DSA 8305

Moving Beyond Linearity

2025 May 08 (Thu)

Introduction

We have focused on linear models which are simple and easy to interpret. However, these models can be limited, especially their predictive power – due to the linearity assumption which might not be entirely met.

In this section, we intend to relax the linear assumptions but maintain model interpretability as much as possible.

We will discuss a set of models which extends linear models, including:

- Polynomial regression
- Step functions
- Regression splines
- Smoothing splines
- Local regression
- Generalized additive models (GAMs)

Polynomial Regression

Extends linear models by adding extra predictor – each of the original predictor is raised to a power. This makes the relationship between the response and the predictor become non-linear. For example:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

linear model can be extend to a polynomial model:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d + \epsilon_i$$

For large d, the polynomial regression produces extremely non-linear curve. The parameters of this model can be estimated using least squares approach as in the previous sections – the model is treated as just a standard linear model with predictors $(x_i, x_i^2, \cdots, x_i^d)$.

The d is the degree of the polynomial, and in most cases, $d = 3$ or $d = 4$ is good enough. Values higher than this can lead to a polynomial which is very flexible, with very strange shapes.

Example 1

Consider an example to model individual's **wage** as a function of their **age**, with a polynomial regression model of 4-degree:

$$wage = \beta_0 + \beta_1 Age + \beta_2 Age^2 + \beta_3 Age^3 + \beta_4 Age^4 + \epsilon_i$$

Even though this is a linear model just like any other we previously discussed, we do not focus on the individual coefficients but instead, we look at the entire fitted function, i.e.,

$$\hat{f}(Age) = \hat{\beta}_0 + \hat{\beta}_1 Age + \hat{\beta}_2 Age^2 + \hat{\beta}_3 Age^3 + \hat{\beta}_4 Age^4$$

with the variance

$$\text{Var}[\hat{f}(Age)] = X^T \hat{V} X$$

This quantity can be used to approximate the 95% CI.

Example 1 Cont'd

For a binary outcome, e.g., by converting **Wage** to binary i.e., income $> 250K$ (high income) and $< 250K$ (low income). We can then apply logistic regression with **Age** as a polynomial predictor of degree 4:

$$P(wage = high|Age) = \frac{\exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d)}{1 + \exp \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d}$$

Step Functions

This approach divides the range of the predictor into K distinct regions, i.e., onto a piecewise function. As opposed to polynomial function which imposes a global structure on the non-linear function of the predictor, this approach breaks the range of predictor into bins, and fit different constant in each bin, which has the effect that it converts continuous predictor into an ordered categorical predictor.

In particular, define $C_1, C_2, \cdots, C_k$, cut points in the range of X, and then generate $k + 1$ new variables

$$
\begin{aligned}
C_0(X) &= I(X < C_1) \\
C_1(X) &= I(C_1 \leq X < C_2) \\
C_2(X) &= I(C_2 \leq X < C_3) \\
&\vdots \\
C_{k-1}(X) &= I(C_{k-1} \leq X < C_k) \\
C_k(X) &= I(C_k \leq X)
\end{aligned}
$$

Step Functions Cont'd

Where I is an indicator function, returning 1 if the condition is met, 0 otherwise. These are sometimes referred to as dummy variables.

For any value of X,

$$C_0(X) + C_1(X) + \cdots + C_k(X) = 1$$

This is because X must be in exactly one of the $k + 1$ intervals.

The $C_0(X), C_1(X), \cdots, C_k(X)$ are the predictor used on the model

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \cdots + \beta_k C_k(x_i) + \epsilon_i$$

Note: If there are no natural breakpoints in the predictor, piecewise-constant functions can miss some trends.

Basis functions

In this approach, we apply a set of functions or transformations on X, such that $b_1(X), b_2(X), \cdots, b_k(X)$.

Polynomial & piecewise-constant regression models are special cases of the basis function method.

In the basis function method, instead of fitting a linear model, we fit:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + +\beta_k b_k(x_i) + \epsilon_i$$

where $b_1(.), b_2(.), \cdots, b_k(.)$ are the basis functions, chosen before fitting the model.

For instance, for polynomial regression, the basis functions are

$$b_j(x_i) = x_i^2$$

and for piecewise-constant functions, the basis functions are

$$b_j(x_i) = I(C_j \leq x_i C_{j+1})$$

Basis functions Cont'd

This model is treated as the standard linear model and we can make all the inference about the model, as we did in the linear models.

Regression splines

These are flexible basis functions that extend the methods discussed above.

(a) Piecewise polynomial

We pick several low degree polynomials and fit them over different regions of X, as opposed to fitting a high to fitting high degree polynomial on the whole range of X.

For example, consider the cubic polynomial

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i,$$

we can fit a piecewise cubic polynomial by choosing coefficients $\beta_0, \beta_1, \beta_2$ and β_3 which differ in various parts of the range of X. These parts are called **Knots**. The cubic polynomial has no knots.

A cubic polynomial with a single Knot c is defined as

$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i & \text{if} \quad x_i < c \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i & \text{if} \quad x_i \geq c \end{cases}$$

(a) Piecewise polynomial cont'd

In other words, we fit two different polynomials to the subsets of data i.e., when $x_i < c$, otherwise when $x_i \geq c$. The first polynomial has coefficients β_{01}, β_{01} and β_{31} while the second polynomial has β_{02}, β_{12} and β_{32}.

More Knots will lead to a more flexible polynomial, with $k + 1$ different cubic polynomial if we have k different Knots on the entire range of X.

(b) Constraints and splines

In some cases, there could be "jumps" at the Knots. In such cases, we can add a constraint so that the fitted curve is continuous. This ensures that both **first** and **second** derivatives are continuous at the Knots.

The constraint we impose reduces the model complexity of the piecewise polynomial by reducing the degrees of freedom. For example, a cubic spline with k knots, has $4 + k$ degrees of freedom.

(c) Spline basis representation

Suppose we want to fit a piecewise degree-d polynomial under the constraint that it is continuous at all the Knots i.e., all the $d-1$ first derivates are continuous.

For instance, the basis representation of a cubic spline with k Knots is defined as:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{k+3} b_{k+3}(x_i) + \epsilon_i$$

where $b_1, b_2, \cdots, b_{k+3}$ are the basis functions.

There are several ways to represent cubic splines using various choices of the basis functions. For instance, we can start with cubic polynomial basis i.e., x, x^2, x^3 and then add truncated power functions:

$$\begin{aligned} h(x,\xi) &= (x-\xi)_+^3 \\ &= \begin{cases} (x-\xi)^3 & x > \xi \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

per Knot, ξ.

(c) Spline basis representation cont'd

All the models discussed above can be fitted via least squares regression. For instance, for the cubic spline, we fit a model with intercept and $3 + k$ predictors:

$$(x, x^2, x^3, h(x, \xi_1), h(x, \xi_2), \cdots, h(x, \xi_k))$$

where $\xi, \cdots, \xi_k$ are the knots, which translates to estimating a total of $k + 4$ regression coefficients – with $k + 4$ degrees of freedom.

Note: *Natural splines* – The confidence bands for cubic splines may be off, especially, at the boundary. To remedy this, we can introduce boundary constraints to produce a more stable estimates at the boundary.

(d) Choosing the number and location of the Knots

Where should we place the knots? – When there are so many knots, the spline is very flexible due to the shifts in the coefficients.

Therefore, one option would be to place more knots where there are more variations and less in stable regions. We can achieve this is to specify the degrees of freedom and let **R** automatically place the number of knots at the uniform quantile of the data.

Now, how many Knots or degrees of freedom should the spline contain?

- We can try different knots and see which produces the best fitting curve. However, a better approach is to use **cross-validation**and pick the number of knots which gives the smallest **RSS**.

(e) Comparison to polynomial regression

Remember, for regression, would want to choose $g(x)$ such that

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

is very small.

Ideally, what we want is to choose $g(.)$ that makes RSS very small and also smooth. A natural way is to find a function $g(.)$ that minimizes

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

where $\lambda > 0$ is the tuning parameter.

The function $g(.)$ that minimizes the equation above is the **smoothing spline**.

Smoothing splines

The equation is of the form

$$Loss + Penalty$$

and $\sum_{i=1}^{n}(y_i - g(x_i))^2$ is the RSS we discussed in the previous sections, representing the *Loss* function and the term $\lambda \int g''(t)^2 dt$ is the *penalty* term that penalizes the variability in $g(.)$.

The second derivative $g''(.)$ is a measure of the *roughness* of the curve. It is large in absolute value if $g(t)$ is wiggly near t, and close to zero otherwise.

When $\lambda \longrightarrow 0$, the penalty term has no effect and hence $g(.)$ will simply interpolate the data. When $\lambda \inf$, $g(.)$ will be smooth and passes as closely as possible to data points – In other words, λ controls the *bias-variance trade-off* of the smoothing spline.

Smoothing splines cont'd

In fact $g(.)$ that minimizes the equation above is a piecewise polynomial with knots at the unique values $x_1, \cdots, x_n$ and continuous first and second derivatives at each knot – and also a shrinkage version of the *natural splines*.

(a) Choosing the smoothing parameter λ

Remember, a smoothing spline is a natural cubic spline with knots at every unique value of x_i – implying that it will have large degrees of freedom and hence allow for great deal of flexibility.

The *tuning parameter* λ controls the roughness of the smoothing spline, and hence the *effective degrees of freedom*. As $\lambda \longrightarrow \inf$, the effective df decreases from n to 2, and it is defined as:

$$df_\lambda = \sum_{i=1}^{n} \{\mathbf{S}_\lambda\}_{ii}$$

where $\mathbf{S}_\lambda$ is a $n \times n$ matrix of the fitted values and df_λ is simply the sum of the diagonal of the matrix.

(a) Choosing the smoothing parameter λ cont'd

We choose λ using cross-validation – we choose λ that makes the cross-validation RSS as small as possible.

It turns out that the *LOOCV* error can be computed very effectively for smoothing splines, with the same computational cost as a single fit, using the formula

$$\begin{aligned} RSS(\lambda) &= \sum_{i=1}^{n} (y_i - \hat{g}_{\lambda}^{(-1)}(x_i))^2 \\ &= \sum_{i=1}^{n} \left[\frac{y_i - \hat{g}_{\lambda}(x_i)}{1 - \{\mathbf{S}_{\lambda}\}_{ii}} \right] \end{aligned}$$

where $\{\mathbf{S}_{\lambda}\}_{ii}$ denote the fitted value for the smoothing spline evaluated at x_i, where the fit uses all the of the data except the ith observation (x_i, y_i). On the other hand, $\hat{g}_{\lambda}(x)$ indicates the smoothing spline function fit to all the data, evaluated at x_i.

Generalized Additive Models (GAMs)

All the methods discussed in the previous sections are based on a single predictor – extension of simple linear models. Here, we explore an extension of the methods to include multiple predictors $X_1, \cdots, X_p$, i.e., an extension of multiple linear models.

GAMs provide a general framework for extending standard linear models by allowing non-linear functions of each predictor, while maintaining *additivity*. GAMs can be used for both gaussian and non-gaussian outcomes.

(a) GAMs for Regression Models

Consider a multiple linear model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i$$

For GAMs, we replace each predictor with a (smooth) non-linear function $f_j(x_{ij})$, thus

$$\begin{aligned} y_i &= \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i \\ &= \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i \end{aligned}$$

This model is called additive model because we calculate f_j separately for each x_j, and add all together their contribution.

For non-gaussian outcome, we apply the same concept and pass the RHS (non-linear predictor) through the appropriate link function.