→ Example →



The fitted equation is

$$\begin{aligned} \hat{y} &= 43.6522 - 0.9359x + 0.0139x^2 \\ &\quad + 7.8452 \cdot I(35.132 \le x < 59.868) + 129.5699 \cdot I(x \ge 59.868) \\ &\quad - 0.0935x \cdot I(35.132 \le x < 59.868) - 3.2328x \cdot I(x \ge 59.868) \\ &\quad - 0.0043x^2 \cdot I(35.132 \le x < 59.868) + 0.0153x^2 \cdot I(x \ge 59.868) \\ &= \begin{cases} 43.6522 - 0.9359x + 0.0139x^2, & x < 35.132 \\ 51.4974 - 1.0294x + 0.0096x^2, & 35.132 \le x < 59.868 \\ 173.2221 - 4.1687x + 0.0292x^2, & x \ge 59.868 \end{cases} \end{aligned}$$

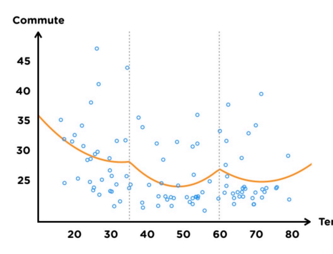
→ The discontinuous fitted curve is unappealing; it would be better for it to be a continuous function & relatively smooth

→ Degrees of freedom

→ As always, the df refers to the # of regression coefficients, $p+1$. This is used as an indicator of flexibility

3.11.2 → Regression splines

→ overview → if we enforce the three quadratics to be continuous at the knots, we obtain the following fit:



→ This is called a continuous piecewise quadratic regression. To do this, we impose two constraints (one per knot) so that the quadratics are unbroken at the knots. Imposing two constraints implies that we drop from $2+2+2 \cdot 2 = 8$ predictors to $8 - 2 = 6$ predictors.

→ Even so, the fitted curve still looks awkward near the knots. To smooth a fitted curve, we can impose two more constraints: the first derivative of the piecewise quadratic must also be continuous at each knot. If we did this, the regression would only consist of $6 - 2 = 4$ predictors, since another two constraints would be imposed

→ There is an example to show how we are the # of predictors.

→ model equation → $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$
→ now lets impose the constraint that $\beta_1 = \beta_2$. Then the model equation simplifies to

→ $Y = \beta_0 + \beta_1 x_1 + \beta_1 x_2 + \varepsilon$

$= \beta_0 + \beta_1 (x_1 + x_2) + \varepsilon$

→ The result is a SLR whose sole predictor is $x_1 + x_2$; the model features drop from 2 to 1 due to the constraint

→ In general, a constraint is an equation that allows one β to be completely expressed in terms of the other β's

→ Splines

→ A regression spline is a continuous piecewise polynomial regression that is smooth at the knots. In using a d^{th} degree polynomial, smoothness is attained by requiring the first $d-1$ derivatives of the curve to all be continuous at the k knots. This means, for example, a cubic spline has

→ continuity at the knots
→ first derivative continuity at the knots, &
→ second derivative continuity at the knots

→ Given that a cubic spline has 3 constraints per knot, its predictor count is

$$p = 3 + k + \underbrace{3k}_{\text{interactions}} - \underbrace{3k}_{\text{constraints}}$$

$$= 3 + k$$

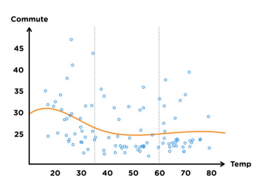
→ Since a cubic is often sufficiently flexible in a given interval, increasing the flexibility of a spline usually comes from increasing the # of knots rather than the polynomial order. In other words, the # of knots is a flexibility measure whose ideal value can be determined through cross-validation.

→ There are many ways to express the $3+k$ basis functions for a cubic spline. For this example, we use the following set of functions: x, x^2, x^3 & the remaining k are

$$b_4(x) = (x - \xi_1)^3_+, \; b_5(x) = (x - \xi_2)^3_+, \; \ldots, \; b_{3+k}(x) = (x - \xi_k)^3_+$$

→ known as truncated power basis functions. $+ \Rightarrow$ if $x > \xi$, then the basis function $= (x - \xi)^3$, otherwise $= 0$

→ Example

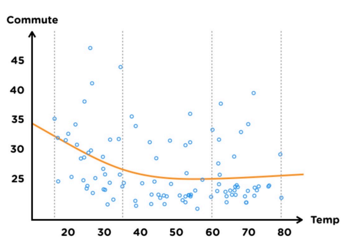


→ Natural splines

→ One disadvantage of fitting polynomials is that $se(\hat{y})$ is typically large near the lower & upper boundaries of the covariate. A method to minimize this for a cubic spline (or splines in general) is to assume that the curve becomes linear in the intervals $x < \xi_1$ & $x > \xi_k$. It is then implied that ξ_1 & ξ_k are boundary knots that are respectively near the lower & upper boundaries of x; the remaining knots are considered as interior knots.

→ This results in a natural cubic spline. Relative to a cubic spline, we want the first & the last of the $k+1$ cubics become lines. This further imposes 4 more constraints; we "drop the coefficients" of x^2 & x^3 in both boundary intervals. Therefore, a natural cubic spline operates with $3 + k - 4 = k - 1$ predictors. Although there are fewer basis functions now, they become quite complex & are not shown here.

→ Example →



→ Summary → with a d degree polynomial & k knots, the table below gives the # of predictors for the models we have discussed:

Model	Number of Predictors, p
Polynomial	d
Piecewise constant	k
Piecewise polynomial	$d + k + dk$
Continuous piecewise polynomial	$d + dk$
Cubic spline	$3 + k$
Natural cubic spline	$k - 1$

3.11.3 → Smoothing splines

→ overview → For the types of regressions we have considered thus far, the SSE is

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum \left[y_i - (\hat{\beta}_0 + \hat{\beta}_1 b_1(x_i) + \cdots + \hat{\beta}_p b_p(x_i)) \right]^2$$

→ As MLRs, realize $\hat{\beta}_0, \ldots, \hat{\beta}_p$ above are OLS estimates

→ lets take a step back & imagine a more generic form for $\hat{y}$. It is ultimately a function of x, so lets denote it as $g(x)$. The function $g(x)$ contains the estimates $\hat{\beta}_0, \ldots, \hat{\beta}_p$ but their relation to x is left unspecified. In other words, $\hat{\beta}_0, \ldots, \hat{\beta}_p$ are not assumed to be OLS estimates.

→ Desiring to minimize SSE is problematic at this stage.

$$\sum_{i=1}^{n} [y_i - g(x_i)]^2$$

→ So far, nothing about our setup prevents $g(x_i)$ from equalling y_i for all i, hence minimizing the expression is achieved by overfitting. Our goal is to establish a criterion that can prevent overfitting while having a smooth $g(x)$ that predicts the response well.

→ As overfitting means there is too much flexibility, we can incorporate a criterion that mimics a shrinkage method like ridge & lasso. Recall that these methods reduce flexibility by introducing a shrinkage penalty.

→ A smoothing spline seeks to minimize the expression

$$\underbrace{\sum_{i=1}^{n} [y_i - g(x_i)]^2}_{SSE} + \underbrace{\lambda \int_{-\infty}^{\infty} g''(t)^2 \, dt}_{\text{penalty}}$$

→ where λ is called a smoothing parameter or tuning parameter.

→ The shrinkage penalty involves $g''(\cdot)$ b/c the second derivative is an indicator of function roughness or wiggliness. The more wiggly $g(\cdot)$ is, the larger the integral becomes. As a result, minimizing this equation promotes a smooth $g(x)$. The level of smoothing is controlled by λ; as it increases, $g(x)$ must be smoother so the overall equation is minimized

→ The fitted curve $g(x)$ for a smoothing spline has the same form as the fitted natural cubic spline w/ knots at the n observed values of the covariate. To be precise, $g(x)$ is

→ linear in the intervals $x < \min(x_1, \ldots, x_n)$ & $x > \max(x_1, \ldots, x_n)$
→ piecewise cubic between the ordered $x_1, \ldots, x_n$ &
→ has continuity, first derivative continuity, & second degree continuity at $x_1, \ldots, x_n$

when all x_i are unique.

→ However, a smoothing spline is distinct from a natural cubic spline w/ n knots at the x values b/c the latter has coefficients estimated by OLS. Moreover, such a natural cubic spline perfectly predicts the response b/c it estimates $p + 1 = (n-1) + 1 = n$ regression coefficients. To understand how a smoothing spline is different, keep in mind that λ dictates the resulting $g(x)$ (& hence the values of $\hat{\beta}_0, \ldots, \hat{\beta}_p$ as well) by controlling its smoothness.

→ when $\lambda = 0$, there is no shrinkage penalty, so SSE is minimized to 0 due to perfect fit. This smoothing spline is the same as the OLS natural cubic spline (i.e. overfitted & very flexible).

→ As $\lambda \to \infty$, the entire expression is minimized when the integral is 0. This means $g(x)$ must be linear in x; this smoothing spline is the same as SLR (i.e. smooth & inflexible).

→ Effective degrees of freedom

→ In MLR, recall H is the hat matrix such that

→ $\hat{y} = Hy$
→ the leverage h_i is the i^{th} diagonal entry of H
→ $\sum_{i=1}^{n} h_i = p + 1$

→ As already mentioned, $p+1$ is the df in the sense that it indicates flexibility. However, like ridge regression, the flexibility of a smoothing spline cannot be measured by the # of regression coefficients.

→ The effective df measures flexibility by summing values similar to leverages. For clarity, we will use $\hat{y}$ instead of $g(x)$ to represent the fitted response of a smoothing spline. Specifically, let $\hat{y}_{\lambda i}$ be the fitted response for the i^{th} observation given tuning parameter value of λ. Then there exists a matrix S_λ such that

$$\hat{y}_\lambda = S_\lambda y$$

→ Denote the i^{th} diagonal entry of S_λ as $h_{\lambda i}$. As a result, the effective df is

$$df_\lambda = \sum_{i=1}^{n} h_{\lambda i}$$

→ As λ increases from 0 to ∞, the effective df from n to 2, alluding to

→ the n df of the overfitted natural cubic spline &
→ the 2 df of SLR

→ Furthermore, recall that performing LOOCV doesn't require n fits to calculate the LOOCV error when using OLS estimation, i.e. an earlier equation for LOOCV error only needs the single fit w/ all the observations. This is exactly my stance for smoothing splines when LOOCV is used to tune λ

$$\text{LOOCV error} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_{\lambda i}}{1 - h_{\lambda i}} \right)^2$$