ST720 Data Science

Regression

Seung Jun Shin (sjshin@krea.ac.kr)

Department of Statistics, Korea University

Introduction

- Suppose we are given a set of data, $(y_i, \mathbf{x}_i), i = 1, \cdots, n$

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \qquad i = 1, \cdots, n$$

with $\epsilon_i \sim F$ being a random error.

Linear Regression

- The well-known linear regression assumes

$$y_i = \beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i + \epsilon_i, \qquad \epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

- LSE sovles

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} (y - \beta_0 - \boldsymbol{\beta}^T \mathbf{x}_i)^2$$

  which can be viewed as an ERM formulation with

$$r_i = y_i - \beta_0 - \boldsymbol{\beta}^T \mathbf{x}_i \qquad \text{, and} \qquad L(r) = r^2$$

- This yields a LS estimator.

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

Quantile Regression

- Under ERM formulation, we can use an alternative loss function.
- Notice that

$$E(Y \mid \mathbf{X} = \mathbf{x}) = \operatorname*{argmin}_f E(\{Y - f(\mathbf{X})\}^2 \mid \mathbf{X} = \mathbf{x})$$

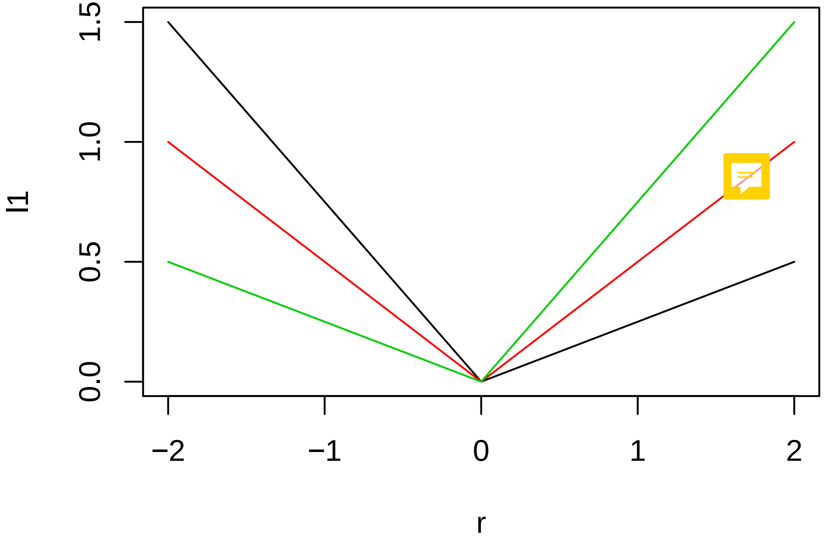
- Similary, we have

$$F^{-1}_{Y|\mathbf{X}=\mathbf{x}}(\tau) = \operatorname*{argmin}_f E[\rho_\tau\{Y - f(\mathbf{X})\} \mid \mathbf{X} = \mathbf{x}]$$

where

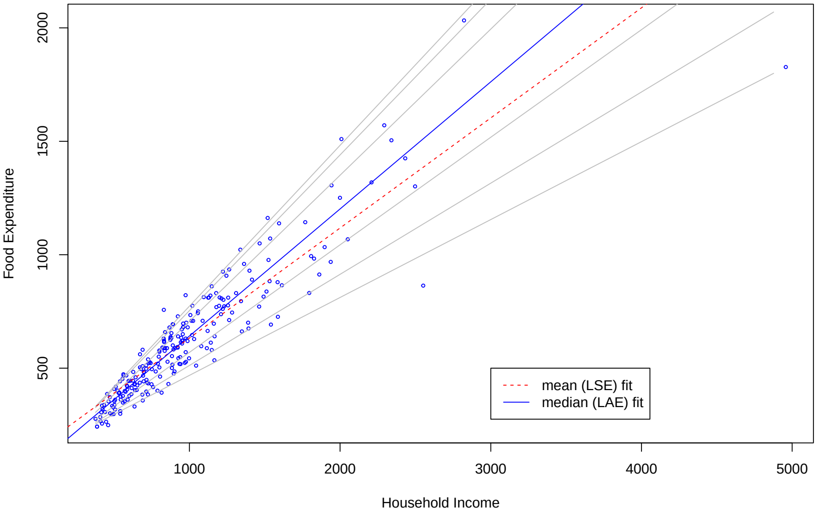
$$\rho_\tau(r) = r(\tau - \mathbb{1}\{r < 0\})$$

Check Loss Function



Quantile Regression

```r
library(quantreg)
rq(foodexp ~ income, tau = c(.05,.1,.25,.75,.9,.95))
```



Locally Weighted Smoothing Scatter Plot (LOWESS)

- Nonlinear learning for one dimensional regression function.
- LOWESS algorithm: WLOG, predicotrs are sorted $x_1 < x_2 < \cdots < x_n$.
  - Consider windows of width $K = (2k+1)$ centered at x_i:

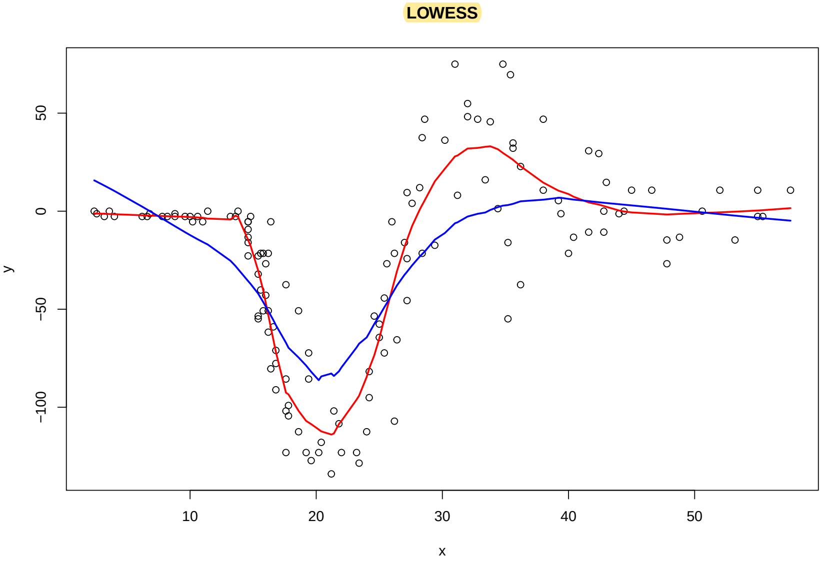
$$(x_{i-k}, y_{i-k}), \cdots, (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}), \cdots, (x_{i+k}, y_{i+k})$$

  - Apply WLS within the window and compute the fitted values of x_i with

$$w_j = \left(1 - (d_j/d_{max})^3\right)^3, \qquad j = 0, \pm 1, \cdots, \pm k$$

    with $d_j = |x_{i+j} - x_i|$ and $d_{max} = \max(d_0, d_{\pm 1}, \cdots, d_{\pm k})$.
  - Connect the n observations: $(x_1, \hat{y}_1), \cdots, (x_i, \hat{y}_i), \cdots, (x_n, \hat{y}_n)$ where $\hat{y}_i$ denotes the fitted value of y_i from the WLS regression.

Locally Weighted Smoothing Scatter Plot (LOWESS)



Local Polynomial Regression

- Taylor expansion of $f(x)$

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \cdots + \frac{f^{(p)}(x_0)}{p!}(x - x_0)^p$$

- To estimate $\hat{f}(x_0)$, LPR solves

$$\min_{\beta} \sum_{i=1}^{n} \{y_i - \beta_0 + \beta_1 x + \cdots + \beta_p x^p\}^2 K_h(x_i - x_0)$$

where $K_h(x_i - x)$ denotes a kernel function with bandwidth h such as

$$K_h(x_i - x) = \frac{1}{h}\phi\left(\frac{x_i - x_0}{h}\right)$$

Local Polynomial Regression

- When $p = 0$: Nadaraya-Watson (kernel regression) estimator.

$$\hat{f}(x_0) = \underset{\beta_0}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \beta_0)^2 K_h(x_i - x_0)$$

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{n} y_i K_h(x_i - x_0)}{\sum_{i=1}^{n} K_h(x_i - x_0)}$$

- When $p = 1$: $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$ where

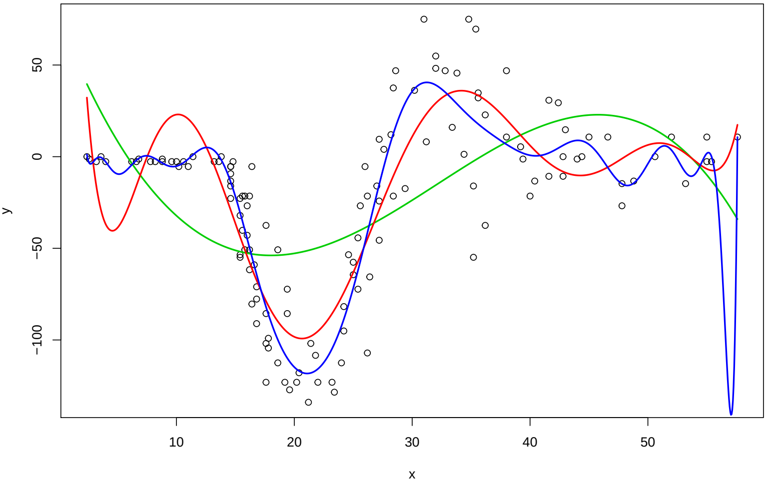
$$(\hat{\beta}_0, \hat{\beta}_1) = \underset{\beta_0}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_i x_i)^2 K_h(x_i - x_0)$$

which is equivalent to LOWESS with a suitable choice of the kernel.

Polynomial Regression

- Polynomial Regression

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_p x_i^p + \epsilon_i$$



Regression Spline

- Let's generalize a little bit:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_K b_K(x_i) + \epsilon_i$$

for a given set of function $\{b_1, \cdots, b_K\}$, which we call Basis function.

Cubic Regression Spline

- A Cubic spline with K knots $\{t_1 < \cdots t_K\}$ can be modeled as

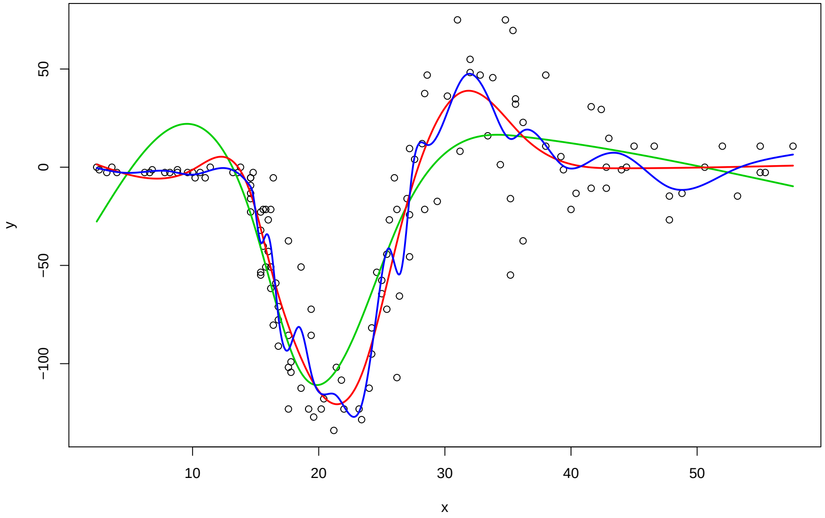
$$y_i = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \sum_{k=1}^{K} \beta_{k+3} h(x, t_k)$$

where $h(x, t)$ is the truncated power basis function:

$$h(x,t) = (x-t)_+^3 = \begin{cases} (x-t)^3 & \text{if } x > t \\ 0 & \text{otherwise} \end{cases}$$

- Notice that the regression function is continous and second order differentiable.
- Natural cubic spline requires additional requirements: regression function is linear on $(-\infty, t_1]$ and $[t_K, \infty]$.

Regression Spline



Smoothing Spline

- Regression spline may overfit the model.
- A natural remidy is to solve

$$\underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \{y_i - f(x_i)\}^2 + \lambda \int \{f^{(k+1)/2}(t)\}^2 dt$$

- Remarkably, the solution is a natural kth order spline with knots at the input points $x_1, \cdots, x_n$.

B-spline

- The most populat basis function.

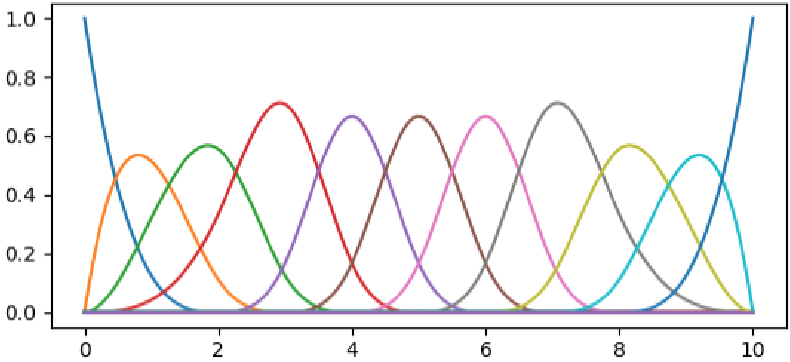
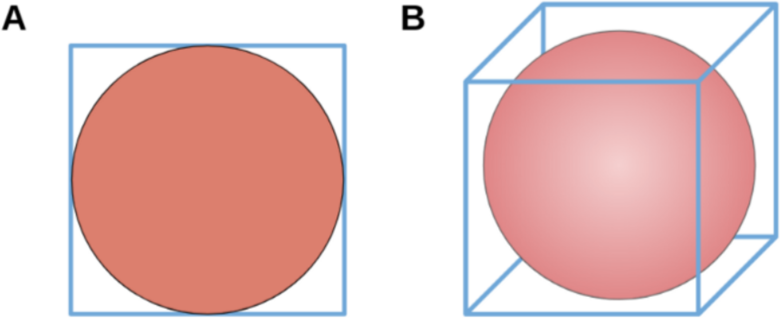


Figure 1: B-spline Basis Functions

Curse of Dimensionality

- What if x is not univariate?
- Both local regression and spline method suffers due to the Curse of Dimensionality.
- Namely, the dense of data in the entire space exponentially decreases as the dimension increases.



Generalized Additive Model (GAM)

- GAM model assumes

$$y = \beta_0 + g_1(x_1) + g_2(x_2) + \cdots + g_p(x_p) + \epsilon$$

  where $E(g_j(X_j)) = 0$ for all $j = 1, \cdots, p$.

- The goal is to estimate g_js as well as β_0.

Backfitting Algorithm

- Initialize $\hat{g}_j(x) = 0$ for all $j = 1, \cdots, p$ and $\hat{\mu} = \bar{y}$.
- For each $k = 1, \cdots, p$:
  - Compute partial residual:

$$\tilde{y}_i = y_i - \hat{\mu} - \sum_{j \neq k} \hat{g}_k(x_{ij})$$

  - Apply a nonparametric regression to $(x_{ij}, \tilde{y}_i)$ to update $\hat{f}_k(\cdot)$.
  - Centering $\hat{f}_k(\cdot)$ by computing

$$\hat{f}_k(x) \leftarrow \hat{f}_k(x) - \frac{1}{n} \sum_{i=1}^{n} \hat{f}_j(x_{ij})$$

- Repeat the above until convergence.

GAM: SAT Data Example

```r
library(mosaic)
library(mgcv)
data(SAT)
head(SAT)
```

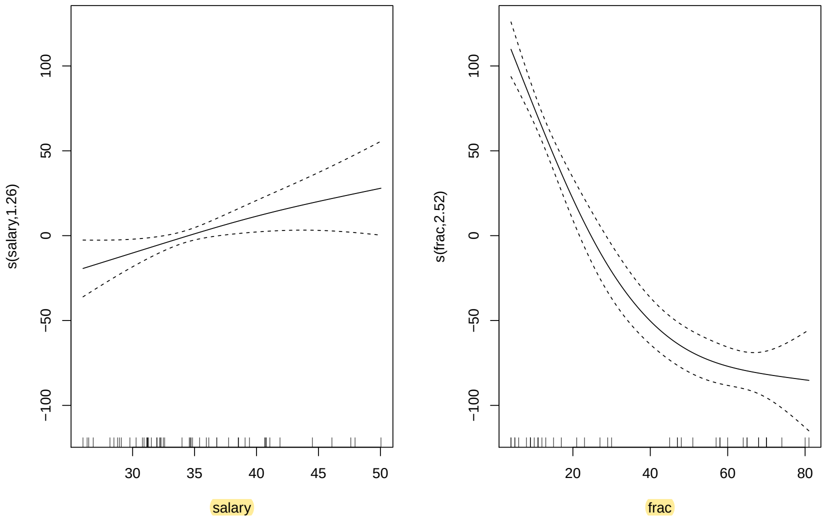
```
##         state expend ratio salary frac verbal math  sat
## 1     Alabama  4.405  17.2 31.144    8    491  538 1029
## 2      Alaska  8.963  17.6 47.951   47    445  489  934
## 3     Arizona  4.778  19.3 32.175   27    448  496  944
## 4    Arkansas  4.459  17.1 28.934    6    482  523 1005
## 5  California  4.992  24.0 41.078   45    417  485  902
## 6    Colorado  5.443  18.4 34.571   29    462  518  980
```

```r
obj <- gam(sat ~ s(salary, k = 4) + s(frac, k = 4), data = SAT)
summary(obj)
```

GAM: SAT Data Example

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## sat ~ s(salary, k = 4) + s(frac, k = 4)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  965.920      3.696   261.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## s(salary) 1.259  1.467  3.706  0.0344 *
## s(frac)   2.516  2.826 99.916  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.878   Deviance explained = 88.7%
## GCV = 755.33  Scale est. = 683.2     n = 50
```

GAM: SAT Data Example



Kernel Ridge Regression

- Kernel trick can be applied to regression.
- Consider

$$\min_{f \in \mathcal{H}_K} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 + \lambda \|f\|^2_{\mathcal{H}_K}$$

where

$$\mathcal{H}_K = \left\{ f : f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \theta_i K(\mathbf{x}, \mathbf{x}_i) \right\}$$

- Now we have

$$\min_{\beta_0, \boldsymbol{\theta}} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \theta_i K(\mathbf{x}_i, \mathbf{x}_j))^2 + \lambda \boldsymbol{\theta}^T \mathbf{K} \boldsymbol{\theta}$$

Kernel Ridge Regression

- Let $\boldsymbol{\theta}^* = (\beta_0, \boldsymbol{\theta}^T)^T \in \mathbb{R}^{n+1}$, we have

$$\min_{\boldsymbol{\theta}^*} \ (\mathbf{y} - \mathbf{K}^*\boldsymbol{\theta}^*)^T(\mathbf{y} - \mathbf{K}^*\boldsymbol{\theta}^*) + \lambda \boldsymbol{\theta}^{*T}\mathbf{K}\boldsymbol{\theta}^*$$

  where

$$\mathbf{K}^* = (\mathbf{1}, \mathbf{K}) \in \mathbb{R}^{n\times(n+1)}, \quad \text{and} \quad \tilde{\mathbf{K}} = \text{diag}(1, \mathbf{K}) \in \mathbb{R}^{(n+1)\times(n+1)}$$

- The kerenel ridge regression estimator is

$$\hat{\boldsymbol{\theta}}^* = \left(\mathbf{K}^{*T}\mathbf{K}^* + \tilde{\mathbf{K}}\right)^T \mathbf{K}^{*T}\mathbf{y}$$

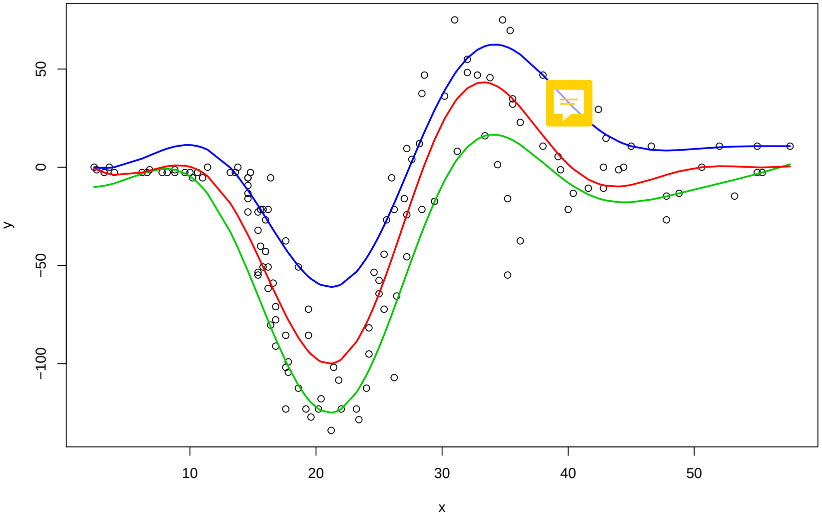
Kernel Quantile Regression

- Instead of LS loss, the check loss yields the KQR

$$\min_{\beta_0, \boldsymbol{\theta}} \sum_{i=1}^{n} \rho_\tau \left\{ y_i - \beta_0 - \sum_{j=1}^{p} \theta_i K(\mathbf{x}_i, \mathbf{x}_j) \right\} + \lambda \boldsymbol{\theta}^T \mathbf{K} \boldsymbol{\theta}$$

  where $\rho_\tau(r) = r(\tau - \mathbb{1}\{r < 0\})$.
- The KQR estimates the τth conditional quantile of $y \mid \mathbf{X} = \mathbf{x}$.

KQR: Example



Controling Flexibility

- Tuning is important in flexible learning.
- Our goal is to optimaize the prediction performance. (i.e., minimizing test error rate)
  - AIC, BIC
  - Cross Validation