pg 315 Chapter 11. Bivariate Smoothing

Look at Fig 11.1 pg 316. I bet you could draw a smooth curve that fits the data pretty well. How did you do it?

Suppose we have n bivariate data points (x_i, y_i), $i=1,\ldots,n$. Also, suppose that it is predictor-response data, that is the random response Y is assumed to be a (stochastic most likely) function of the value of the predictor variable X.

Ex/ $Y_i = s(X_i) + \varepsilon_i$ where ε_i are mean-zero stochastic noise and s is a smooth function.

Then the conditional dist $Y|X$ describes how Y depends on $X=x$. A possible choice for smoothing is the smooth curve through the data connecting the conditional means of $Y|X$.

We will focus on smoothing pred-resp. data.

If not p-r. data (Sec 11.6). i.e. there is no clear distinction between X_1, X_2 (X_1, X_2). It does not do to simply set one as pred & one as response. (See example pg 342).

So, the methods in this Chapter will most likely fail.

Sec 11.1 Predictor - Response Data

Suppose we have P-R data (x_i, y_i). and suppose $E[Y|X] = s(x)$ for a smooth function s. Then the goal of smoothing is to estimate s → sometimes called nonparametric regression.

move to ⊛

Now, for a given point x let an estimate be $\hat{s}(x)$ ⊛. How do we know if our est is good? Most commonly. we use

$$MSE_\lambda(\hat{s}_\lambda(x)) = E[(\hat{s}_\lambda(x) - s(x))^2]$$
$$= (\text{bias}(\hat{s}_\lambda(x)))^2 + \text{var}(\hat{s}_\lambda(x)). \quad \text{(pointwise)}.$$

⊛ Now, usually the smoother $\hat{s}(x)$ is based not only on the obs. data (x_i, y_i) but also on a user specified smoothing parameter λ, whose value is chosen to control the overall behavior of the smoother. So we write $\hat{S}_\lambda(x)$

Also, if we consider a new point x^* and we want to predict the value $s(x^*)$, we can asses the quality of $\hat{s}_\lambda(x^*)$ as an estimator of $s(x^*) = E[Y|X=x^*]$ by the Mean Squared Prediction Error at x^*.

$$MSPE_\lambda(\hat{s}_\lambda(x^*)) = E[(Y - \hat{s}_\lambda(x^*))^2 \mid X = x^*]$$
$$= \text{var}(Y|X=x^*) + MSE_\lambda(\hat{S}_\lambda(x^*)).$$

Then this can be averaged to give a global measure of the quality of the smooth. These are the measures we will use to assess $\hat{S}(x)$.

How do we construct good smoothers?

Basic Idea: Want the smoother to summarize the conditional distribution of Y_i given $X_i = x_i$ by some measure of location → like cond mean "going through the center of the data points".

So smoothers rely on the idea of local averaging → the Y_i whose corresp. x_i are near x should be averaged in some way to glean info about the approp value of the smooth at x.

Generically: $\hat{S}(x) = \text{ave}\{Y_i \mid x_i \in N(x)\}$
↳ neighborhood of x.

& so diff smoothers are obtained by diff choices of "ave" and "neighborhoods".
- ave – mean, median, wt mean etc.
- Neighborhood – # of neighbors, distance etc.

The parameter λ most commonly represents the span of the neighborhood (size, prop, b w) & indicates a measure of inclusiveness. & how heavily the smoother relies on local points.
small λ → local → higher variance
large λ → distant points → introduce bias.
Keep this in mind when forming estimators.

We will look at strategies for constructing local averaging smoothers.

Sec 11.2 Linear Smoothers

→ The prediction at any point x is a linear combination of the response values. (We focus on estimating the smooth at obs x_i). So given $X = (x_1, \dots, x_n)^T$ and $Y = (Y_1, \dots, Y_n)^T$, then $\hat{s} = (\hat{s}(x_1), \dots, \hat{s}(x_n))^T$ can be expressed as

* and then interpolating for prediction.

$$\hat{s} = SY$$

where S is an $n \times n$ smoothing matrix that does not depend on Y. These linear smoothers are faster to compute & easier to analyze than non-linear smoothers.

11.2.1 Constant span running mean (a.k.a moving average)

Idea: Take the sample mean of k nearby points.

$$\hat{s}_k(x_i) = \sum_{\{j:\, x_j \in N(x_i)\}} Y_j / k$$

~~Comments~~

(B) If k is odd, then $N(x_i)$ is x_i along with $(k-1)/2$ values nearest below & above. & is called sym. nearest neighborhood.

So from here on out we assume sorted x_i, & odd k. then

$$\hat{s}_k(x_i) = \text{mean}\left[Y_j \text{ for } \max\left(i-\tfrac{k-1}{2}, 1\right) \le j \le \min\left(i+\tfrac{k-1}{2}, n\right)\right]$$

and can be computed by stepping through i in order recursively.

$$\hat{S}_K(x_{i+1}) = \hat{S}_K(x_i) - \frac{Y_{i-(K-1)/2}}{K} + \frac{Y_{i+(K+1)/2}}{K}$$

being careful near the edges.

Is this a linear smoother? Yes w/ matrix S having middle rows $(0 \ldots 0\ \tfrac{1}{K} \cdots \tfrac{1}{K}\ 0 \cdots 0)$

? How to compute data near edges.

Possible options. $(K=5)$

① Shrink neighborhoods.

$$S = \begin{pmatrix} 1 & 0 & \cdots & & & & & \\ \tfrac{1}{3} & \tfrac{1}{3} & \tfrac{1}{3} & 0 & 0 & \cdots & & \\ \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & 0 & \cdots & \\ 0 & \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & 0 & \cdots \end{pmatrix}$$

② truncate neihborhoods

$$S = \begin{pmatrix} \tfrac{1}{3} & \tfrac{1}{3} & \tfrac{1}{3} & 0 & \cdots & & & \\ \tfrac{1}{4} & \tfrac{1}{4} & \tfrac{1}{4} & \tfrac{1}{4} & 0 & \cdots & & \\ \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & 0 & \cdots & \\ 0 & \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & \tfrac{1}{5} & 0 & \cdot \end{pmatrix}$$

Truncation preferred & as defined.

Ex/ pg 320

$n = 200$ & $Y_i = s(x_i) + \varepsilon_i$

$\varepsilon_i \sim N(0, 1.5^2)$

$$s(x) = x^3 \sin\left(\frac{x+3.4}{2}\right)$$

$K = 13.$

11.2.1.1 <u>Effect of span</u>

- Here $\lambda = k$. & for interior point we have

$$MSPE_k(\hat{S}_k(x_i)) = E\, var(Y \mid X = x_i) + MSE_k(\hat{S}_k(x_i))$$

which if we assume $Var(Y \mid X = x_i) = \sigma^2$ we can show that

$$MSPE_k(\hat{S}_k(x_i)) = (1 + 1/k)\sigma^2 + \left(bias(\hat{S}_k(x_i))\right)^2$$

and so as $k \uparrow$ decreases, but bias $\uparrow$.

Ex p 321. $K=3$ v/s $K=43$.

11.2.1.2 <u>How to select span for linear smoothers.</u>

- Want to balance var. w/ bias.
- Would be lovely if we could minimize $\overline{MSPE_k(\hat{S}_k)}$, but depends on unknowns.

Instead consider minimizing w.r.t k the residual mean sq. error

$$RSS_k(\hat{S}_k)/n = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{S}_k(x_i))^2$$

However

$$E\left[RSS_k(\hat{S}_k)/n\right] = \overline{MSPE_k(\hat{S}_k)} - \frac{1}{n}\sum_{i \neq j} cov(Y_i, \hat{S}_k(x_j))$$

and hence is biased.

$\therefore$ to eliminate bias, use cross validation.

$$CVRSS_k(S_k)/n = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{S}_k^{(-i)}(x_i)\right)^2$$

Where $\hat{S}_k^{(-i)}(x_i)$ is value of smooth when omitting (x_i, Y_i).

Typically a plot of the $CVRSS_k(\hat{S}_k)$ v/s K is viewed. Ex pg 323.

However Cross validation can be time consuming. To speed it up.

1. Leave out groups & not just 1.
2. Define

$$\hat{S}_k^{(-i)}(x_i) = \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{Y_j S_{ij}}{1-S_{ii}}$$

where S_{ij} is the (i,j)th element of S.

Recall original $\hat{S}_k(x_i) = \sum_{j=1}^{n} Y_j S_{ij}$ $\qquad S_{ij} \to 0$, $S_{ij} \to 1/k$

So, this is basically replacing the (i,i)th element of S with zero and rescaling the row so that it sums to 1 (ie re dist. the weight).

In this case for linear smoothers

$$CVRSS_k(\hat{S}_k)/n = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{S}_k(x_i)}{1-S_{ii}}\right)^2.$$

and is much easier to compute.

Sec 11.2.3 Kernel Smoothers

For the running mean smoothers there is a discontinuous change to the fit each time the neighborhood changes. ∴ they tend to fit well statistically, but have visually unapealing wiggles. → Instead, redefine the neighborhoods so that points only gradually enter or leave.

Instead, use a kenel function.

Let K be a symmetric kernel centered at zero. Ex/ N(0,1) $K(z) = \frac{1}{\sqrt{2\pi}} \exp\{-z^2/2\}$. Let h be our smoothing parameter & bandwidth of the kernel. Then we can define the smooth

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

Comments: ① In this case, K can be viewed as a weighting function, weighting neighborhood membership and impact.

② K can be chosen so that only some of the obs datapoints are in a Neighborhood, or more often (ex normal) all data points are used to calculate the ~~[illegible]~~ smooth at each point.

③ Retains concept of local averaging since proximity determines weight.

④ Large h → smoother, small h → wiggly.

⑤ As w/ estimation of densities, choice of K is not as important as choice of h & no real reason to go beyond using std. Normal.

⑥ These are linear smoothers. HW. Explain why & give S.

⑦ Can use CV to optimize b.w..

Example pg 328.

Normal h = .16

Sec 11.2.5 **Spline smoothing**

If so far our linear smoothers have been to wiggly → consider the following.

Assume the are sorted so that $x_1 < \cdots < x_n$. and define

$$Q_\lambda(\hat{S}) = \underbrace{\sum_{i=1}^{n} (Y_i - \hat{S}(x_i))^2}_{1} + \underbrace{\lambda \int_{x_i}^{x_n} \hat{S}''(x)^2 dx}_{2}$$

where $\hat{S}''(x)$ is the 2nd derivative of $\hat{S}(x)$. Then in Q_λ

1. is a penalty for misfitting the obs data
2. penalty for wigginess

λ controls the weighting of the penalties.

So we ask what type of (twice diff) function will minimize Q_λ?

The asuer is is a cubic smoothing spline w/ knots $x_1, \ldots, x_n$.

Comments: 1. This is a linear smoother. (see ref). and can be computed eff → in software.

Ex/ pg 330

2. As $\lambda \to \infty$ $\hat{S}_\lambda$ approaches a least squares line. As $\lambda = 0$, it is an interpolating spline connecting data points.
3. How to choose λ? CVRSS can be eff used.

Sec 11.4 <u>Nonlinear Smoothers</u>

- often much slower to calculate than linear smoothers & not much is gained. However there are certain types of data (ex when var $(Y|X)$ varies w/ X) for which other methods do poorly.

Sec 11.4.2 Supersmoother.

Look at figure 11.11 pg 335 How would you choose a span for this data?

- Left: smooth curve w/ large variance in the data ⇒ large span
- Right: wiggly data w/ low variability ⇒ small span

The supersmoother was designed for this type of problem. Basic idea ⇒ variable span.

Supersmoothing approach.

- Begin by calculating m different smooths, $\hat{S}_1(x), \ldots, \hat{S}_m(x)$, each w/ a fixed (diff) span $h_1, \ldots, h_m$. Can use linear.

Ex/ $m=3$ $h_1 = .05n$, $h_2 = .2n$, $h_3 = .5n$

See Fig 11.12 pg 336

- Next define $p(h_j, x)$ to be a measure of performance of the jth smooth at point x. ($j = 1, \ldots, m$).

Idealy we would use $E[g(Y - \hat{s}_j^{(i)}(x_i)) \mid X = x_i]$ where $g(x)$ is a symm. function that penalizes large deviations. & $\hat{s}_j^{(i)}$ is cross validation. Unfortunately, this is unknown, so est.

$$\hat{p}(h_j, x_i) = \hat{s}^*(g(Y_i - \hat{s}_j^{(i)}(x_i)))$$

where $\hat{s}^*$ is a fixed span smoother.

Ex/ $\hat{s}^* = \hat{s}_2$ & $g(z) = |z|$. See Fig 11.3 pg 336

- Then at each x_i denote by $\hat{h}_i$ the best of these spans, i.e. lowest $\hat{p}(h_j, x_i)$

Ex/ See Fig 11.14 pg 338

Note that adj points can have very diff best spans.

- Pass this data $(x_i, \hat{h}_i)$ throug some $\hat{s}^*$ to est. optimal ~~fun~~ span as a function of x. to obtain $\hat{h}(x)$

Ex $\hat{s}^* = \hat{s}_2$ see Fig 11.14 again.

Now we need to create the final smooth.

- Final smooth: A linear interpolation between $\hat{S}_{h^-(x_i)}(x_i)$ & $\hat{S}_{h^+(x_i)}(x_i)$ where among the m fixed spans

$h^-(x_i)$ is the largest span $< \hat{h}(x_i)$

$h^+(x_i)$ is the smallest span $> \hat{h}(x_i)$

Thus
$$\hat{s}(x_i) = \frac{\hat{h}(x_i) - h^-(x_i)}{h^+(x_i) - h^-(x_i)} S_{h^+(x_i)}(x_i) + \frac{h^+(x_i) - \hat{h}(x_i)}{h^+(x_i) - h^-(x_i)} \hat{S}_{h^-(x_i)}(x_i)$$

See Fig 11.15 pg 338
- Compared to a spline.

This method is *fast* compared to most other nonlinear smoothers.