Statistics - Week 4

Jimmy Tsz Ming Yue[*]

University of Sydney

jyue6728@uni.sydney.edu.au

Semester 2 Statistics 2018

Contents

1 The Linear Regression Model

Definition. A linear regression model is described as:

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{1}$$

Where:

1. X is predictor (independent variable) and Y is the response (dependent variable).
2. β_0 is the intercept of the regression line, (the expected value of Y when $X = 0$)

[*]440159151

3. β_1 is the slope of the regression line (the average increase in Y associated with a one-unit increase in X)
4. ϵ is a residual (error); random with zero mean and finite variance.

There are multiple ways to determine ”optimal” line through data points. An example of this is the least squares regression:

1.1 Least Squares Regression Line

Definition. The residual sum of squares is defined to be:

$$\begin{aligned}\hat{\text{RSS}} &= \sum_{i=1}^{n} e_i^2 \\ &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2\end{aligned}$$

Then the following theorem defines the least squares regression:

Theorem The least squares approach minimises the RSS by choosing $\hat{\beta}_0$ and $\hat{\beta}_1$ such that:

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$$

with:

$$\begin{aligned}\hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \\ &= \frac{\text{Cov}(x, y)}{\text{Var}(x)}\end{aligned}$$

with $\bar{y}$ and $\bar{x}$ defined as:

$$\begin{aligned}\bar{y} &= \frac{1}{n}\sum_{i=1}^{n} y_i \\ \bar{x} &= \frac{1}{n}\sum_{i=1}^{n} x_i\end{aligned}$$

1.2 Linear Regression Fit

Linear regression often fits well to data that have linear relationships. Various theoretical analysis of linear regression fit are available for further inference of the data and the relationship of predictor and response variables.

2 Smoothing

With predictor-response data, the random response varialbe Y is assume to be a stochastic function of the value of a predictor variable X.

Definition. A typical model for predicto-response data is:

$$Y_i = s(x_i) + \epsilon_i \tag{2}$$

where:

1. ϵ_i are zero-mean stochastic noise and
2. s is a smooth function

The conditional distribution of $Y|X$ describes how Y depends on X. One sensible smooth curve through the data would connect the conditional means of $Y|X$ for the range of predictor values observed $(x_1, \ldots, x_n)$

2.1 Local Averaging

Most smoothers (smoothing function) rely on the concept of local averaging. The Y_i whose corresponding x_i are near x should be averaged in some way to gleam information about the approrpiate value of the smooth at x.

Definition. A generic local smoother can be written as:

$$\hat{s}(x) = \text{avg}\left\{Y_i | x_i \in \mathcal{N}(x)\right\} \tag{3}$$

for

1. some generalised average function "avg" and;
2. some neighbourhood of x say $\mathcal{N}(x)$

2.2 Smoothers

Different smoothing function results from different choices for the averaging function. for example we may have means, weighted means, medians or M-estimates. and the neighbourhood (e.g, the nearest few neighbouring points, or all points within some distance.) The form of $\mathcal{N}(x)$ may even vary with x so that different neighbourhood sizes or shapes may be used in different regions of the dataset.

2.3 Span

The most important characteristic of a nieghbourhood is its span, which is represented by the smoothing parameter λ. The span of a neighbourhood measures its inclusiveness, that means that neighbourhood with small spans are locally strong, including only very nearby points, whereas neighbourhoods with large spans have a wide membership.

There are ways to measure a neighbourhood's inclusiveness such as:

1. size: which refers to the number of points

2. span which refers to the proportion of sample points that are members
3. bandwidth, referring to the physical length or value of the neighbourhood

2.4 Constant- Span Running Mean

Let us consider a simple smoother, which takes the sample mean of k nearby points. Let us then define the symmetric nearest neighbourhood:

Definition. Symmetric Nearest Neighbourhood $\mathcal{N}(x_i)$ as the collection of x_i, with the $(k-1)/2$ points whose predictor values are nearest below x_i and the $(k-1)/2$ points whose predictor values are above x_i

The smoother who uses this neighbourhood set, is called a moving average or a k-nearest neighbours smoother. Then let us assume without the loss of generality, assume that data pairs have been sorted so that the x_i are in increasing order. Then we can write the constant-spam running-mean smoother outlined above as;

$$\hat{s_k}(x_i) = \text{mean}\left\{Y_j \text{ for } \max\left(i - \frac{k-1}{2}, 1\right) \leq j \leq \min\left(i + \frac{k-1}{2}, n\right)\right\} \tag{4}$$

For the purposes of graphing or prediction, one can compute $\hat{s}$ at each of the x_i and interpolate linearly in between.

2.5 Effect of Span

A natural smoothing parameter for the constant-span running-mean is $\lambda = k$. As with all smoothers, the parameter controls the degree of wiggliness. For this current case, the parameter is controlling the number of data points contained in any neighbourhood.

3 Performance of Smoothers

Let us consider a generic example, For a given point x, let $\hat{s}(x)$ be an estimator of $s(x)$. Which estimator is best. We can assess the quality of $\hat{s}$ as an estimator of $s(x)$ at x using mean squared error at x:

$$\text{MSE}(\hat{s}(x)) = E\left\{[\hat{s}(x) - s(x)]^2\right\} \tag{5}$$

Which can be further decomposed to bias and variance

$$= (\text{bias}\{\hat{s}(x)\})^2 + \text{var}\{\hat{s}(x)\} \tag{6}$$

The best choice for k must balance a trade-off between bias and variance.

1. For small k, the estimated curve will be wiggly but exhibit more flexibility to fit the data.
2. For large k, the estimated curve will be smooth but exhibit substantial bias in some regions.

The role of the smoothing parameter is to control this tradeoff between bias and variance. One way to select best k is to use leave one out cross validation which gives cross-validated residual sum of squares $\text{CVRSS}_k(\hat{s}_k)$

$$\frac{\text{CVRSS}_k(\hat{s}_k)}{n} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{s}_k^{(-i)}(x_i)\right)^2 \tag{7}$$

Typically we plot CVRSS_k is plotted against k .

4 Running Lines and Running Polynomials

The constant-span running-mean smoother exhibits visually unappealing wiggliness for any reasonable k. It also can have strong bias at the edges because it fails to recognise the local trend in the data. As such we consider another smoother which can mitigate both problems. This is the running-line smoother. Consider fitting a linear regression model to the k data points in $\mathcal{N}(x)_i$. Then the least squares linear regression prediction at x is:

$$l_i(x) = \bar{Y}_i + \hat{\beta}_i(x - \bar{x}_i) \tag{8}$$

where $\bar{Y}_i,\ \bar{x}_i),\ \hat{\beta}_i.$ are the mean response, the mean predictor and the estimated slope of the regression line, respectively, for the data in $\mathcal{N}(x_i)$. The running line smooth at x_i is $\hat{s}_k(x_i) = l_i(x_i)$

5 Kernel Smoothers

For the smoothers mentioned so far, there is a discontinuous change to the fit each time the neighbourhood membership changes. Therefore, they tend to fit well statistically but exhibit visually unappealing jitters or wiggles. One approach to increasing smoothness is to redefine the neighbourhood so that points gradually gain or lose membership in it. Let K be a symmetric kernel centred at 0. A kernel is essentially a weighting function, which here we weight neighbourhood membership. One reasonable kernel choice would be the standard normal density,

$$K(z) = \left(\frac{1}{\sqrt{2\pi}}\right) e^{-\frac{z^2}{2}} \tag{9}$$

Then for such a kernel;

$$\hat{(s)}_h = \sum_{i=1}^{n} Y_i \frac{K\left((x - x_i)/h\right)}{\sum_{i=1}^{n} K\left((x - x_j)/h\right)} \tag{10}$$

where the smoothing parameter h is called the bandwidth. Notice that for many common kernels such as the normal kernel, all data points are used to calculate the smoothness at each point, but there is a downside in that very distant data points receive very little weight.

5.1 Distant data points recieve very little weight

Proximity increases a point's influence on the local fit; in this sense the concept of local averaging remains.

1. A large bandwidth yields a quite smooth result because the weightings of the data points change little across the range of the smooth .

2. A small bandwidth ensures a much greater dominance of nearby points, thus producing more wiggles.

The choice of smoothing kernel is much less important than the choice of bandwidth and there are few reasons to look beyoond a normal kernel .

5.2 Normal Kernel Smoother

Since neighbourhood entries and exits are gradiual, the result exhibits characteristcally rounded features. It should be noted to the reader that kernel smoothing does not eliminate systematic bias at the edges, as the running-line smoothing does.

6 Spline Smoothing

Assume that the data have been sorted in increasing order of the predictor, so x_1 is the smallest predictor value, x_n is the largest. Let us define:

Definition. Penalty:

$$Q_\lambda(\hat{s}) = \sum_{i=1}^{N} (Y_i - \hat{s}(x_i))^2 + \lambda \int_{x_1}^{x_n} \hat{s}''(x)^2 dx \tag{11}$$

where

$$\hat{s}''(x) = \frac{d^2\hat{s}}{dx^2} \tag{12}$$

where the penalty for wiggliness is given by the integral and the penalty for misfitting is given by the summation. λ controls the relative weighting of the two penalties.

Minimising $Q_\lambda(\hat{s})$ leads to a cubic smoothing spline This function is a cubic polynomial in each interval $[x_i, x_i + 1]]\,, \forall i = 1, \ldots n-1$ with the resultant function twice continously differentiable at each x_i

7 Nonlinear smoothers

Non linear smoothers can be much slower to calculate, and in ordinary cases they offer little improvement over simpler approaches. However the simpler methods can exhibit very poor peforamnces for some types of data. The loess smoother provides improved robustness to outliers that would introduce substantial noise in an ordinary smoother.

7.1 Loess

Loess is a Locally weighted scatterplot smoothing method. It is essentially a weighted running-line smoother, except that each local line is fitted using a robust method rather than least squares. As a result the smoother is nonlinear. Loess is fitted itteratively:

Definition. LOESSa

$$K_i(x) = K\left(\frac{x - x_i}{d_k(x_i)}\right)$$

where:

$$K(z) = \begin{cases} (1 - |z|^3)^3 & |z| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Then the smoother is:

$$\sum_{j=1}^{N} \left(Y_j - \left(\beta_{0,i}^{(t)} + \beta_{1,i}^{(t)} x_j\right)\right)^2 K_i(x_j)$$

8 Tutorial

The “datasmooth.txt” contains <x, y> pairs generated from function(x)$\{(x^3) * cos((x + 1.4)/2)\}$ with random noise.

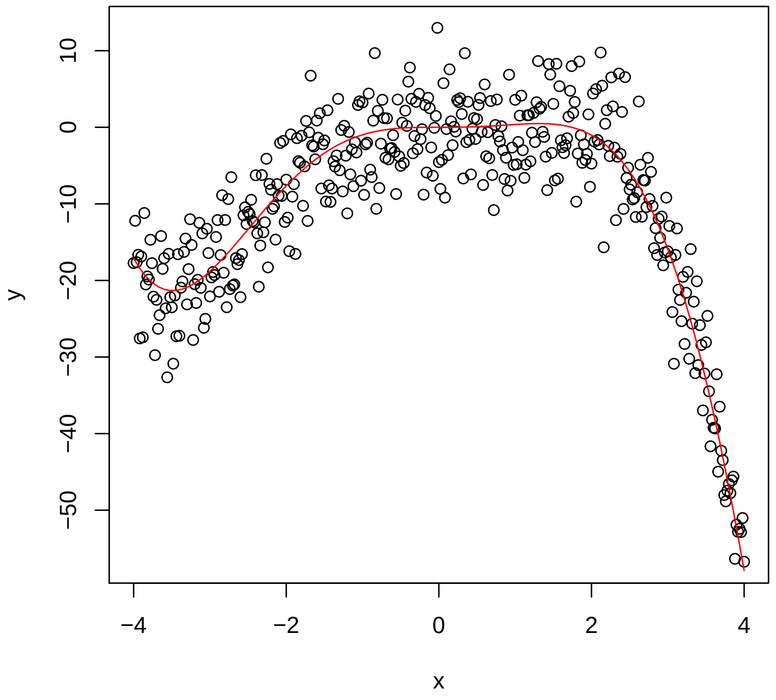
1. Create a scatter plot of the data and overlay the true relationship line on the plot.

Solution. Let us import the datasmooth dataset;

```
> data <- read.table("datasmooth.txt", header=TRUE, sep = "\t")
> head(data)

      x         y
1 -4.00 -17.73607
2 -3.98 -12.20970
3 -3.96 -17.61862
4 -3.94 -16.64749
5 -3.92 -27.56882
6 -3.90 -16.87353

> plot(data)
> eq = function(x){(x^3)*cos((x+1.4)/2)}
> curve(eq, add = TRUE, col = "red")
```

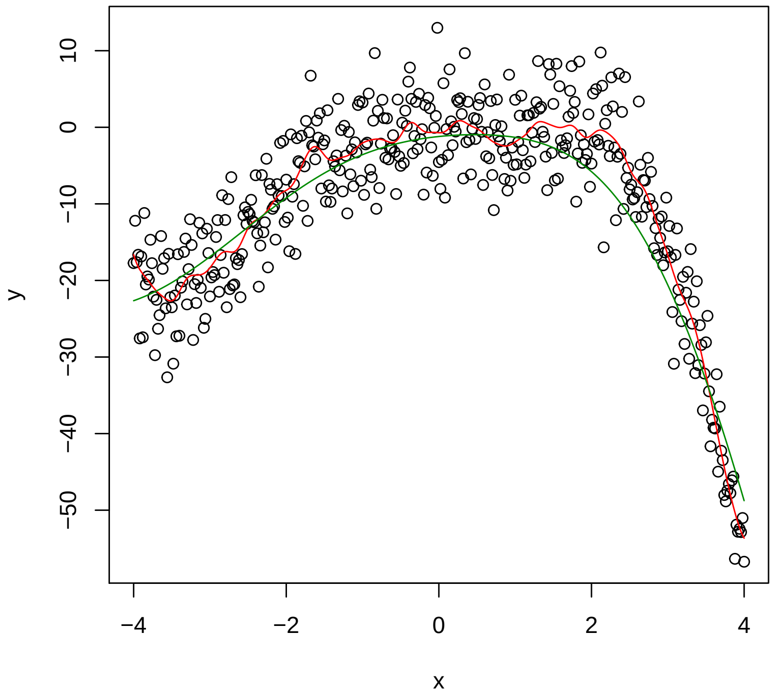


□

2. Try normal kernel smoothing with different bandwidths on “datasmooth.txt”.

Solution. Let us first import the Kernel Smoothing package

```
> library(KernSmooth)
> x <- data$x
> y <- data$y
> fit1 <- locpoly(x, y, kernel="normal", bandwidth=0.1)
> fit2 <- locpoly(x, y, kernel="normal", bandwidth=0.9)
> plot(data)
> lines(fit1, col = "red")
> lines(fit2, col = "green4")
```



□

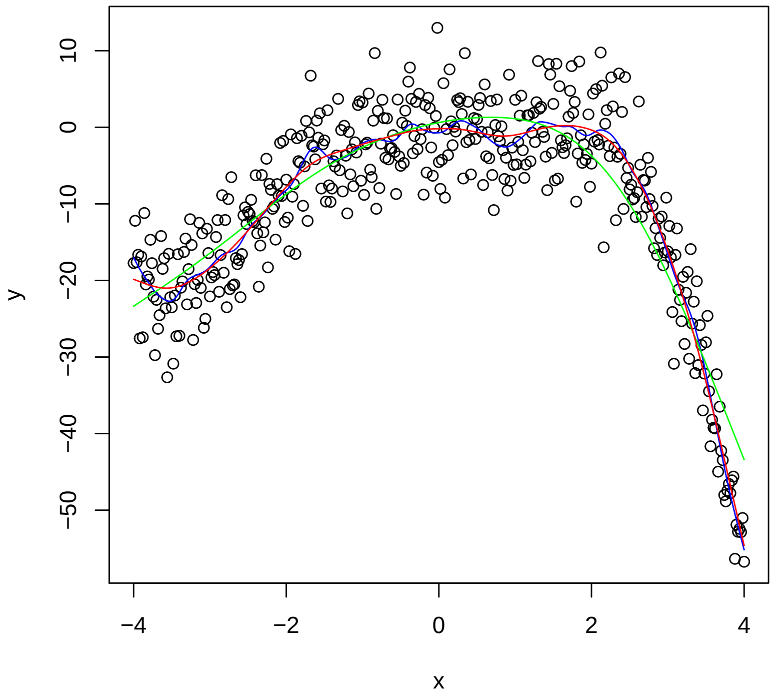
3. Apply cubic spline with different spars on “datasmooth.txt”.

Solution. Let us make use of the R cubic spline;

```
> plot(data)
> cubicSpline1.fit <- smooth.spline(x=x, y=y, cv=FALSE, spar=0.5)
> cubicSpline2.fit <- smooth.spline(x=x, y=y, cv=FALSE, spar=1)
> cubicSpline3.fit <- smooth.spline(x=x, y=y, cv=TRUE)
> lines(cubicSpline1.fit, col = "blue")
> lines(cubicSpline2.fit, col = "green")
> lines(cubicSpline3.fit, col = "red")
> cubicSpline3.fit

Call:
smooth.spline(x = x, y = y, cv = TRUE)

Smoothing Parameter  spar= 0.7581144  lambda= 0.0005645639 (13 iterations)
Equivalent Degrees of Freedom (Df): 11.27981
Penalized Criterion (RSS): 9036.802
PRESS(l.o.o. CV): 23.83662
```



□

4. "newDatasmooth.txt" contains the <x, y> pairs generated from the same function. Utilise this new dataset to estimate mean squared error and select best bandwidths and spars for kernel smoother and cubic spline, respectively.

Solution. Let us first import the new data set:

```
> newdata <- read.table("newDatasmooth.txt", header=TRUE, sep = "\t")
> head(newdata)

      x         y
1 -4.00 -12.61164
2 -3.96 -15.17901
3 -3.92 -23.28943
4 -3.88 -20.42623
5 -3.84 -14.98288
6 -3.80 -22.82512

> xn <- newdata$x
> yn <- newdata$y
> optimBw <- dpill(xn, yn)
> optimBw
```

```
[1] 0.3034217

> cubicSpline3.fit

Call:
smooth.spline(x = x, y = y, cv = TRUE)

Smoothing Parameter  spar= 0.7581144  lambda= 0.0005645639 (13 iterations)
Equivalent Degrees of Freedom (Df): 11.27981
Penalized Criterion (RSS): 9036.802
PRESS(l.o.o. CV): 23.83662
```

Therefore the optimal bandwidth for the gaussian smoother is 0.303 and the 0.76 respectively. □

5. Normal kernel smoother or cubic spline fits better to "datasmooth.txt"?

Solution. Cubic spline is better in this case more closely matching the true curve.

□