Regression basics

Werner Stuetzle

March 31, 2020

R code for slides courtesy of Yen-Chi Chen

A simple prediction problem – a single predictor variable

- Given training sample $(x_1, y_1) \ldots (x_n, y_n)$ $\quad x_i, y_i \in \mathbb{R}$
- Want to create prediction rule $f(x)$ that predicts y for given query point x.
- To cast this problem into a framework amenable to analysis we regard $(x_1, y_1) \ldots (x_n, y_n)$ as iid observations of a pair (X, Y) of random variables with joint density $p(x, y)$.

  X = predictor

  Y = response
- We measure the performance of a prediction rule $f(x)$ by its Expected Squared Prediction Error

$$ESE = E[(Y - f(X))^2]$$

$$= \int (y - f(x))^2 \, p(x, y) \, dx \, dy$$

- Define

  $p(x) = \int p(x, y) \, dy$ marginal density of X

  $p(y|x) = p(x, y) / p(x)$
- $ESE = \int_x \left[\int_y (y - f(x))^2 \, p(y|x) \, dy \right] p(x) \, dx$

$$= \int_x ESE(x) \, p(x) \, dx$$

$ESE(x)$ = expected squared prediction error for query x.

- $ESE(x)$ is minimized by choosing $f(x) = \int y \, p(y|x) \, dy = E(Y|x)$
- The optimal predictor of Y for given query x is the conditional mean $E(Y|x)$.
- The expected squared prediction error of the optimal predictor f for query x is the conditional variance $V(Y|x)$.

$V(Y|x)$ is the "irreducible error". No prediction rule can do better.

Kernel smoothing

Theoretical analysis gave insights but is not directly useful: We don't know $p(x,y)$ — we only have a sample

Natural apprach for estimating conditional mean $f(x) = E(Y|x)$: Local averaging or Kernel smoothing.

Simplest version

$$\hat{f}(x) = \text{mean}(y_i \mid |x - x_i| \leq h)$$

h = "bandwidth"

Better option: Give more weight to training obs with x_i close to query x.

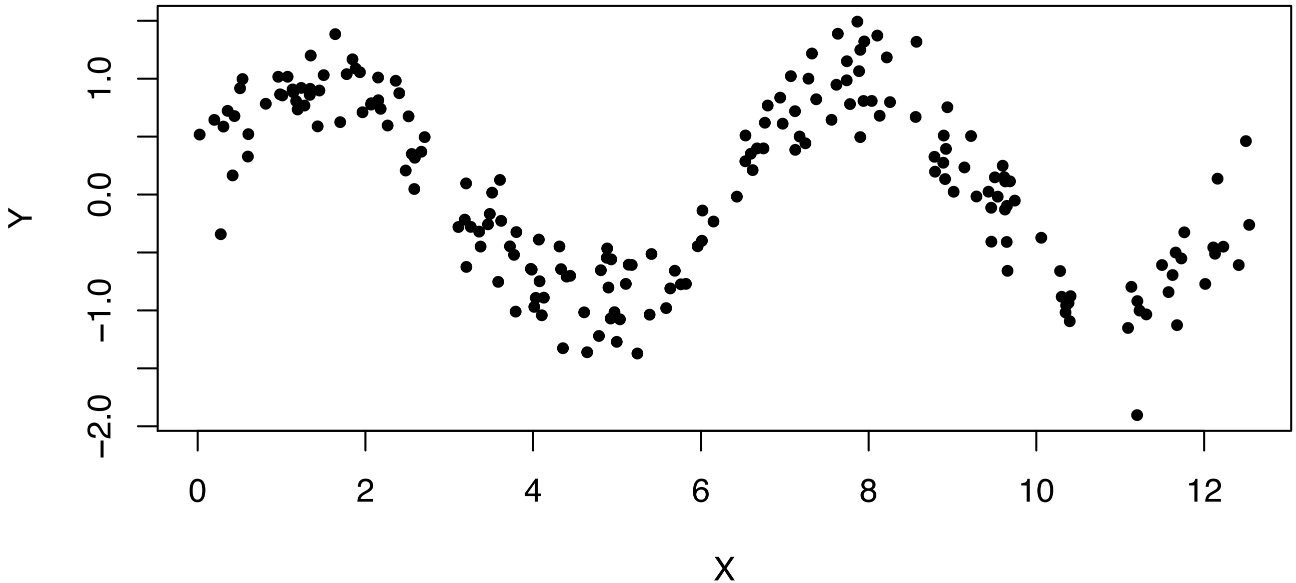
For example, define $\varphi_\sigma(x)$ = Gaussian density with mean 0 and standard dev σ

Define

$$\hat{f}(x) = \sum_i y_i \, \varphi_\sigma(x - x_i) \Big/ \sum_i \varphi_\sigma(x - x_i)$$

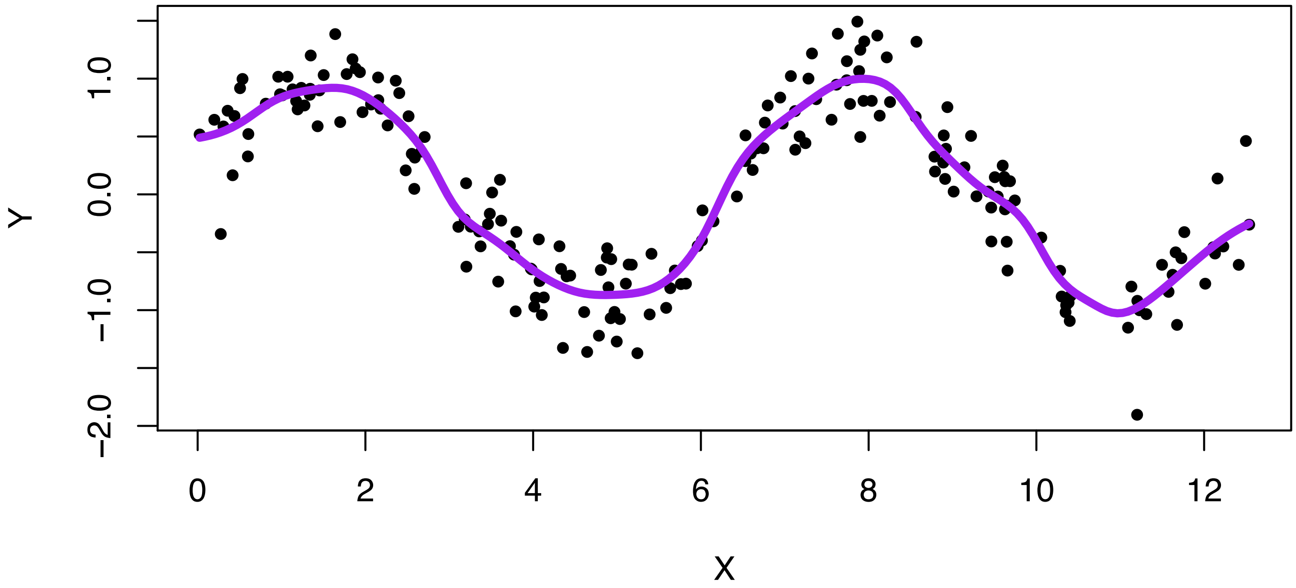
A simulated data set

```r
X = sort(runif(200, min=0, max=4*pi))
Y = sin(X) + rnorm(200, sd=0.3)
plot(X,Y, pch=20)
```



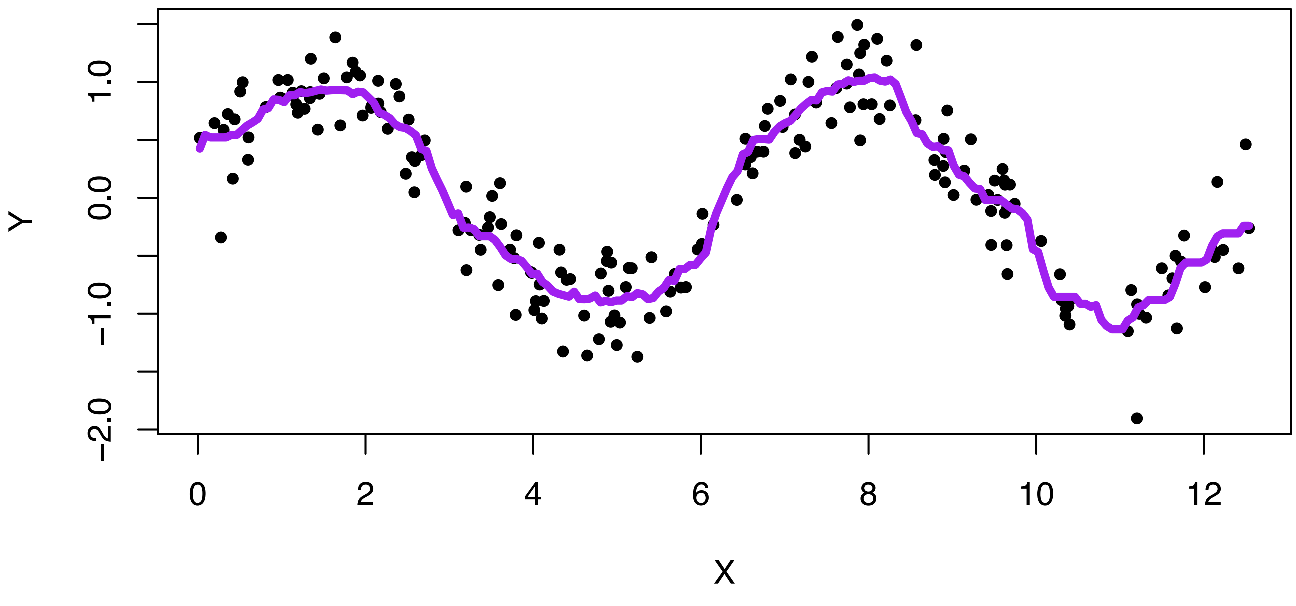
Local averaging, Gaussian kernel

```
Kreg = ksmooth(x=X,y=Y,kernel = "normal",bandwidth = 0.9)
plot(X,Y,pch=20)
lines(Kreg, lwd=4, col="purple")
```



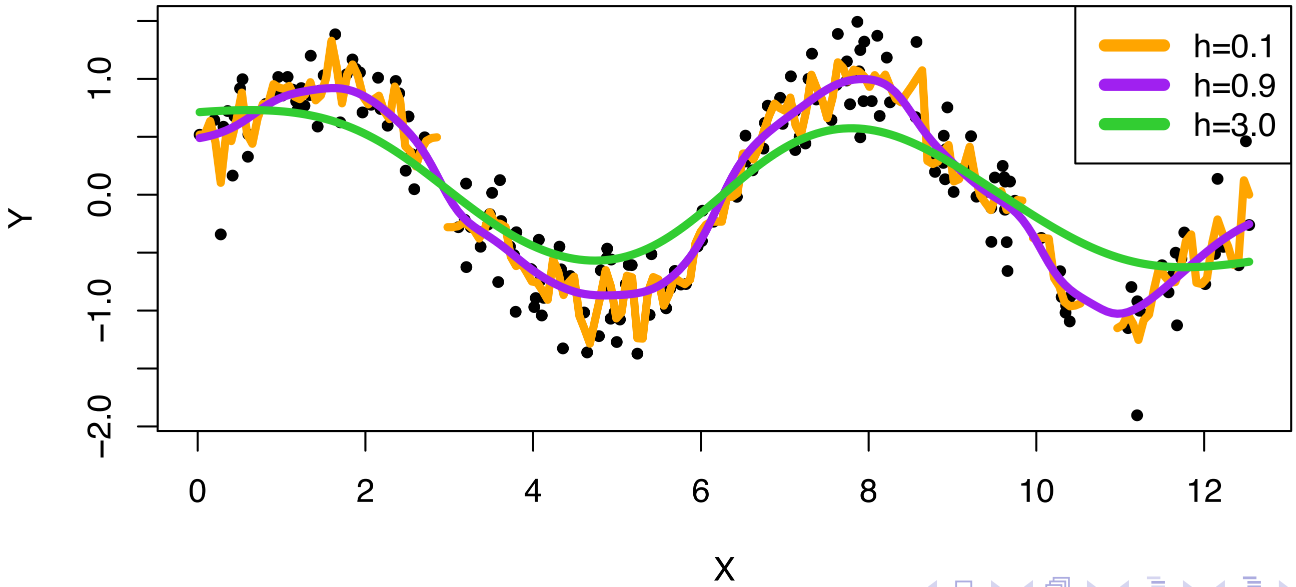
Local averaging, box kernel

```
Kreg = ksmooth(x=X,y=Y,kernel = "box",bandwidth = 0.9)
plot(X,Y,pch=20)
lines(Kreg, lwd=4, col="purple")
```

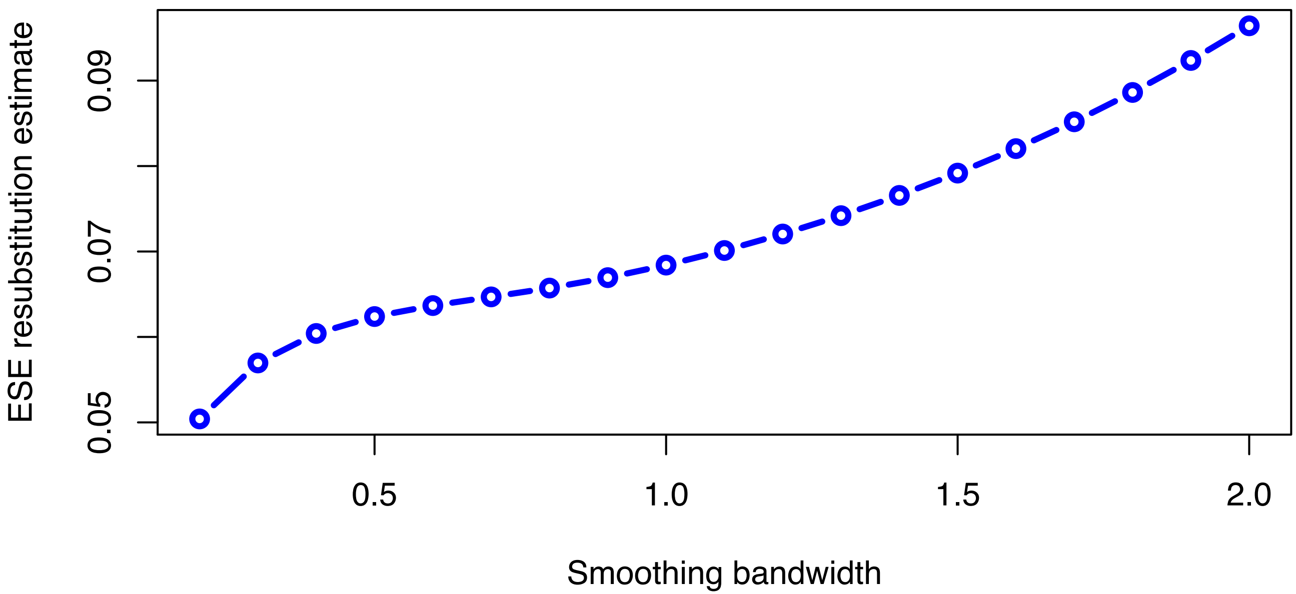


Different bandwidths

```
Kreg1 = ksmooth(x=X,y=Y,kernel = "normal",bandwidth = 0.1)
Kreg2 = ksmooth(x=X,y=Y,kernel = "normal",bandwidth = 0.9)
Kreg3 = ksmooth(x=X,y=Y,kernel = "normal",bandwidth = 3.0)
plot(X,Y,pch=20)
lines(Kreg1, lwd=4, col="orange")
lines(Kreg2, lwd=4, col="purple")
lines(Kreg3, lwd=4, col="limegreen")
legend("topright", c("h=0.1","h=0.9","h=3.0"), lwd=6,
       col=c("orange","purple","limegreen"))
```

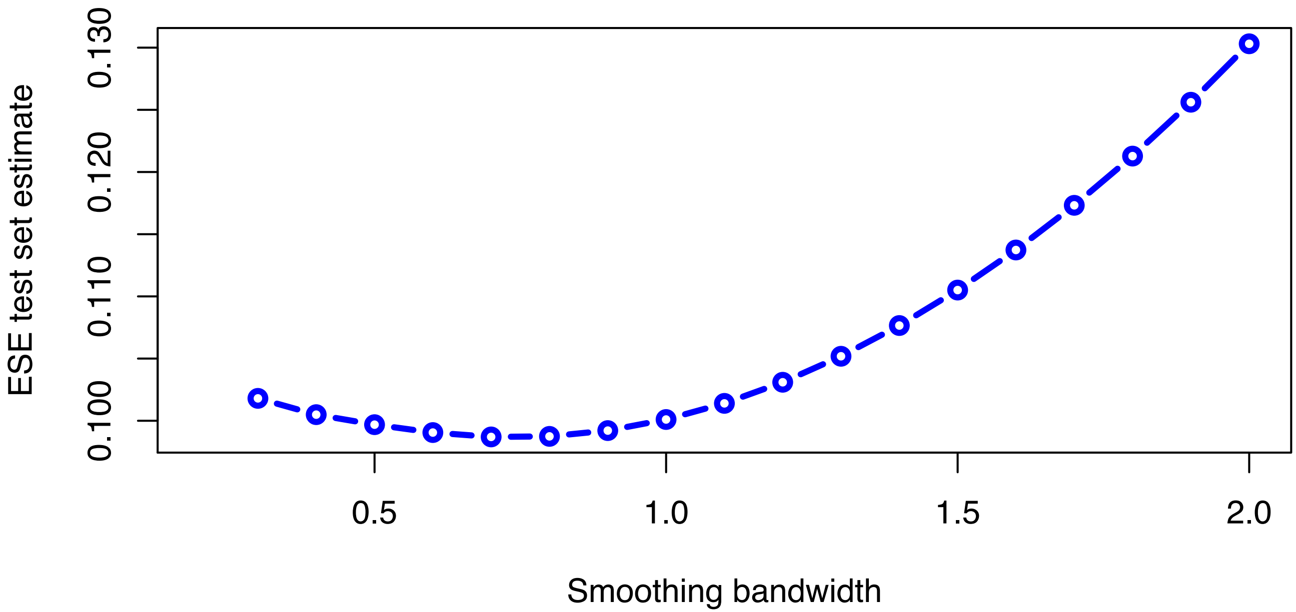


ESE resubstitution estimate



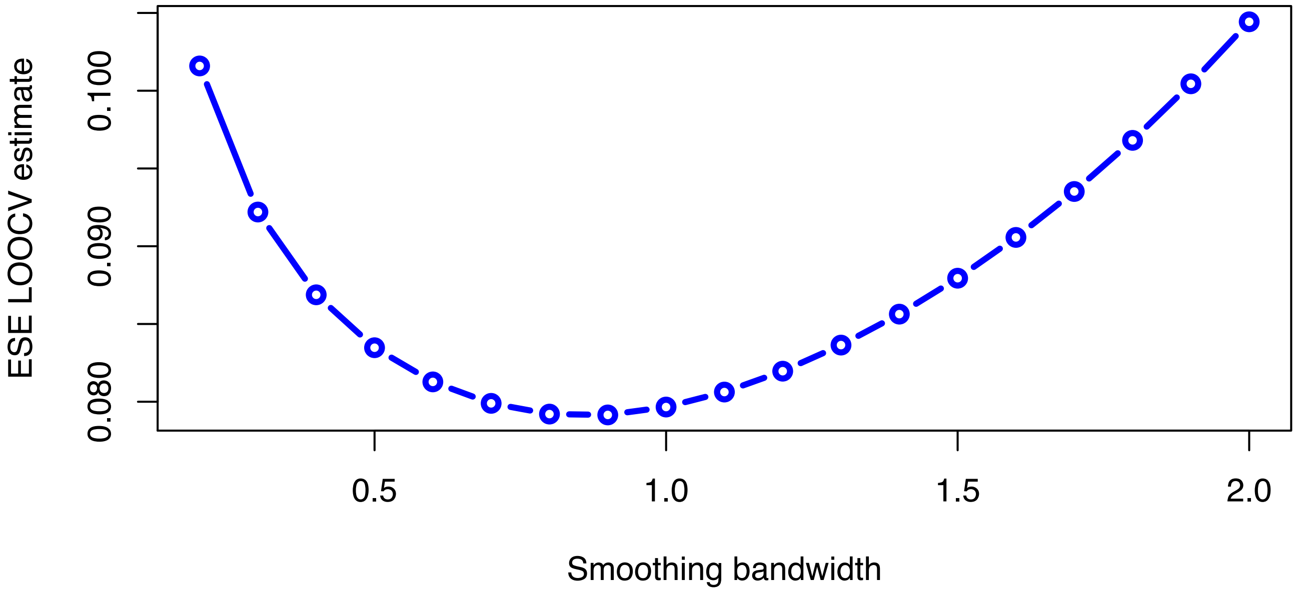

```
## hopt =  0.2
```

ESE test set estimate



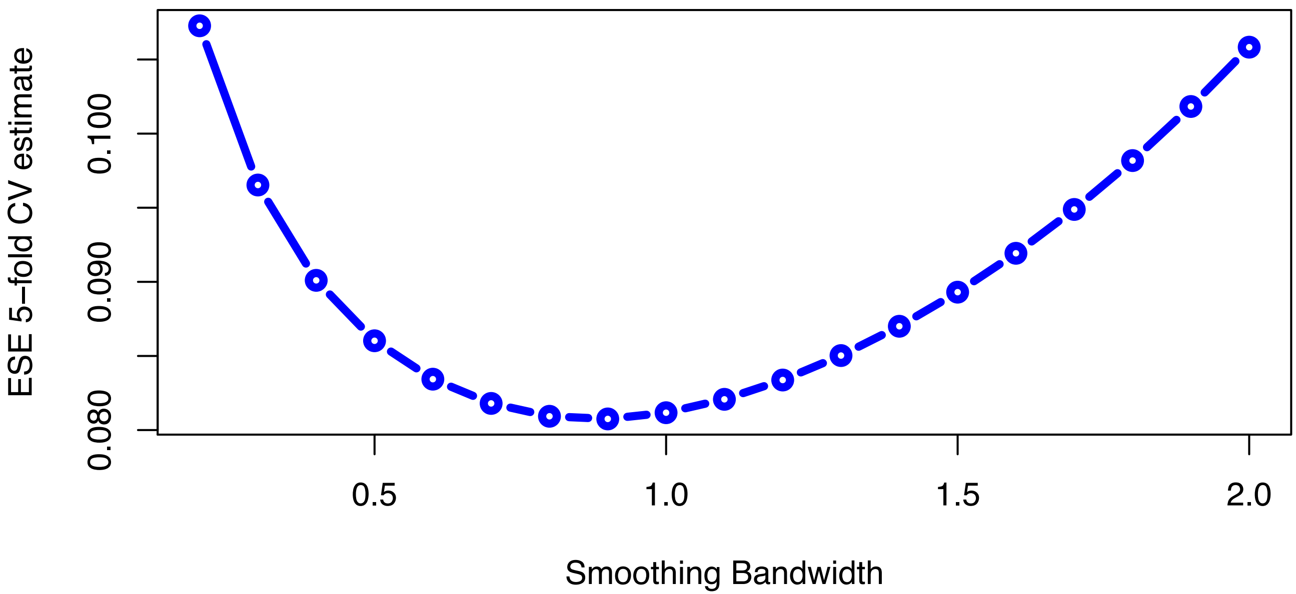

```
## hopt =  0.7
```

ESE leave-one-out cross-validation estimate




```
## hopt =  0.9
```

ESE 5-fold cross-validation estimate




```
## hopt =  0.9
```