STATS 790 - Homework 3

Yicen Li

March 4, 2023

1 Question 1

We have successfully replicated Figure 5.6 from the ESL book. The book [1] provides clear details regarding the data source and the parameter settings used for the smoothing spline fit, making the replication process relatively straightforward.

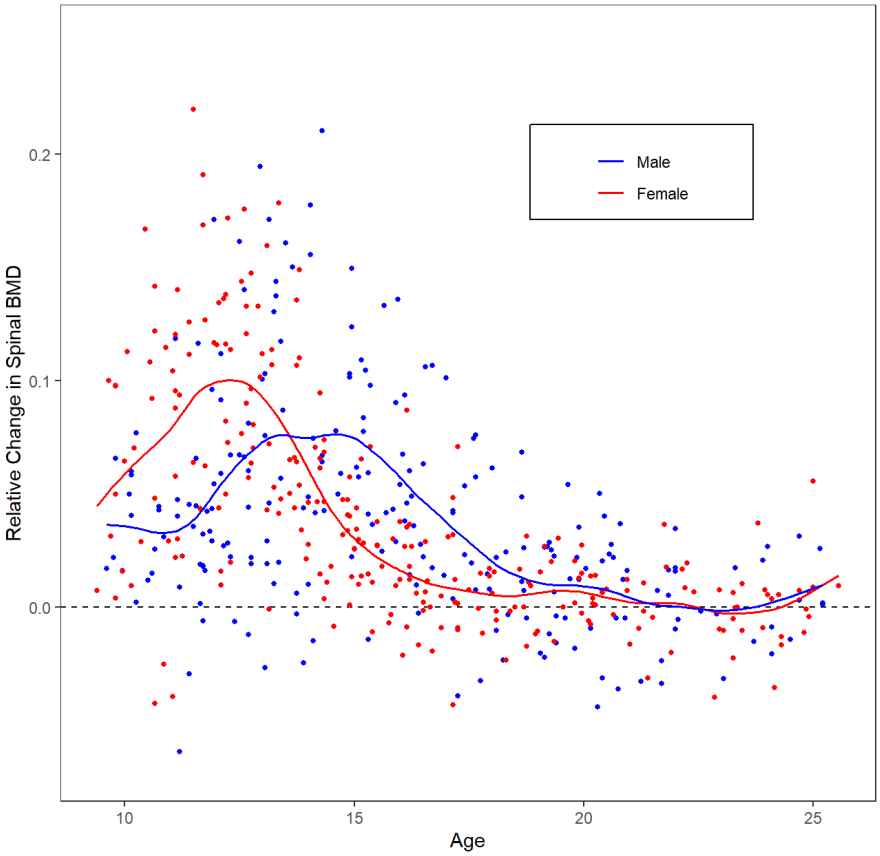

Figure 1: The replicate of smoothing spline fit figure 5.6 when $\lambda = 0.00022$ on bone mineral density dataset provided by ESL. Codes available on Github

2 Question 2

We used the South African coronary heart disease dataset and applied different spline bases to it. Our goal was to fit a logistic regression model using the tobacco variable as the predictor and the chd variable as the response. However, we encountered an issue while directly applying the quantile method to obtain five knots due to the presence of several 0 values in the tobacco level. Since the value of 0 would overlap with two quantiles, we decided to manually create six knots using the values $(0.000000, 0.000000, 0.280000, 1.211429, 3.000000, 4.642857, 7.814286, 31.20$ We then merged the first two intervals into one to obtain the final set of five knots: $0.280000, 1.211429, 3.000000, 4.642857, 7.81$

To calculate log odds, we created a simple function that uses the formula $\beta_0 + \beta_1 x_1 + ... + \beta_n x_n$, where n is the number of bases, instead of using the predict() function. The resulting figures are shown below.

(1) The figure of B-spline bases.

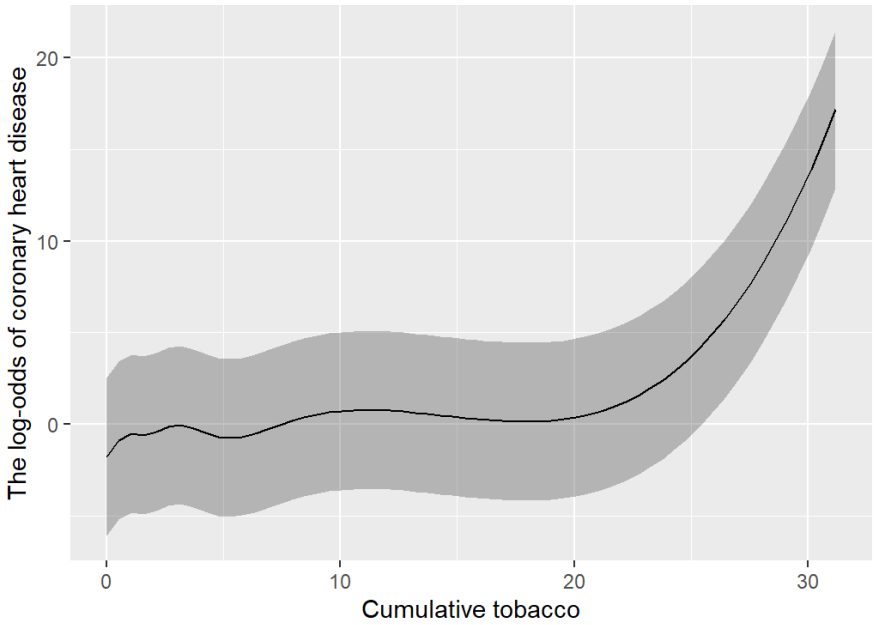

Figure 2: The predictions ± 1 SE plot for the B-spline bases. Codes available on Github

(2) The figure of truncated polynomial spline bases.

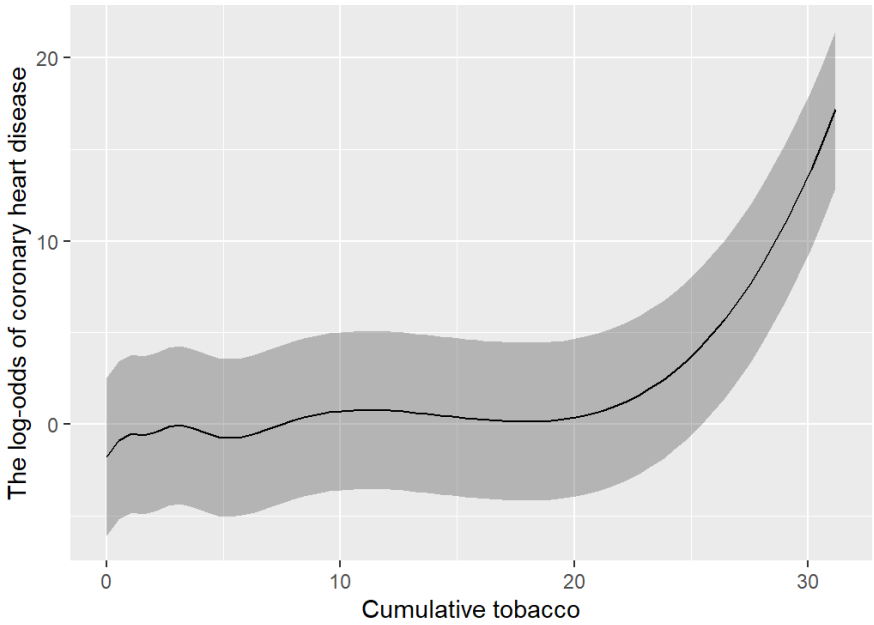

Figure 3: The predictions ± 1 SE plot for the truncated polynomial spline bases. Codes available on Github

(3) The figure of natural bases.

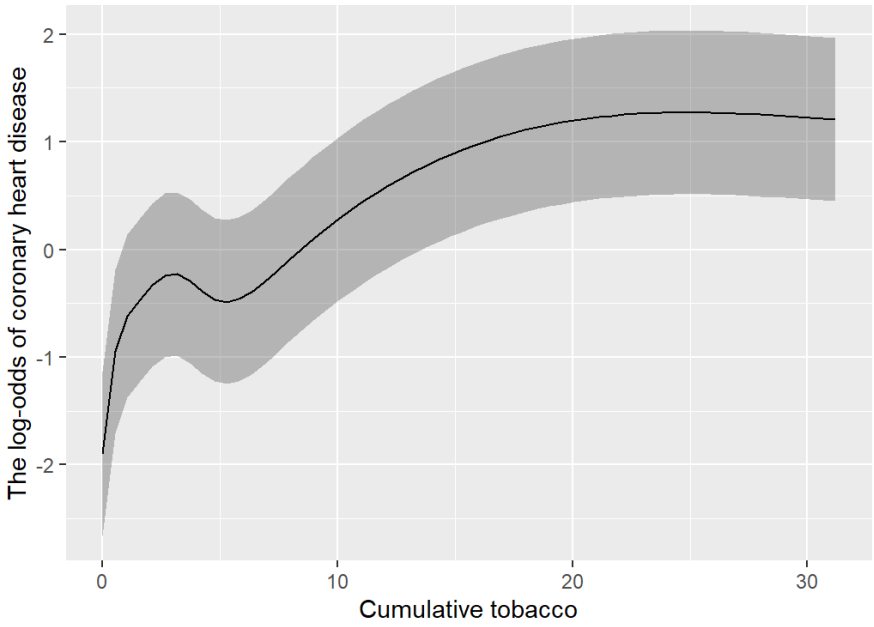

Figure 4: The example of regular and natural bases. Codes available on Github

Please note that the y-axis of the figures is represented in log odds. We observed that the shapes of the B-spline and truncated polynomial spline regressions were very similar, while the natural spline outcomes followed a different pattern. This difference could be due to the choice of knots and the disparity in the number of variables used during regression fitting, as natural spline bases have fewer degrees of freedom when compared to the others.

3 Question 3

Based on the structure of the lecture note code and the constraint equation :

$$\begin{aligned} \beta_2 &= 0 \\ \beta_3 &= 0 \\ \sum_{k=1}^{K} \theta_k &= 0 \\ \sum_{k=1}^{K} \theta_k \xi_k &= 0 \end{aligned}$$

under the corresponding basis description, we built a function providing natural truncated polynomial bases. This function will return the spline bases for the natural truncated polynomial when the input is true. Codes available on Github.

The side-by-side comparison figure on a random interval with five knots is:

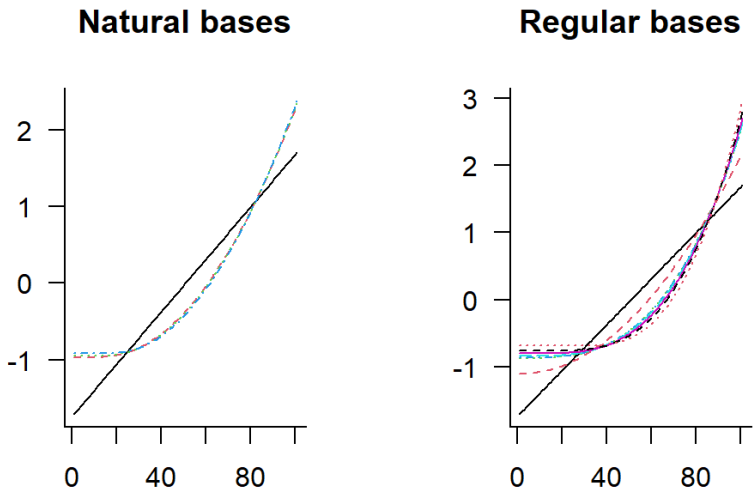

Figure 5: The comparison of two bases on a random interval with five knots. Codes available on Github

As expected, we observed a decrease of 4 in the degree of freedom of the natural spline bases.

4 Question 4

For this question, we create a basic function that simulates data from a high-order bivariate polynomial on the unit square, with the addition of Gaussian noise. The simplest possible scenario would be to use a 2D unit circle as the polynomial.

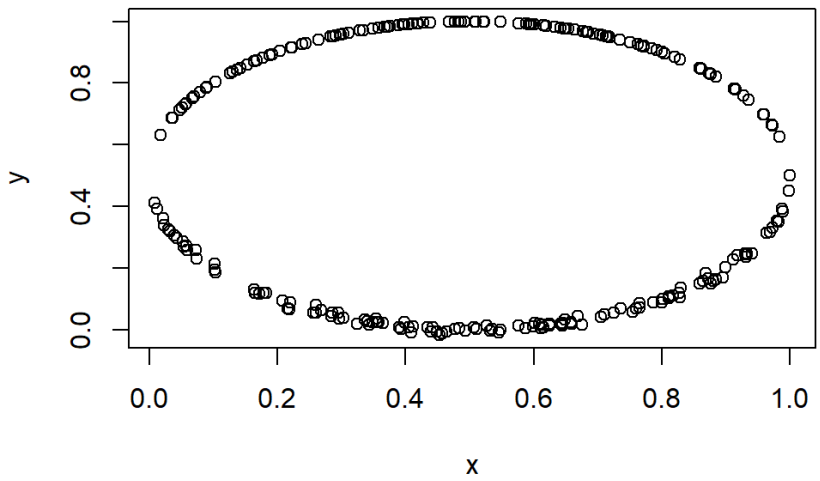

Figure 6: The simulation data points from the function.Codes available on Github

To fit the two-dimensional splines, we generate 300 data points in each iteration. We repeat this process 250 times and compute the average computation time in milliseconds using the microbenchmark() function.

We use residuals to represent the bias, and we calculate the mean squared error as the variance divided by the degrees of freedom for the residuals. The resulting data is presented in Table 1.

	GCV	REML
Time	29.71263	181.3016
Bias	-1.536085e-14	-2.606682e-14
Variance	0.1673443	0.1673443
Mean squared error	0.0007206342	0.0007166191

Table 1: The outcomes of different methods, GCV and REML to fit the simulation data.

5 Question ESL 5.4

This part followed the work provided by here.

Suppose that $N_1, .., N_k$ is a basis for K knots of the natural truncated polynomial splines. It is important to recall that the natural constraint for these splines is that "the function is linear beyond the boundary knots".

Given knots ξ_k, we have

$$f(X) = \sum_{j=0}^{3} \beta_j x^j + \sum_{k=1}^{K} \theta_k (x - \xi_k)_+^3.$$

When $x < \xi_1$, we have

$$f(x) = \sum_{j=0}^{3} \beta_j x^j$$
$$f''(x) = 2\beta_2 + 6\beta_3 x.$$

In order to fulfill the constraint, we make $f''(x) = 0$, which leads to $2\beta_2 + 6\beta_3 x = 0$, for $x < \xi_1$. Therefore, this achieves only if $\beta_2 = 0, \beta_3 = 0$. When $x \geq \xi_1, \beta_2 = 0, \beta_3 = 0$, we get:

$$f(X) = \sum_{j=0}^{1} \beta_j x^j + \sum_{k=1}^{K} \theta_k (x - \xi_k)^3$$
$$f''(x) = 6 \sum_{k=1}^{K} \theta_k (x - \xi_k).$$

When $f''(X) = 0$, we find that $(\sum_{k=1}^{K} \theta_k)x - \sum_{k=1}^{K} \theta_k \xi_k = 0$ for $x \geq \xi_k$. Thus, we conclude $\sum_{k=1}^{K} \theta_k = 0$ and $\sum_{k=1}^{K} \theta_k \xi_k = 0$.

6 Question ESL 5.13

This solution mostly refers to here.

We know that:

$$\hat{f}_\lambda = argmin \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt$$

By refit the $(x_0, \hat{f}_\lambda(x_0)$, now the $\hat{f}_\lambda$ turns out to be:

$$\hat{f}_\lambda = argmin \sum_{i=1}^{N}(y_i - f(x_i))^2 + (f(x_0) - f(x_0))^2\lambda \int (f''(t))^2 dt$$

Thus we have:

$$\begin{pmatrix} \hat{f}_\lambda^{(-i)}(x_1) \\ ... \\ \hat{f}_\lambda^{(-i)}(x_i) \\ ... \\ \hat{f}_\lambda^{(-i)}(x_N) \end{pmatrix} = S_\lambda \begin{pmatrix} y_1 \\ ... \\ \hat{f}_\lambda^{(-i)}(x_i) \\ ... \\ y_N \end{pmatrix}$$

and

$$\hat{f}_\lambda = S_\lambda y$$

Then,

$$\begin{aligned} \hat{f}_\lambda^{(-i)}(x_i) &= \sum_{j=1}^{N} S_\lambda(i,j) y_j \\ &= \sum_{i \neq j} S_\lambda(i,j) y_j + S_\lambda(i,i) \hat{f}_\lambda^{(-i)}(x_i) \end{aligned} \tag{1}$$

$$\begin{aligned} f_\lambda(x_i) &= \sum_{j=1}^{N} S_\lambda(i,j) y_j \\ &= \sum_{i \neq j} S_\lambda(i,j) y_j + S_\lambda(i,i) y_i \end{aligned} \tag{2}$$

By equation 2 - equation 1, we obtain:

$$\begin{aligned} f_\lambda(x_i) - \hat{f}_\lambda^{(-i)}(x_i) &= S_\lambda(i,i)(y_i - \hat{f}_\lambda(x_i)) \\ (1 - S_\lambda(i,i))\hat{f}_\lambda(x_i) &= f_\lambda(x_i) - S_\lambda(i,i)(y_i) \end{aligned}$$

Hence we get:

$$\begin{aligned} y_i - \hat{f}_\lambda^{(-i)}(x_i) &= y_i - \frac{f_\lambda(x_i) - S_\lambda(i,i)(y_i)}{1 - S_\lambda(i,i)} \\ &= \frac{y_i - f_\lambda(x_i)}{1 - S_\lambda(i,i)} \end{aligned}$$

which is exactly the N-fold cross-validation formula.

References

[1] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.