Transformation of the X's

Transformations like Box-cox, log, or square-root can be applied on predictors too. In this section, we focus on the type of transformations of X's which in fact generates new predictors.

- Polynomials Regression
- Local Polynomials (Splines) Regression.
- From now on, assume we have only one predictor.

Polynomials Regression

- Assume $x \in \mathbb{R}$:

$$y = \beta_0 + \beta_1 x + \cdots + \beta_d x^d + \text{err}.$$

How to choose d?

- Forward approach: keep adding terms until the added term is not significant.

- Backward approach: start with a large d, keep eliminating the insignificant term starting with the highest order term.

- Question: Suppose we've picked d, then should we test whether the other terms, x^j's with $j = 1, \ldots, d-1$, are significant or not?

  Usually, we don't test the significance of the lower-order terms. When we decide to use a polynomial with degree d, by default, we include all the lower-order terms in our model.

- Why? For regression analysis, we usually don't want our results affected by any location/scale change of the data. (What if the temperature is recorded by *F* not *C*?) Suppose the data $\{y_i, x_i\}_{i=1}^n$ are generated by

$$y_i = x_i^2 + e_i, \quad e_i \sim \mathsf{N}(0, \sigma^2).$$

  But the data are recorded as $\{y_i, z_i\}_{i=1}^n$ where $z_i = x_i + 2$, that is,

$$y_i = (z_i - 2)^2 + e_i = 4 - 4z_i + z_i^2 + e_i.$$

  So the linear term could become significant if we shift the x values.

- However, if you have a particular polynomial function in mind, e.g., the data are collected to test a particular physics formula $Y \approx X^2 +$ constant, then you should test whether you can drop the linear term.

- Or if experts believe the relationship between Y and X should be $Y \approx (X-2)^2$, then you should check the R output for

  ```
  lm(Y ~ X + I((X-2)^2))
  ```

  to test whether you can drop the linear term and the intercept.

Piece-Wise Polynomials

- If the true mean $\mathbb{E}(Y \mid X = x) = f(x)$ is too wiggly, we have to fit the data using a high-order polynomial. But high-order polynomials are not recommended in practice: results are not stable and difficult to interpret.

- Instead we'll consider piece-wise polynomials: we divide the range of x into several intervals, and within each interval $f(x)$ is a low-order polynomial, e.g., cubic or quadratic, but the polynomial coefficients change from interval to interval; in addition we require overall $f(x)$ is continuous up to certain derivatives.

Cubic Splines

- knots: $a < \xi_1 < \xi_2 < \cdots < \xi_m < b$
- A function g defined on $[a, b]$ is a cubic spline w.r.t knots $\{\xi_i\}_{i=1}^m$ if:

1) g is a cubic polynomial in each of the $m + 1$ intervals,

$$g(x) = d_i x^3 + c_i x^2 + b_i x + a_i, \quad x \in [\xi_i, \xi_{i+1}]$$

where $i = 0 : m$, $\xi_0 = a$ and $\xi_{m+1} = b$;

2) g is continuous up to the 2nd derivative: since g is continuous up to the 2nd derivative for any point inside an interval, it suffices to check

$$g^{(0,1,2)}(\xi_i^+) = g^{(0,1,2)}(\xi_i^-), i = 1 : m.$$

- How many free parameters we need to represent g? $m + 4$.

  We need 4 parameters (d_1, c_i, b_i, a_i) for each of the $(m+1)$ intervals, but we also have 3 constraints at each of the m knots, so

$$4(m+1) - 3m = m + 4.$$

Suppose the knots $\{\xi_i\}_{i=1}^m$ are given.

If $g_1(x)$ and $g_2(x)$ are two cubic splines, so is $a_1 g_1(x) + a_2 g_2(x)$, where a_1 and a_2 are two constants.

That is, for a set of given knots, the corresponding cubic splines form a linear space (of functions) with dim $(m + 4)$.

- A set of basis functions for cubic splines (wrt knots $\{\xi_i\}_{i=1}^m$) is given by

$$h_0(x) = 1;\ h_1(x) = x;$$

$$h_2(x) = x^2;\ h_3(x) = x^3;$$

$$h_{i+3}(x) = (x - \xi_i)_+^3, \quad i = 1, 2, \ldots, m.$$

- That is, any cubic spline $f(x)$ can be uniquely expressed as

$$f(x) = \beta_0 + \sum_{i=1}^{m+3} \beta_j h_j(x).$$

- Of course, there are many other choices of the basis functions. For example, `R` uses the B-splines basis functions.

Natural Cubic Splines (NCS)

- A cubic spline on $[a, b]$ is a NCS if its second and third derivatives are zero at a and b.
- That is, a NCS is linear in the two extreme intervals $[a, \xi_1]$ and $[\xi_m, b]$. Note that the linear function in two extreme intervals are totally determined by their neighboring intervals.
- The degree of freedom of NCS's with m knots is m.
- For a curve estimation problem with data $(x_i, y_i)_{i=1}^n$, if we put n knots at the n data points (assumed to be unique), then we obtain a smooth curve (using NCS) passing through all y's.

Regression Splines

- A basis expansion approach:

$$g(x) = \beta_1 h_1(x) + \beta_2 h_2(x) + \cdots + \beta_p h_p(x),$$

where $p = m + 4$ for regression with cubic splines and $p = m$ for NCS.

- Represent the model on the observed n data points using matrix notation,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{F}\boldsymbol{\beta}\|^2,$$

where

$$
\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}_{n\times 1} = \begin{pmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_p(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_p(x_2) \\ & & & \\ h_1(x_n) & h_2(x_n) & \cdots & h_p(x_n) \end{pmatrix}_{n\times p} \begin{pmatrix} \beta_1 \\ \cdots \\ \beta_p \end{pmatrix}_{p\times 1}
$$

- We can obtain the design matrix F by commands `bs` or `ns` in `R`, and then call the regression function `lm`.
- Use K-fold CV to select the number of knots.

Understand how R counts the degree-of-feedom.

- To generate a cubic spline basis for a given set of x_i's, you can use the command `bs`.
- You can tell `R` the location of knots.
- Or you can tell `R` the df. Recall that a cubic spline with m knots has $m + 4$ df, so we need $m = \text{df} - 4$ knots. By default, `R` puts knots at the $1/(m + 1), \ldots, m/(m + 1)$ quantiles of $x_{1:n}$.

How `R` counts the df is a little confusing. The `df` in command `bs` actually means the number of columns of the design matrix returned by `bs` . So if the intercept is not included in the design matrix (which is the default), then the `df` in command `bs` is equal to the real df minus 1.

So the following three design matrices (the first two are of $n \times 5$ and the last one is of $n \times 6$) correspond to the same regression model with cubic splines of df 6.

```
> bs(x, knots=quantile(x, c(1/3, 2/3)));
> bs(x, df=5);
> bs(x, df=6, intercept=TRUE);
```

- To generate a NCS basis for a given set of x_i's, use the command `ns`.

- Recall that the linear functions in the two extreme intervals are totally determined by the other cubic splines. So data points in the two extreme intervals (i.e., outside the two boundary knots) are wasted since they do not affect the fitting. Therefore, by default, `R` puts the two boundary knots as the min and max of x_i's.

- You can tell `R` the location of knots, which are the interior knots. Recall that a NCS with m knots has m df. So the df is equal to the number of (interior) knots plus 2, where 2 means the two boundary knots.

- Or you can tell `R` the df. If `intercept = TRUE`, then we need $m = \text{df} - 2$ knots, otherwise we need $m = \text{df} - 1$ knots. Again, by default, `R` puts knots at the $1/(m+1), \ldots, m/(m+1)$ quantiles of $x_{1:n}$.

- The following three design matrices (the first two are of $n \times 3$ and the last one is of $n \times 4$) correspond to the same regression model with NCS of df 4.

```
> ns(x, knots=quantile(x, c(1/3, 2/3)));

> ns(x, df=3);

> ns(x, df=4, intercept=TRUE);
```