Statistics April 21, 2021

$$MSE = \text{bias}^2 + \text{Variance}$$ | $y = r(x) + \epsilon$

$$\text{bias} = \mathbb{E}(\hat{r}(x)) - r(x)$$

Non-Parametric Regression. In machine learning this is known as "learning a function".

$$Y_i = r(x_i) + \epsilon_i \quad \leftarrow \text{model}$$

We will construct $\hat{r} = \hat{r}(x)$ ← estimator of r, or smoother

Assume that $Var(\epsilon_i) = \sigma^2$, independent of x_i.

Linear Smoother (not linear regression exactly).

Def: $\hat{r}$ is a linear smoother if for each x, there exists a vector $\vec{l}(x) = \begin{pmatrix} l_1(x) \\ l_2(x) \\ \vdots \\ l_n(x) \end{pmatrix}$ such that

$$\hat{r}(x) = \sum_{i=1}^{n} l_i(x) Y_i$$

$$\Rightarrow \vec{r} = \begin{pmatrix} \hat{r}(x_1) \\ \vdots \\ \hat{r}(x_n) \end{pmatrix}$$ fitted values

$$\vec{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

$\Rightarrow \quad \hat{r} = L\,Y \quad , \quad$ where $\quad L_{ij} = \ell_j(x_i)$

- The matrix L is called the smoothing matrix, or the "hat" matrix. i^{th} row is the "effective kernel" for the estimator of $r(x_i)$.

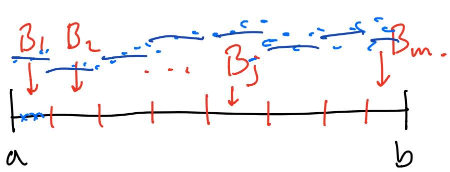
- The effective degrees of freedom $= \nu = tr(L)$.

<u>Exercise</u> Reinterp linear least squares in this matrix vector formulation. (I.e. find L, Y).

It will turn out that most linear smoothers have the property that for all x, $\sum \ell_i(x) = 1$. If this is true, then if $Y_i = c$ for all i, then $\hat{r}(x) = \sum \ell_i(x) Y_i = c$. (The smoother preserves constants.)

<u>Example</u> <u>Regressogram</u>

Let $x_i \in (a,b)$, and compute $\underline{m}$ bins on this interval:



$\Big\}$ Decide on m, call $h = \dfrac{b-a}{m}$

For $x \in B_j$, define $\hat{r}(x) = \underbrace{\dfrac{1}{k_j} \sum_{i : x_i \in B_j} Y_i}_{\text{computing the average of } Y_i \text{ over bin } B_j}$, $\quad k_j = \#\, x_i \in B_j$

computing the average of Y_i over bin B_j.

$\Rightarrow$ For $x \in B_j$, set $l_i(x) = \begin{cases} \frac{1}{k_j} & \text{if } x_i \in B_j \\ 0 & \text{otherwise} \end{cases}$

$\Rightarrow$ $\hat{r}(x) = \sum l_i(x) Y_i$ for $x \in B_j$

and $l(x) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1/k_j \\ 1/k_j \\ \vdots \\ 1/k_j \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ } i such that $x_i \in B_j$.

If $n=9$, $m=3$, and $k_1 = k_2 = k_3 = 3$.

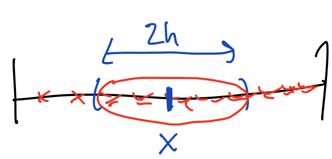
$$\Rightarrow \quad L = \begin{pmatrix} \begin{matrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{matrix} & 0 & 0 \\ 0 & \begin{matrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{matrix} & 0 \\ 0 & 0 & \begin{matrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{matrix} \end{pmatrix}$$

$\Rightarrow$ $\nu = tr(L) = 3$

Ex: Local Smoothing / Averaging

For $h > 0$, set $B_x = \{ i : |x_i - x| \le h \}$

$n_x = |B_x|$



$\Rightarrow$ $\hat{r}(x) = \frac{1}{n_x} \sum_{i \in B_x} Y_i$ average our points within a distance of h from x.

The key question in both of these examples is how to choose h?

- if h is big, we get a very smooth $\hat{r}$
- if h is small, then $\hat{r}$ looks a lot like Y_i

h is generally referred to as the <u>BANDWIDTH</u>

<u>Choosing the smoothing parameter</u>

To recall the risk: $R(h) = \underbrace{\mathbb{E}\left(\frac{1}{n} \sum_i \left(\hat{r}(x_i) - r(x_i) \right)^2 \right)}_{\text{this depends on the unknown } r.}$

Instead, choose h to minimize an estimate of R, $\hat{R}(h)$.

<u>Idea</u>: Use the "training error": $\underbrace{\frac{1}{n} \sum \left(Y_i - \hat{r}(x_i) \right)^2}_{\text{residual sum of squares RSS.}}$

- Poor estimate of $R(h)$
- Tends to lead to overfitting since outliers are given equal weight as other data points.

<u>A better option</u>

<u>Def</u>: "Leave-one-out cross validation"

$$\underset{\text{cross validation}}{CV} = \hat{R}(h) = \frac{1}{n} \sum \left(Y_i - \hat{r}_{(-i)}(x_i) \right)^2$$

$\hat{r}$ obtained by ignoring the i^{th} data point.

For linear smoothers, $\hat{r}(x) = \sum Y_i \, l_i(x)$

$$\hat{r}_{(-i)}(x) = \text{leave one out estimator}$$

$$= \sum_j Y_j \, l_{j,(-i)}(x)$$

where $l_{j(-i)}(x) = \begin{cases} 0 & \text{if } j = i \\ \dfrac{l_j(x)}{\sum_{k \neq i} l_k(x)} & \text{if } j \neq i \end{cases}$ } re-weighting of l_j

<u>General idea</u> compute $\hat{r}$ with only part of the data and then check.

If we use this as $\hat{R}(h)$, then what can we say?

$$\mathbb{E}\left(Y_i - \hat{r}_{(-i)}(x_i)\right)^2 = \mathbb{E}\left(\left(Y_i - r(x_i)\right) + \left(r(x_i) - \hat{r}_{(-i)}(x_i)\right)\right)^2$$

$$= \mathbb{E}\Big(\left(Y_i - r(x_i)\right)^2 + 2\left(Y_i - r(x_i)\right)\left(r(x_i) - \hat{r}_{(-i)}(x_i)\right)$$

model: $Y_i = r(x_i) + \epsilon_i$

$$+ \left(r(x_i) - \hat{r}_{(-i)}(x_i)\right)^2\Big)$$

$$= \sigma^2 + \mathbb{E}\left(\left(r(x_i) - \hat{r}_{(-i)}(x_i)\right)^2\right)$$

for large n $\approx \sigma^2 + \mathbb{E}\left(\left(r(x_i) - \hat{r}(x_i)\right)^2\right)$

$$\Rightarrow \mathbb{E}(\hat{R}) \simeq \underbrace{\sigma^2 + R}_{= \text{predictive risk}}$$

"nearly unbiased" since presumably $R \gg \sigma^2$.

Is this expensive to do for all i?

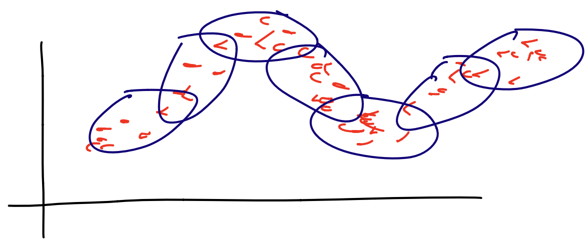
Thm For a linear smoother, the CV estimator can be written as

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{r}(x_i)}{1 - \underbrace{L_{ii}}_{=\ell_i(x_i)}}\right)^2$$

Then h can be chosen by minimizing $\hat{R}(h)$.

Local Regression

Idea: Give more weight to x_i, Y_i that are near to where you want to evaluate $\hat{r}$



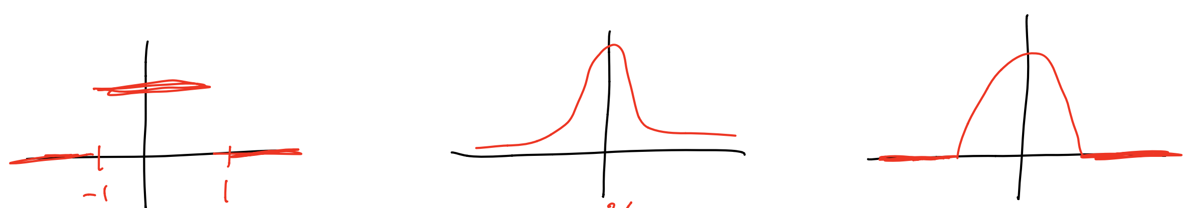
Definition A kernel is a function $K = K(x)$ such that

$$\int K(x)\,dx = 1$$

$$\int x K(x)\,dx = 0$$

$$\int x^2 K(x)\,dx = \sigma_K^2 > 0, \quad < \infty$$

Ex:



$\frac{1}{2}\mathbb{1}(x)$ $\quad \mathbb{1}(x) = \begin{cases} 1 & \text{if } |x| \le 1 \\ 0 & \text{else} \end{cases}$

$\frac{1}{\sqrt{2\pi}} e^{-x^2/2}$

$\frac{3}{4}(1-x^2)\,\mathbb{1}(x)$

Note that: For $h > 0$, and some x', we have that

$K_h(x - x') = \frac{1}{h} K\left(\frac{x - x'}{h}\right)$ is also a kernel centered at x'.

Idea: Use these "kernels" to smooth out the noisy data (x_i, Y_i).

Definition: For $h>0$, the Nadaraya-Watson kernel estimator is

$$\hat{r}(x) = \sum_{i=1}^{n} l_i(x)\, Y_i$$

where

$$l_i(x) = \frac{\frac{1}{h}K\left(\frac{x-x_i}{h}\right)}{\sum_{j=1}^{n} \frac{1}{h}K\left(\frac{x-x_j}{h}\right)}$$

This form is similar to barycentric interpolation formula, but it is not an interpolant.

Goal: ① Choose h via cross-validation.

② The choice of K is less important, especially when there is a lot of data.

In order to choose h, we need to be able to estimate the risk. For our purposes, assume that the x_j's are randomly drawn from some density f.

The risk can then be written as:

Thm

$$R(\hat{r}, r) = \underbrace{\frac{h^4}{4}\Big(\overbrace{\int x^2 K(x)\,dx}^{\sigma_K^2}\Big)^2 \left(\int \left(r'' + 2r'\frac{f'}{f}\right)^2 dx\right)}_{\text{bias}^2} + \underbrace{\frac{\sigma^2}{nh}\left(\int K^2(x)\,dx\right)\left(\int \frac{1}{f}\,dx\right)}_{\text{variance}} + o\left(\frac{n}{h}\right) + o(h^4)$$

where we have assumed that $h \to 0$ and $nh \to \infty$.

$\text{bias}^2 \sim h^4$

$\text{variance} \sim \frac{1}{nh}$

Comments

(1) "Design bias": $2r' \frac{f'}{f}$

Since this depends on f, the bias is sensitive to the distribution of the x_j's.

(2) To minimize R, set $\frac{dR}{dh} = 0$ and solve.

$\Rightarrow h_* \sim O\left(\frac{1}{n^{1/5}}\right)$

$\Rightarrow R \sim O\left(\frac{1}{n^{4/5}}\right)$. (vs. MLE for least squares, $R \sim \frac{1}{n}$).

Note

Homoscedasticity vs. Heteroskedacity

$y = r(x) + \epsilon$

$Var(\epsilon) = \sigma^2$,

and ϵ is independent of x ($\epsilon | x = \epsilon$).

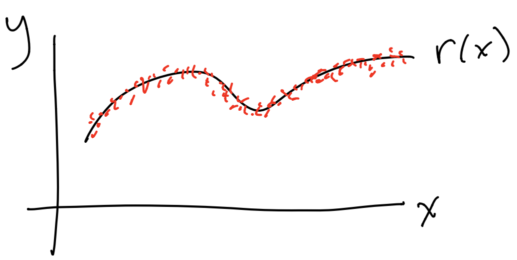
$y = r(x) + \sigma(x)\,\epsilon$

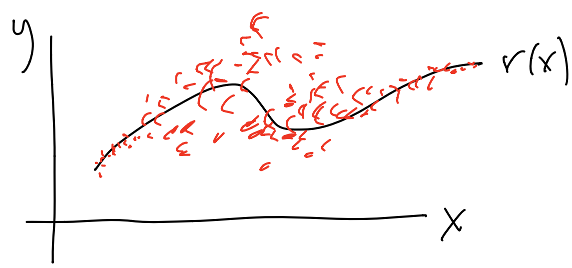
$Var(\epsilon) = 1$

function of x

$\Rightarrow Var(\sigma(x)\,\epsilon) = \sigma^2(x)\,Var(\epsilon)$

$= \sigma^2(x)$.





Reformulate the model to separate σ from ϵ.

$y - r = \sigma\epsilon$

$(y-r)^2 = \sigma^2 \epsilon^2$

$\log (y-r)^2 = \log \sigma^2 + \log \epsilon^2$

Set $Z_i = \log (Y_i - r(x_i))^2$

$\delta_i = \log \epsilon^2$

$\Rightarrow Z_i = \log \sigma^2(x_i) + \delta_i$

independent of x_i

Then our goal is to estimate $\log \sigma^2$.

There is a two-step procedure for doing this:

1. Estimate r in $Y_i = r(x_i) + \tilde{\varepsilon}_i$ to get $\hat{r}$.

2. Compute $Z_i = \log(Y_i - \hat{r}(x_i))^2$

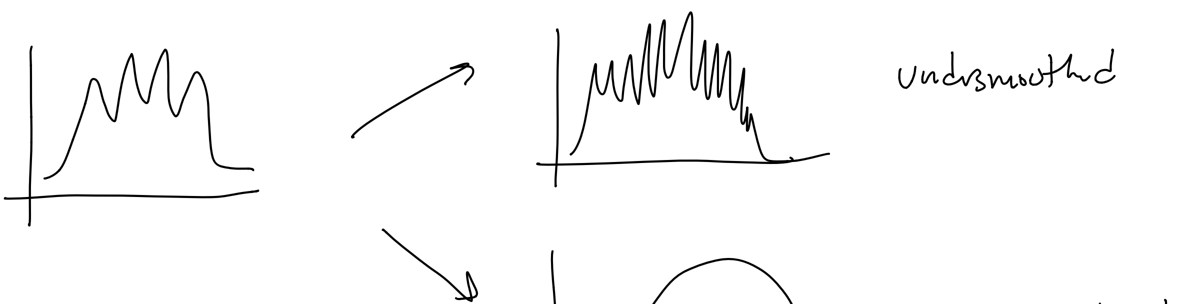
3. Regress Z_i on X_i again to get an estimator $\hat{q} \approx \log(\sigma^2(x))$. Set $\hat{\sigma}^2(x) = e^{\hat{q}(x)} > 0$.

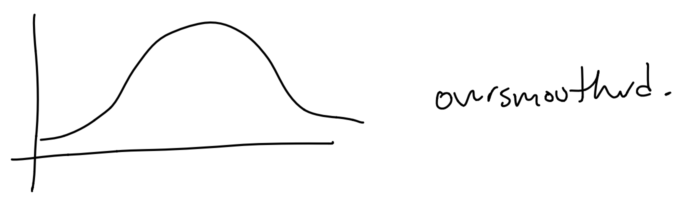
Density Estimation

Setup: Observe some data $X_1, \dots, X_n \sim F$, and therefore the density is $f = F'$.

Goal: Estimate f using as few assumptions as possible.

Still a smoothing problem:





Let $\hat{f}$ be our estimate of f.

One way to measure the error:

$L = \int (\hat{f}(x) -$