Study Notes of Regression Splines

Lingtian Bu*
WISE 2017

Update: November 3, 2019

Abstract

After learning J. Mao's lecture- ***Regression***, I have learned a lot of staff about regression such as different kinds of models(linear, polynomial, regression splines), model selection, hypothesis testing, Bootstrap, piecewise constant regression and so on. In this assignment, I focus on the study of ***Regression Splines***: motivation, definition, expression and examples with simulation data.

Keywords: linear regression polynomial regression regression splines linear basis model

1 Motivation

1.1 Linear Rgression

Linear regression is one of the most frequently used regression model in our study. Its basic expression is:

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} + e_i \quad i = 1, \ldots, n \tag{1}$$

The regression fuction is just the conditional expectation function(CEF), $E(y|x)$. Calculating OLS estimator by minimizing the in-sample error, we can get:

$$\overrightarrow{\beta} = (X'X)^{-1}X'Y \tag{2}$$

1.2 Polynomial Regression

1.2.1 Definition

A polynomial of degree D is a function formed by linear combinations of the powers of its argument up to D:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_D x^D + e \tag{3}$$

OlS model with only one independent variable is a special case of polynomial regression model.

* 卜令天 15220172202239

1.3 Examples using linear and polynomial regression

First, we simulate data for our testing.

```
# simulate data

## signal
f = function(x) {
x ^ 3
}

## define data generating processs
get_sim_data = function(f, sample_size = 50) {
x = runif(n = sample_size, min = -1, max = 1)
y = rnorm(n = sample_size, mean = f(x), sd = 0.15)
data.frame(x, y)
}

## simualte training data
set.seed(42)
sim_trn_data = get_sim_data(f = f)

## simulate testing data
set.seed(3)
sim_tst_data = get_sim_data(f = f)

## create grid for plotting
x_grid = data.frame(x = seq(-1.5, 1.5, 0.001))
```

Then I choose three dimension for D, 1, 3 and 22. Degree 1 respects OLS model, degree 3 is a cubic regression model as well as degree 22 is a relatively extreme case of polynomial regression model.

```
# fit the model

## polynomial models
poly_fit_l = lm(y ~ poly(x,  1), data = sim_trn_data)
poly_fit_m = lm(y ~ poly(x,  3), data = sim_trn_data)
poly_fit_h = lm(y ~ poly(x, 22), data = sim_trn_data)

# get predictions

## polynomial models
poly_fit_l_pred = predict(poly_fit_l, newdata = x_grid)
poly_fit_m_pred = predict(poly_fit_m, newdata = x_grid)
poly_fit_h_pred = predict(poly_fit_h, newdata = x_grid)
```

Figure 1 shows the results of fitting. It's apparent that linear model can not fit so well so it is underfitting. The one with degree 22 is too complicated to describe the true underline population and it is overfitting. The cubic model is the most fitted model seemingly among these three.

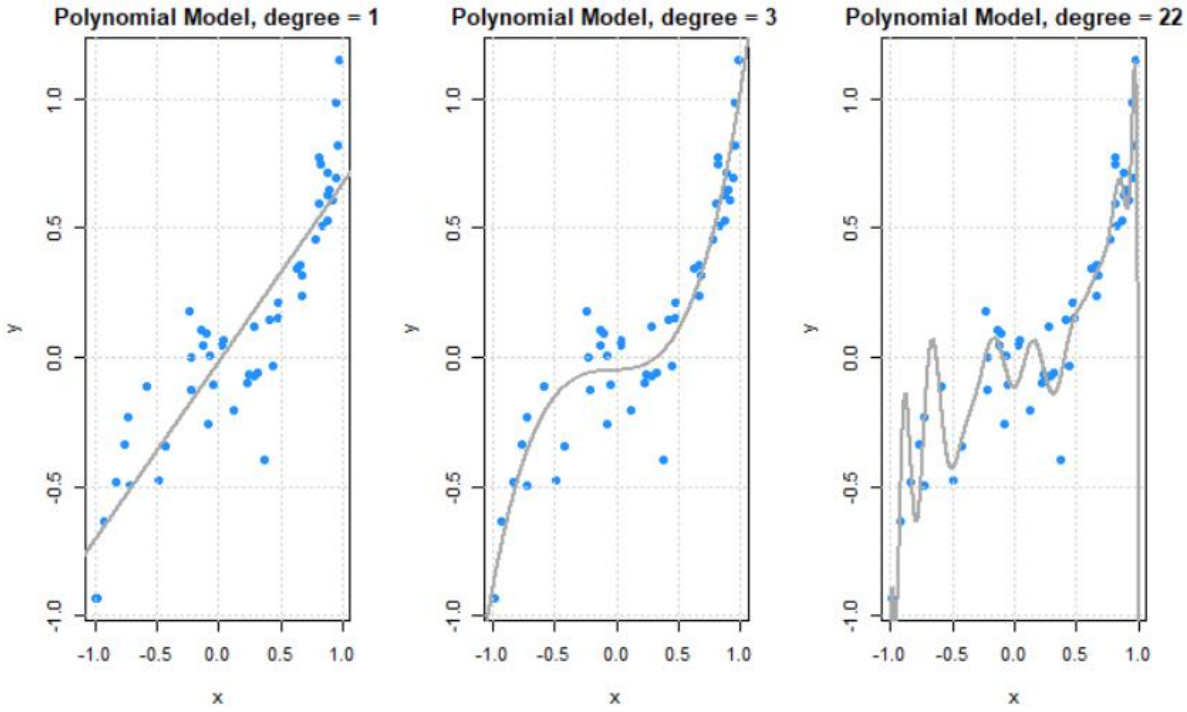

Figure 1: Fitting results of polynomial model

1.4 Motivation

So far, what we have discussed are all *global* structure on the relationship between x and y. However, we can also take a wise step to break the whole x axis into several regions to generate local models. Piecewise regression is such a model breaks the inputs space into distinct regions and fit different relationship in each region. Nervertheless, some times the fitting model is not so perfect. we are desired to generate a continuous model with no jump and smooth enough.

2 Definition

A linear spline is a continuous function formed by connecting linear segments. The points where the segments connect are called the knots of the spline.

A spline of degree D is a function formed by connecting polynomial segments of degree D so that:

- the function is continuous,
- the function has D 1 continuous derivatives, and

- the Dth derivative is constant between knots.

The truncated polynomial of degree D associated with a knot ξ_k is the function which is equal to 0 to the left of ξ_k and equal to $(x - \xi_k)^D$ to the right of ξ_k .

$$(x - \xi_k)_+^D = \begin{cases} 0 & x < \xi_k \\ (x - \xi_k)^D & x \geqslant \xi_k \end{cases} \tag{4}$$

3 expression

The equation for a spline of degree D with K knots is:

$$y = \beta_0 + \sum_{d=1}^{D} \beta_d x^d + \sum_{k=1}^{K} b_k \left((x - \xi_k)_+^D \right) \tag{5}$$

The design matrix for a spline of degree D with K knots is the n by 1 + D + K matrix with entries:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^D & (x_1 - \xi_1)_+^D & \cdots & (x_1 - \xi_K)_+^D \\ 1 & x_2 & x_2^2 & \cdots & x_2^D & (x_2 - \xi_1)_+^D & \cdots & (x_2 - \xi_K)_+^D \\ 1 & x_3 & x_3^2 & \cdots & x_3^D & (x_3 - \xi_1)_+^D & \cdots & (x_3 - \xi_K)_+^D \\ & & & \vdots & & & & \\ 1 & x_n & x_n^2 & \cdots & x_n^D & (x_n - \xi_1)_+^D & \cdots & (x_n - \xi_K)_+^D \end{bmatrix} \tag{6}$$

4 Example of regression splines

Use the data of Montreal's temperature change each day over two year.

```
D = 3
K = 5
knots = 730 * (1: K ) / ( K +1)
X1 = outer ( data $x ,1: D , "␣^␣" )
X2 = outer ( data $x , knots , "␣>" ) *
outer ( data $x , knots , "␣-" )^ D
X = cbind ( X1 , X2 )
round ( X [ c (1 ,150 ,300) ,1:5] ,1)
lmfit = lm ( y ~X , data = data )
```

The result is in figure 2

Splines computed from the truncated polynomials may be numerically unstable because: the values in the design matrix may be very large, and the columns of the design matrix may be highly correlated. Then B-spline design matrix can be constructed via the function bs provided by the splines library.

```
library ( splines )
D = 3
```

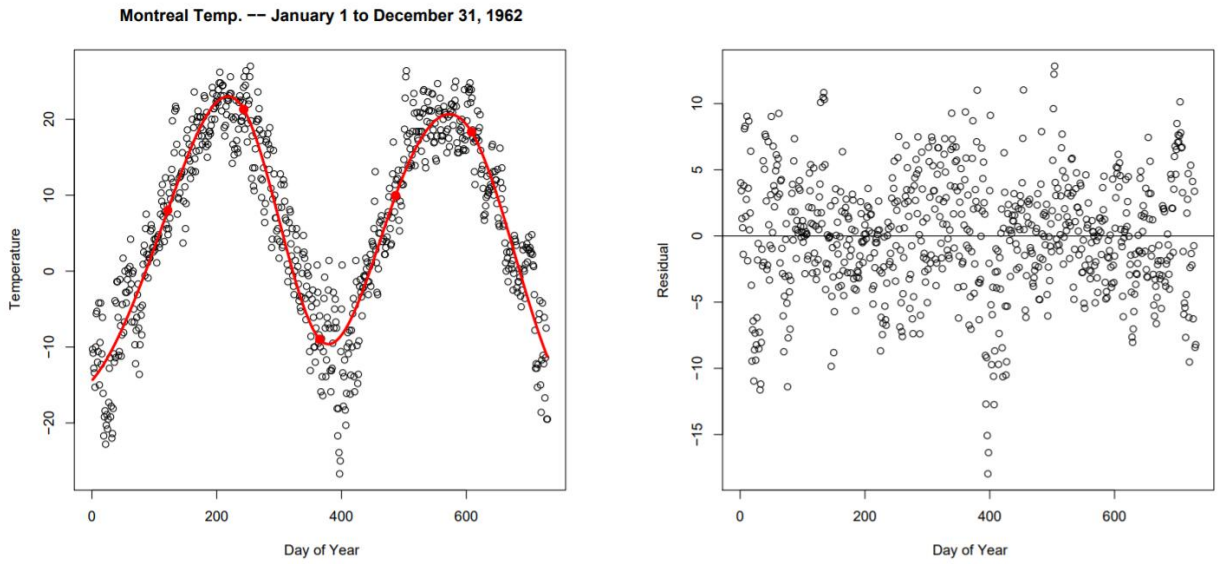

Figure 2: Fitting results of simple spline

```
K = 5
knots = 730 * (1: K ) / ( K +1)
X = bs ( data $x , knots = knots , degree =D , intercept = TRUE )
lmfit = lm ( y ~X -1 , data = data )
```

Figure 3 shows the result.

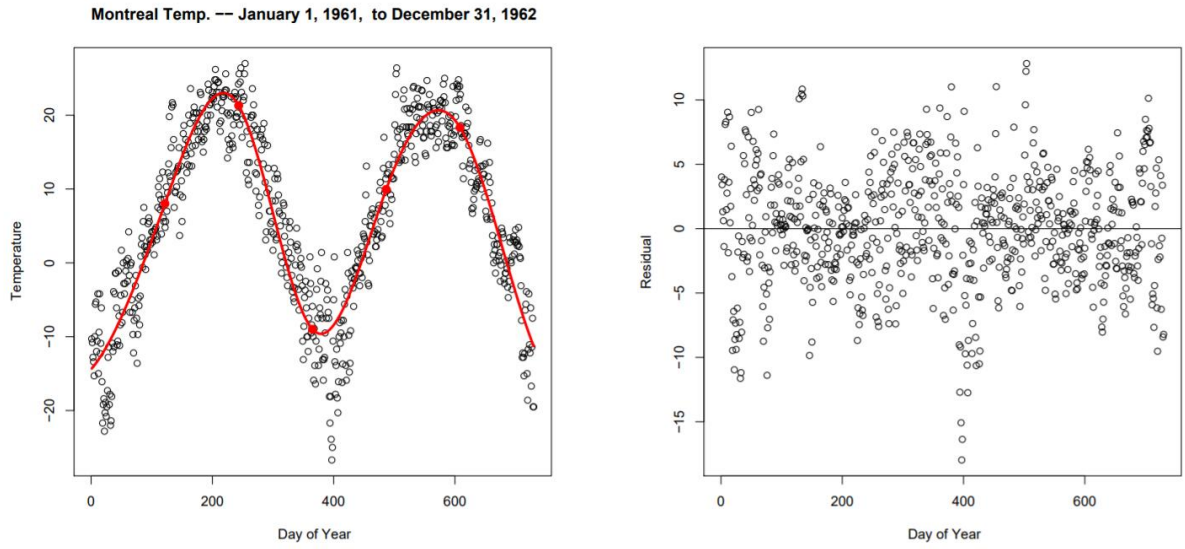

Figure 3: Fitting results of simple spline

5 References

Ng, A. *Machine Learning*. Lecture at Stanford University, retrieved on 2017.01.01.

Taddy, M. *Big Data.* Lecture at the University of Chicago Booth School of Business, retrieved on 2017.01.01.

David D. *R for Statistical Learning*