Moving Beyond Linearity: Extension of linear models

Fangfang Song

June 2018

The truth is never linear. When its not, the methods below offer a lot of flexibility without losing the ease and interpretability of linear models

1 Polynomial Regression

1. Usually, the degree of polynomial d is not greater than 3 or 4, because for large values of d, the polynomial curve can become overly flexible and can take on some very strange shapes, especially near the boundary of the X variables. Or we say polynomials have notorious tail behavior, which is very bad for extrapolating.

2. We can use cross validation to pick d

2 Regression Spline

Instead of a single polynomial in X over its whole domain, we can rather use different polynomials in regions defined by knots. But it's better to add continuity constrains to the polynomial. *Splines* have the maximum amount of continuity(up to $d-1$th derivatives are continuous).For example, a cubic spline with knots at $\epsilon_k, k = 1, ...K$ is a piecewise cubic polynomial with continuous derivative up to order 2 at each knot. If the dth derivative is continuous, it will be a global continuous polynomial functions.

Each constrains we impose on the piece wise polynomials effectively reduces one degree of freedom.

How do we fit a piecewise degree-d polynomial under the constraint that the function and the first $d-1$ derivatives be continuous? It turns out that we can use **truncated power basis function** to represent a regression spline.

For example, culib spline can be written as :

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + ... + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i \tag{1}$$

where the basis functions are :

$$
\begin{gathered}
\mathrm{b}_1(x_i) = x_i \\
\mathrm{b}_2(x_i) = x_i^2; \\
\mathrm{b}_2(x_i) = x_i^3; \\
\ldots \\
\mathrm{b}_{k+3}(x_i) = (x_i - \epsilon_k)_+^3, k = 1, 2, ...K
\end{gathered}
$$

It can be shown the truncated power basis ensures that the function remian continuous, with continuous first and second derivatives at each of the knots.

To summarize, regression splines are created by specifying a set of knots, producing a sequence of basis function, and then using least squares to estimate the spline coefficients.

2.1 Natural Cubic Spline

A natural cubic spline extrapolates linearly beyond the boundary knots. This adds $2 \times 2 = 4$ extra constraints, and allow us to put more internal knots for the same degree of freedom as a regular cubic spline.

A cubic spline with K knots has $K + 4$ parameters or degrees of freedom. A natural spline with K knots has K degree of freedom.

3 Smoothing Splines

In fitting a smooth curve to a set of data, what we really want to do is find some function, say $g(x)$, that fits the observed data well. However, if we don't put any constraints on $g(x_i)$, then we can always make RSS 0 simply by choosing a very flexible function. What we really want is a function g that makes RSS small, but that is also smooth.

Consider the criterion for fitting a smooth function $g(x)$:

$$
minimize_{g \in S} \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt
$$

The first term is RSS, and tries to make g(x) match the data at each x_i. The second term is a roughness penalty and control how wiggly g(x) is. λ controls the bias -variance trade off of the smoothing spline.

The solution that minimizes the above equation can be shown to have some special properties: it is a piecewise cubic polynomial with knots at every unique values of x_i, and continuous first and second derivatives at each knot. Furthmore, it's linear in the region outside of the extreme knots. In other words, the function $g(x)$ is a natural cubic spline with knots at $x_1, x_2, ...x_n$

The smoothing splines avoid the knot-selection issue, leaving a single λ to be chosen.

4 Local Regression

Local regression compute the fit at a target point x_0 using only the nearby training observations. In order to obtain the local regression fit at a new point, we need to fit a new weighted least squares regression model for a new set of weights. Local regression is sometimes referred to as a memory based procedure, since we need all the training data each time we wish to compute a prediction.

To perform local regression, there are a number of choices to be made, such as how to define the weighting function K, and whether to fit a linear, a constant, or quadratic regression.

1. Gather the fraction $s = k/n$ of training points whose x_i are closest to x_0.
2. Assign a weight $K_{i0} = K(x_i, x_0)$ to each point in this neighborhood, so that the point furthest from x_0 has weight zero, and the closest has the highest weight. All but these k nearest neighbors get weight zero.
3. Fit a weighted least square regression of the y_i on the x_i using the weights, by finding $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize $\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)$

We would like to fit locally linear, instead of local constant or moving average, is because it does much better at the boundaries.

5 Generalized Additive Models

Generalized linear models provides a general framework for extending a standard linear model by allowing non-linear functions of each of the variables, while maintaining additivity.