STAT 425

Polynomial Regression

Transformations of the Predictors

- We discussed transformations on the response variable Y to stabilize the variance and to normalize the response. Some of these transformations are the Box-Cox transformation, log, square-root and so on. We can apply these transformations to the 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

Polynomial Regression

Assume $x \in \Re$ and a model of the form:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_d x^d + error$$

d is the degree of the polynomial component. How do we choose d?

- Forward approach: Keep adding terms until the last added term is not significant
- Backward approach: start with a large d, keep eliminating the insignificant terms starting with the highest order term.

- Question: Suppose we have picked a value of d, then should we test whether the other terms, x^j's with $j = 1, \ldots, d-1$, are significant or not? Usually we do not test the significance of the lower-order terms. When we decide to use a polynomial of degree d, by default, we include all the lower-order terms in our model.
- Why is this?. For regression analysis, we usually do not want our results to be affected by a change of location/scale of the data (For example, suppose that the temperature is recorded in F instead of C). Suppose the data $\{y_i, x_i\}_{i=1}^n$ are generated by the model:

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

But the data are recorded as $\{z_i, x_i\}_{i=1}^n$, 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 the model $lm(Y \sim X + I((X-2)^2)$ to test whether you can drop the linear term and the intercept.

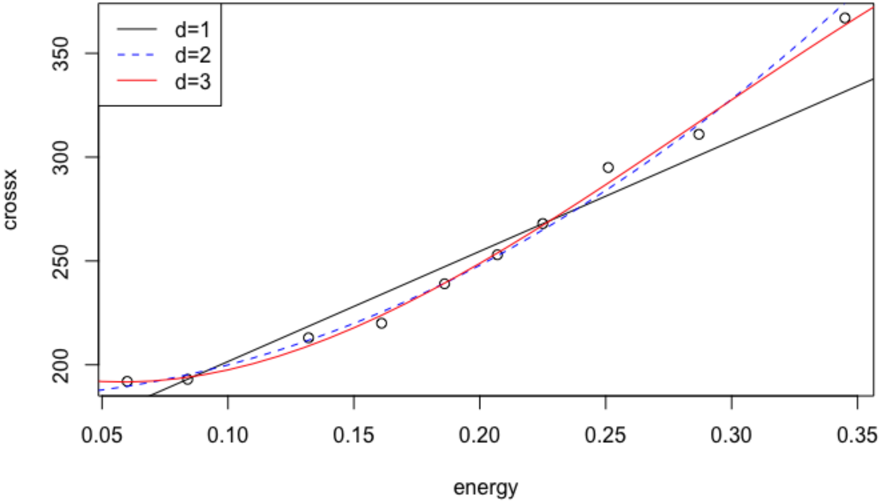
Example: strongx data set

Selecting order d for the polynomial fitted variable *crossx* as a function of *energy*

```
round(summary(lm(crossx ~ energy + I(energy^2) + I(energy^3), weights = sd^-2, strongx)
coef, dig=3)
```

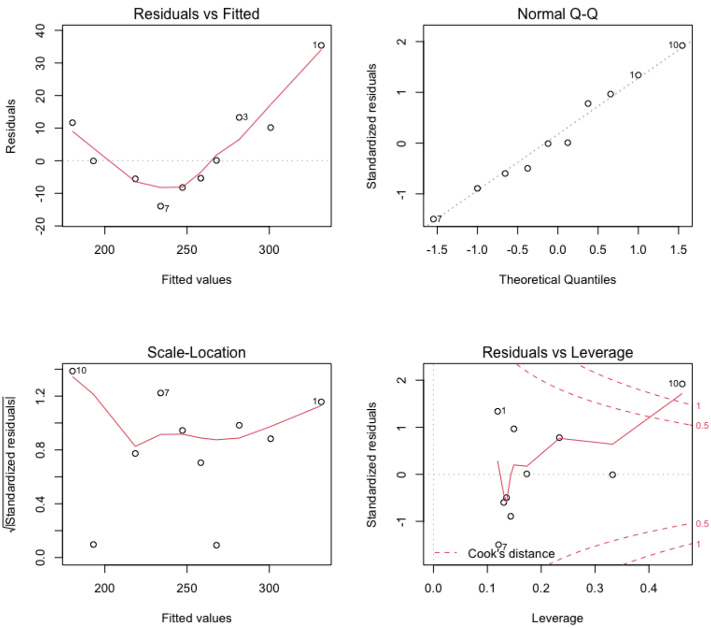
```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     204.992     12.826  15.983    0.000
## energy         -472.866    268.862  -1.759    0.129
## I(energy^2)    4504.475   1600.789   2.814    0.031
## I(energy^3)   -5220.597   2848.373  -1.833    0.117
```

- The cubic term is not significant. We can use a quadratic polynomial.
- Backward and forward elimination methods for d selection yield the same result ($d = 2$). This is not always the case.



d=1
d=2
d=3
crossx
energy
350
300
250
200
0.05
0.10
0.15
0.20
0.25
0.30
0.35

Residuals for the linear model $lm(crossx \sim energy)$



Orthogonal Polynomials

- Fitting high order polynomials is generally do not recommended, since they are very unstable and difficult to interpret.
- Successive predictors x^j are highly correlated introducing multicollinearity problems.
- One way around this is to fit orthogonal polynomials of the form:

$$y_i = \beta_0 + \beta_1 z_1 + \ldots + \beta_d z_d + error$$

  where each $z_j = a_1 + b_2 x + \ldots + \kappa_j x^j$ is a polynomial of order j. Its coefficients are chosen such that $z_i^\top z_j = 0$
- Use function **poly(.)** in **R** to fit orthogonal polynomials.

Standard polynomials vs. Orthogonal polynomials

```
# Plot polynomials with order 1 to 3
# Orthogonal polynomials
x=seq(0, 1, 0.01)
outo= poly(x, 3)
# standard polynomials
outs = cbind(x, x^2, x^3);
```

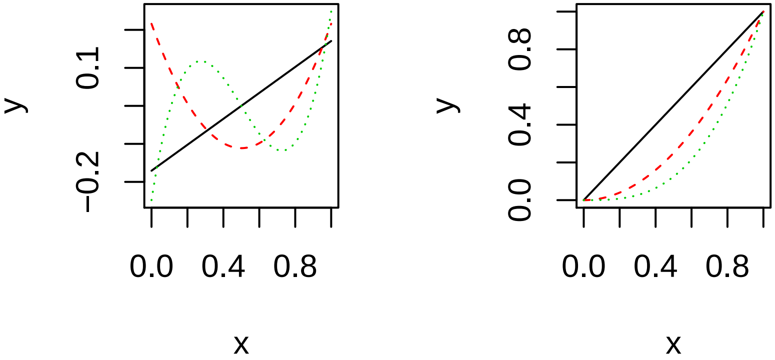


Figure: Orthogonal (left) vs. Standard (right)

Piece-wise Polynomials

- If the true mean of $E[Y|X = x] = f(x)$ is too wiggly, we might need to fit a higher order polynomial, which is not always a good idea.
- Instead we will 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 the overall $f(x)$ to be continuous up to certain derivatives.

Cubic Splines

We want to define a cubic spline function in the interval $[a, b]$

- Define m knots such that: $a < \xi_1 < \xi_2 < \ldots < \xi_m < b$
- A function g defined on $[a, b]$ is a cubic spline with respect to (wrt) 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 the following conditions:

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

This expression indicates that the function and the first and second order derivatives are continuous at the knots.

- How many free parameters we need to represent g?
  We need 4 parameters (d_i, 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 (continuity constraints). The total number of free parameters (similar to the number of degrees of freedom) is:

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

Some properties of the cubic splines

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

- If $g_1(x)$ and $g_2(x)$ are cubic splines, the linear combination $a_1 g_1(x) + a_2 g_2(x)$ is also a cubic spline, where a_1 and a_2 are known 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 (w.r.t knots $\{\xi_i\}_{i=1}^m$) is given by:

$$
\begin{gathered}
h_0(x) = 1; h_1(x) = x; \\
h_2(x) = x^2; h_3(x) = x^3; \\
h_{i+3}(x) = (x - \xi_i)_+^3, \quad i = 1, 2, \ldots, m
\end{gathered}
$$

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

$$f(x) = \beta_o + \sum_{j=1}^{m+3} \beta_j h_j(x)$$

- There are many other choices of basis functions. For example, **R** uses the B-splines basis functions.

Natural Cubic Spline (NCS)

- A cubic spline on $[a, b]$ is a NCS if its second and third derivatives are zero at a and b.
- This condition implies that NCS is a linear function in the two extreme intervals $[a, \xi_1]$ and $[\xi_m, b]$. The linear functions in the two extreme intervals are completely determined by their neighboring intervals.
- The degree of freedom of NCS's with m knots is m:

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

  (4 additional constraints)
- 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.