NONPARAPMETRIC DATA ANALYSIS

ADDITIVE MODEL FOR KERNEL AND SPLINE REGRESSIONS

Wenchen Guo

Contents

I NONPARAMETRIC RREGRESSION

1.1 Data

Below is the first few lines for the data set with one y variable and 3 x-variables. There are 174 observations in this data set. We'll use kernel and spline regressions to fit this data set.

order	y	x1	x2	x3
1	3.80	6934.50	283453	6.10
2	3.50	6979.10	283696	6.00
3	4.00	7009.80	283920	5.80
4	4.10	7029.30	284137	6.10
5	4.50	7022.10	284350	6.60
6	4.30	7036.20	284581	5.90
7	3.70	7083.10	284810	6.30
8	3.70	7097.10	285062	6.00
9	5.00	7109.20	285309	6.80
10	6.10	7146.10	285570	6.90

1.2 Regression Fitting

Go through one-variable-at-a-time kernel- and spline-regression fits to the data for all 3 x-variables. Compare the fitting results using the leave-one-out cross-validation procedure.
1st Step: Fit $y \sim x_1$ and call this $f_1(x_1)$.
2nd Step: Fit $e_1 = y - f_1(x_1) \sim x_2$ and call this $f_2(x_2)$.
3rd Step: Fit $e_2 = y - f_1(x_1) - f_2(x_2) \sim x_3$ and call this $f_3(x_3)$.
The final model is

$$y = f_1(x_1) + f_2(x_2) + f_3(x_3) + \epsilon$$

In this section, we'll use both kernel and spline regressions to fit $f_1(x_1), f_2(x_2), f_3(x_3)$.

1.2.1 Kernel Regression

The range for x_1 is [6934.5,12161.5], the range for x_2 is [283453,320887], and the range for x_3 is [5.8,25.2].
Figure 1 shows the cross-validation results for choosing bandwidth for using kernel to fit

$f_1(x_1), f_2(x_2), f_3(x_3)$. The bandwidth b_1 choosen from cross-validation procedure for $f_1(x_1)$ is

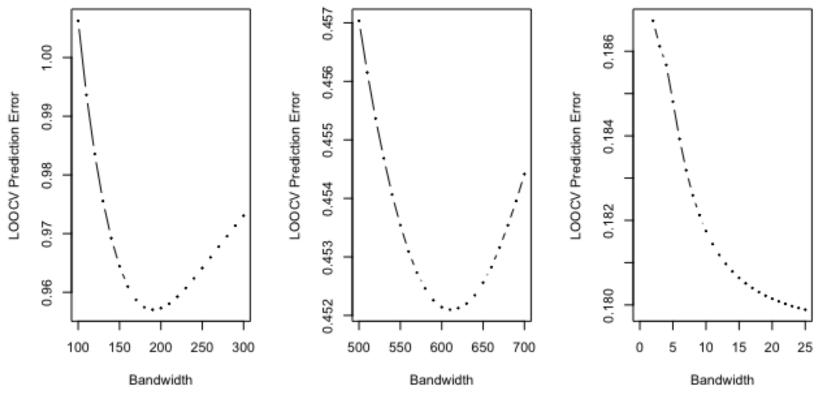

Figure 1: Bandwidth Selection

190. The bandwidth b_2 choosen from cross-validation procedure for $f_2(x_2)$ is 610. However, the MSE decrease when bandwidth b_3 increase.

Figure 2 shows the fitting results and residual plots for $f_1(x_1), f_2(x_2), f_3(x_3)$. The first row are

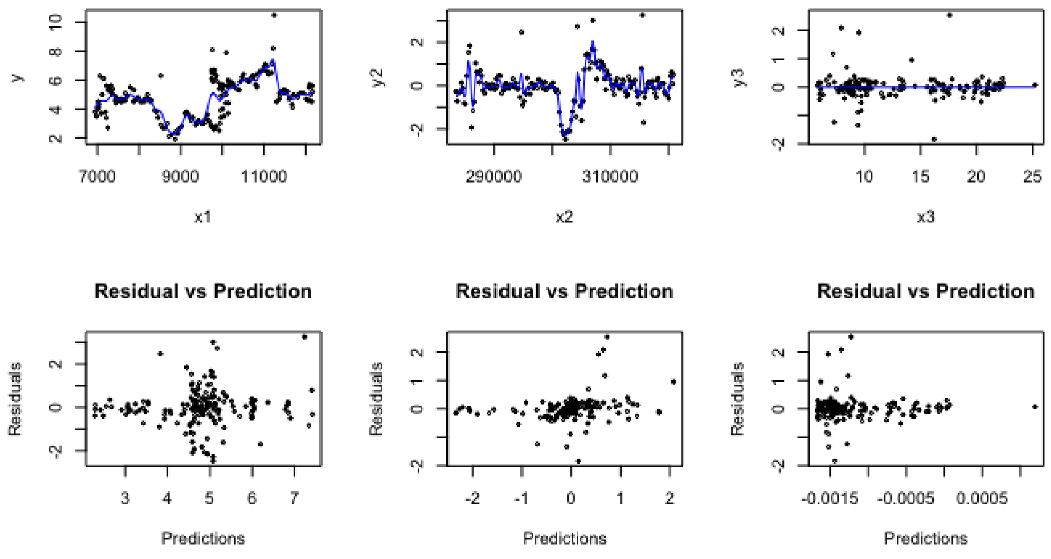

Figure 2: Kernel Regression

the kernel regression fitting for $f_1(x_1), f_2(x_2), f_3(x_3)$. The black points are the plot for data,

and the blue lines are kernel regression lines. The second row are residual plots.
From this two figures we can see that, the kernel fitting for $f_1(x_1), f_2(x_2)$ are good. However, we cannot get the best bandwidth from cross-validation procedure, and the kernel fitting for $f_3(x_3)$ approaches a horizontal line. This is because in step 2, get $f_2(x_2)$ by fitting the residual with x_2, the residuals are all quite small and approach i.id. Doing the 3rd step of getting $f_3(x_3)$ cannot improve our model anymore.
Thus, our final model for kernel regression fitting is

$$y = f_1(x_1) + f_2(x_2) + \epsilon$$

The bandwidth for $f_1(x_1)$ is 190, and the bandwidth for $f_2(x_2)$ is 610.

1.2.2 Spline Regression

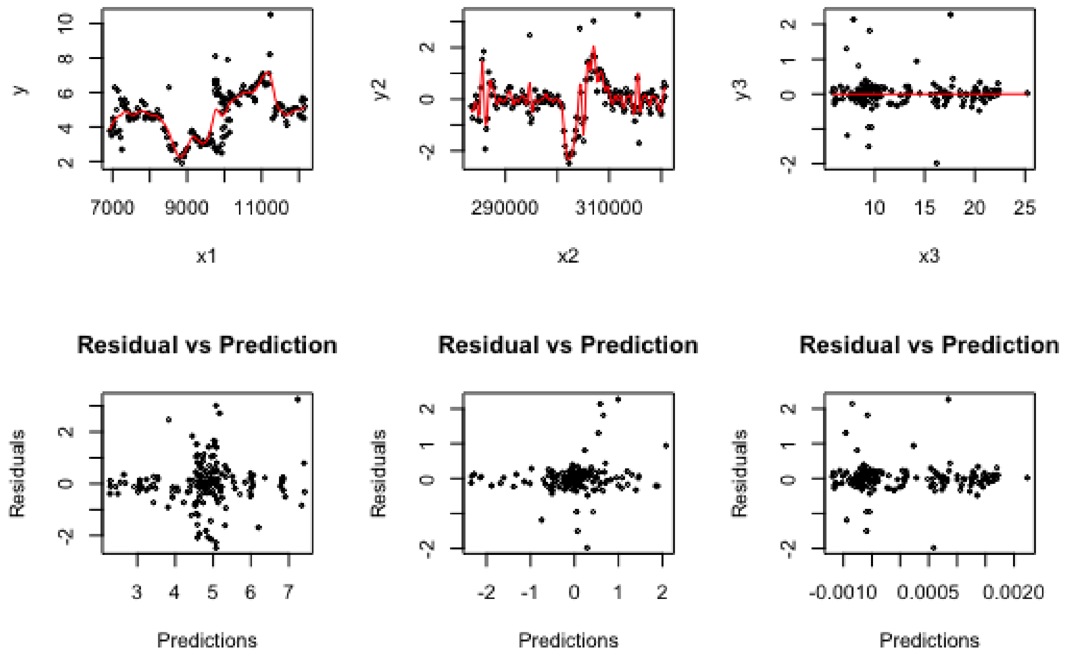

Figure 3: Spline Regression

R function *smooth.spline* can specified "cv=T" for choosing the best smooth parameter λ by the cross-validation procedures. For our data, the spline functions $f_1(x_1), f_2(x_2), f_3(x_3)$, $\lambda_1 = 0.5946$, $\lambda_2 = 0.2146$, $\lambda_3 = 1.4999$.

Figure 3 shows the fitting results and residual plots for $f_1(x_1), f_2(x_2), f_3(x_3)$.
The first row are the spline regression fitting for $f_1(x_1), f_2(x_2), f_3(x_3)$. The black points are the plot for data, and the red lines are spline regression lines. The second row are residual plots. Here, we got similar result as the kernel regression fitting. The spline fitting for $f_1(x_1), f_2(x_2)$ are good. However, the spline fitting for $f_3(x_3)$ approaches a horizontal line. Doing the 3rd step of getting $f_3(x_3)$ cannot improve our model anymore.
Thus, our final model for spline regression fitting is

$$y = f_1(x_1) + f_2(x_2) + \epsilon$$

The smoothing parameter for $f_1(x_1)$ is 0.5946, and the smoothing parameter for $f_2(x_2)$ is 0.2146.

1.2.3 Comparison

Figure 4 shows the fitting results for $f_1(x_1), f_2(x_2)$. The blue lines are kernel fitting and the red lines are spline fitting. We can concluded that, both kernel and spline povide evidence that

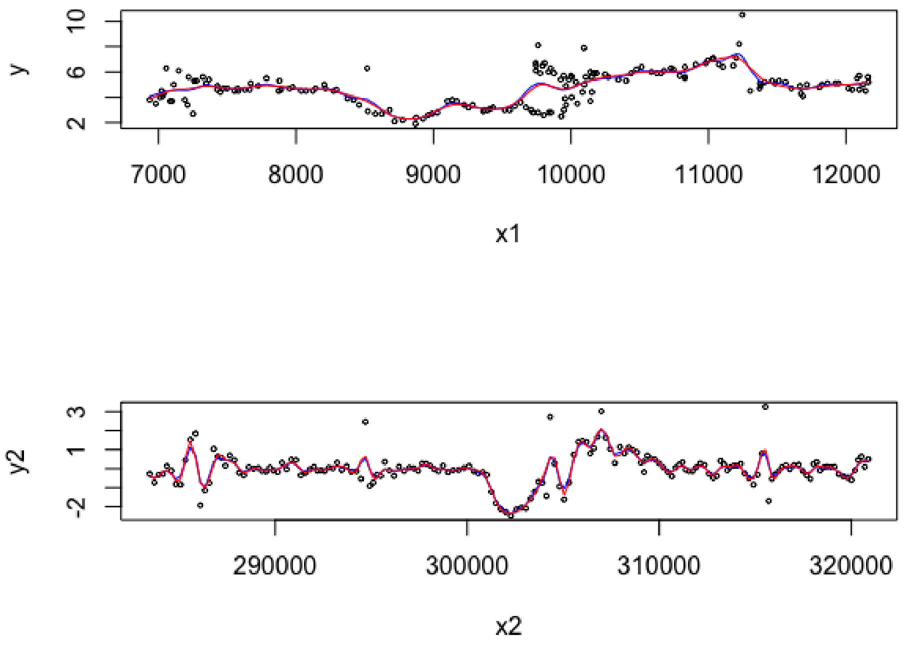

Figure 4: Kernel & Spline Regression

we do not need to include x_3 in our final model. And the fitting results for kernel regression and spline regression are very similar.

1.3 Prediction

The range for x_1 is [6934.5,12161.5], the range for x_2 is [283453,320887]. We'll pick one point $point_1$ that locates close to the center ($x_1 = 9962$, $x_2 = 307680$) and one point $point_2$ ($x_1 = 7055$, $x_2 = 285840$) that locates close to the edge.

Make predictions of Y at these x-data. Compare the predictions from Kernel and Spline methods, and also comment on the impact from x-data-locations.

	Center Point	**Side Point**
Kernel	5.368916	5.152286
Spline	5.316878	4.971514

Find two points in the data set that close to the points we pick:

	x1	**x2**	**Y**
center	9961.9	307685	4.9
side	7083.1	284810	3.7

By comparing the y value we can see, kernel and spline regression fits better in the center than on the side. The prediction errors are larger for the edge data.

1.4 Bootstrap

We can construct the 90% CI with 10000 bootstrapped predictions for two points with kernel and spline regression, then use

$$[LCL, UCL] = [\hat{y} - 1.645se(\hat{y}), \hat{y} - 1.645se(\hat{y})]$$

to get the normal 90% bootstrap CI.

The Acceleration and Bias-Correction (or BCa) method improves on the percentile method by adjusting the percentiles ($\tilde{\theta}(1-\alpha/2)$,$\tilde{\theta}(\alpha/2)$) chosen from the bootstrap sample above.

The bias factor is

$$z_0 = \phi^{-1}(p_0)$$

where

$$p_0 = B^{-1} \sum I(\tilde{\theta}_i < \theta_n)$$

indicates the proportion of the bootstrap estimaters $\tilde{\theta}$ less than θ_n.
The acceleration factor measures the rate of change in σ_{θ_n} as a function of θ.

$$a_0 = \frac{\sum_{i=1}^{B} (\tilde{\theta}^* - \tilde{\theta}_i)^3}{6\left(\sum_{i=1}^{B} \left(\tilde{\theta}^* - \tilde{\theta}_i\right)^2\right)^{3/2}}$$

where $\tilde{\theta}^*$ is the average of the bootstrap estimaters.
The $100(1-\alpha)\%$ BCa interval is

$$\left[\tilde{\theta}(q1),\, \tilde{\theta}(q2)\right]$$

where

$$q1 = \phi\left(z_0 + \frac{z_0 + z_{\alpha/2}}{1 - a_0(z_0 + z_{\alpha/2})}\right)$$

$$q2 = \phi\left(z_0 + \frac{z_0 + z_{1-\alpha/2}}{1 - a_0(z_0 + z_{1-\alpha/2})}\right)$$

Here, θ_n are the 4 prediction results shown in section 1.3.
And $\tilde{\theta}_i$ is the i_{th} bootstrap estimators. $\tilde{\theta}^*$ is the mean of the 10000 bootstrap estimators.
Then we can also get the Bias-Corrected CIs (R code for bias-corrected method is in Appendix).

	Bootstrap CI		**Bias-Corrected CI**	
	Kernel	**Spline**	**Kernel**	**Spline**
Center Point	[4.5364, 6.5780]	[4.4635, 6.8207]	[1.9130%, 88.7305%]	[0.8453%, 81.5572%]
			[4.7004, 6.2849]	[4.3364, 6.4847]
Side Point	[2.6439, 6.7464]	[2.1642, 6.9229]	[4.5611%, 94.5517%]	[7.0560%, 96.5779%]
			[2.5475, 6.2931]	[2.2414, 6.2868]

From the table we can see, the CIs are wider for side point than center point. By using the bias-corrected method, the CIs are impoved by become narrow.