Dongyang Wang

Stat 527

(a) $E((\hat{\theta}-\theta)^2)$

$= E((\hat{\theta} - E(\hat{\theta}) + E(\hat{\theta}) - \theta)^2)$

$= E((\hat{\theta}-E(\hat{\theta}))^2) + 2E(\hat{\theta}-E(\hat{\theta}))\cdot(E(\hat{\theta})-\theta)$
$\quad + (E(\hat{\theta})-\theta)^2$

$= Var(\hat{\theta}) + 2E(\hat{\theta}-E(\hat{\theta}))\cdot(\theta-\theta) + Bias^2(\theta)$

$= Var(\hat{\theta}) + Bias^2(\theta)$

where $Bias^2(\theta) = (E(\hat{\theta})-\theta)^2$

(b) Markov Inequality:

$E(x) = \int_0^a x f(x)\,dx + \int_a^\infty x f(x)\,dx$

$\geq \int_a^\infty x f(x)\,dx$

$\geq a \int_a^\infty f(x)\,dx$

$= a \cdot P(x \geq a)$

$P(x \geq a) \leq \frac{E(x)}{a}$

Thus, $P(\frac{T}{a} > \frac{1}{\varepsilon}) \leq \frac{E(\frac{T}{a})}{\frac{1}{\varepsilon}}$

$= \varepsilon \cdot \frac{E(T)}{a}$

$\leq \varepsilon \cdot 1$

$= \varepsilon$

(c) Since $E(x_i) = 0$ and $Var(x_i) = \Sigma$,

$E(x_i^T\beta) = 0 \qquad E(\hat{\beta}) = \beta$

$Var(x_i^T\beta) = \beta^2\Sigma$

$Var(X_i^T\varepsilon) = \sigma^2 E[X^TX]$

By Slutsky's theorem,

$\sqrt{n}(\hat{\beta}-\beta) = (\frac{X_i^TX_i}{n})^{-1}\frac{1}{\sqrt{n}}X^T\varepsilon$

and

$\sqrt{n}(\hat{\beta}-\beta) \sim N(0, \sigma^2\Sigma^{-1}\Sigma\Sigma^{-1})$

$\Rightarrow N(0, \sigma^2\Sigma^{-1})$

So the rate of convergence of $\hat{\beta}$ to β is $\frac{1}{n}$ for the variance and MSE since

$n(\hat{\beta}-\beta)^2 \sim \chi^2$.

$\Longrightarrow$

(c) $\sqrt{n}(\hat{\beta}-\beta) = \sqrt{n}((X^TX)^{-1}X^TY-\beta)$

$$= \sqrt{n}\left[(X^TX)^{-1}X^TX\beta-\beta+(X^TX)^{-1}X^T\varepsilon\right]$$

$$= \sqrt{n}\left[\beta-\beta+(X^TX)^{-1}X^T\varepsilon\right]$$

$$= \left(\frac{X^TX}{n}\right)^{-1}\cdot\sqrt{n}\,\frac{X^T\varepsilon}{n}$$

By continuous mapping theorem.

$$\left(\frac{X^TX}{n}\right)^{-1} = \left(\frac{1}{n}\sum_{i=1}^{n}X_i^2\right)^{-1}$$

By Law of Large Numbers $\Rightarrow \left(E(X_i^2)\right)^{-1}$

$$= \left(Var(X_i)+E(X_i)^2\right)^{-1}$$

$$= \Sigma^{-1}$$

So, $\left(\frac{X^TX}{n}\right)^{-1} \xrightarrow{P} \Sigma^{-1}$

$$\frac{X^T\varepsilon}{n} = \frac{1}{n}\sum_{i=1}^{n}X_i\varepsilon_i$$

Since $X_i \perp\!\!\!\perp \varepsilon_i$ and they are both iid.,

$$E\left(\frac{X^T\varepsilon}{n}\right) = \frac{1}{n}\sum_{i=1}^{n}E(X_i\varepsilon_i) = \frac{1}{n}\sum_{i=1}^{n}E(X_i)E(\varepsilon_i)$$

$$= \frac{1}{n}\cdot\sum_{i=1}^{n}0\cdot E(\varepsilon_i) = 0$$

$$Var\left(\frac{X^T\varepsilon}{n}\right) = \frac{Var(X_i\varepsilon_i)}{n}$$

$$= \frac{Var(E(X_i\varepsilon_i\mid\varepsilon_i))+E(Var(X_i\varepsilon_i\mid\varepsilon_i))}{n}$$

$$= \frac{\varepsilon_i^2\,Var(E(X_i))+E(\varepsilon_i^2\,Var(X))}{n}$$

$$\cancel{= \frac{0+\sigma^2\cdot\Sigma}{n}}$$

$$= \frac{0+(E(\varepsilon_i)^2+Var(\varepsilon_i))\cdot\Sigma}{n}$$

$$= \frac{\sigma^2\Sigma}{n}$$

Thus by Central Limit Theorem)

$\cancel{\sqrt{n}(\hat{\beta}-\beta)}$ $\sqrt{n}\left(\frac{X^T\varepsilon}{n}-0\right) \xrightarrow{D} N(0,\sigma^2\Sigma)$

Thus. by Slutsky's Theorem,

$$\sqrt{n}(\hat{\beta}-\beta) \xrightarrow{D} \Sigma^{-1}\cdot N(0,\Sigma\sigma^2)$$

Since

$$Var(\Sigma^{-1}N(0,\Sigma\sigma^2))$$

$$= (\Sigma^{-1})^2\cdot\Sigma\sigma^2$$

$$= \Sigma^{-1}\sigma^2$$

Therefore,

$$\sqrt{n}(\hat{\beta}-\beta) \xrightarrow{D} N(0,\sigma^2\Sigma^{-1})$$

So, the rate of convergence of $\hat{\beta}$ to β is $\frac{1}{n}$ for the MSE and variance

since $n(\hat{\beta}-\beta)^2 \sim \chi^2$

HW1

Q2

```r
rm(list =ls())
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(ggplot2)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
sample_size = c(100,500,1000,5000,10000)
total_res = list()

test_vector = seq(0.001, 10.001,0.1)
bandwidth1 = data.frame()
bandwidth2 = data.frame()

for (i in 1:3){
  res = data.frame()
  for (j in 1:length(sample_size)){
  set.seed(100)
  n = sample_size[j]
  x = runif(n)
  e = rnorm(n)

  data1 = data.frame(y = 2*x+e, fx = 2*x, x = x) %>% arrange(x)
  data2 = data.frame(y = sin(2*pi*x)+e, fx = sin(2*pi*x), x = x) %>% arrange(x)
  data3 = data.frame(y = sin(30*x)+e, fx = sin(30*x), x = x) %>% arrange(x)
  data = list(data1, data2, data3)

  # Linear Regression
  linear_model <- lm(y~x, data = data[[i]])
  res[j,1] = mean((data[[i]]$fx - predict(linear_model))^2)
  # Polynomial Regression
  for (d in 2:5){
    ols  <- lm(y ~ poly(x, degree = d, raw = T), data = data[[i]])
    res[j,d]  <- mean((data[[i]]$fx - predict(ols))^2)
  }

  # Box Kernel
  mseb = c()
  for (c in test_vector){
    Box = ksmooth(data[[i]]$x, data[[i]]$y, kernel = 'box',
                bandwidth = c*n^(-1/3), x.points = data[[i]]$x)
    mse_b = mean((data[[i]]$fx - Box$y)^2)
    mseb = c(mseb, mse_b)
  }
  bandwidth1[i,j] =test_vector[which(mseb == min(mseb))]
  res[j,6] = min(mseb)

  # Gaussian Kernel
  mseg = c()
  for (c in test_vector){
    Gaussian = ksmooth(data[[i]]$x, data[[i]]$y, kernel = 'normal',
                bandwidth = c*n^(-1/3), x.points = data[[i]]$x)
    mse_g = mean((data[[i]]$fx - Gaussian$y)^2)
    mseg = c(mseg, mse_g)
  }
  bandwidth2[i,j] = test_vector[which(mseg == min(mseg))]
  res[j,7] = min(mseg)

  }
  total_res[[i]] = res
}

total_res
```

```
## [[1]]
##             V1           V2           V3           V4           V5          V6
## 1 0.0079818501 0.0103426531 0.0250556965 0.0760476786 0.0783464563 0.021944810
## 2 0.0009830938 0.0035687585 0.0039786335 0.0039796820 0.0041632781 0.002895338
## 3 0.0009126252 0.0046596559 0.0049121767 0.0049128294 0.0068597300 0.007810911
## 4 0.0004288144 0.0005128162 0.0005897578 0.0006592695 0.0006600119 0.001086535
## 5 0.0001059443 0.0003836297 0.0006014305 0.0006152165 0.0006173871 0.001153934
##             V7
## 1 0.0223068482
## 2 0.0025168313
## 3 0.0075348748
## 4 0.0009644999
## 5 0.0011455829
##
## [[2]]
##          V1        V2          V3          V4           V5          V6
## 1 0.1792165 0.1747448 0.028500803 0.079346392 0.0783591796 0.058477237
## 2 0.2069668 0.2084479 0.008263360 0.008237423 0.0041815890 0.010764534
## 3 0.2077984 0.2115082 0.009542063 0.009523694 0.0068784878 0.011283186
## 4 0.1988099 0.1988375 0.005144396 0.005213381 0.0006787481 0.002856624
## 5 0.1976002 0.1978471 0.005098232 0.005111807 0.0006359356 0.002343034
##            V7
## 1 0.059512213
## 2 0.009571311
## 3 0.010469916
## 4 0.002674200
## 5 0.002244159
##
## [[3]]
##          V1        V2        V3        V4        V5          V6          V7
## 1 0.5572074 0.5575200 0.5671735 0.6177555 0.5742761 0.147528405 0.135258924
## 2 0.5184589 0.5190555 0.4957680 0.4956021 0.4767644 0.039874509 0.037769134
## 3 0.5099599 0.5114768 0.4922005 0.4912148 0.4673247 0.027076698 0.025919426
## 4 0.5083514 0.5069468 0.4872515 0.4872270 0.4605000 0.008767326 0.008435396
## 5 0.4998430 0.4988743 0.4827795 0.4827359 0.4555464 0.004530650 0.004200468
```

For the bandwidth, I have used the following code for calculation. Basically, I tried to obtain the lowest training MSE while simulating the data. Admittedly, this practice is not perfect because we want to use cross validation when possible; but this practice still works because this is a simulation study. The choices I made are based the theretical framework that the bandwidth is a multiple of the optimal bandwidth, $h = n^{-1/3}$. The bandwidth I used is reported in following tables: the first for the box kernel, the second for the Gaussian kernel.

```
sample <- matrix(rep(sample_size,length(sample_size)), byrow = T, ncol = length(sample_size), nrow = len
optimal_bw1 = bandwidth1 * sample^(-1/3)
optimal_bw1
```

```
##           V1         V2         V3        V4     V5
## 1 0.45264673 0.58191281 0.28990783 0.3907015 0.3001
## 2 0.25874761 0.25874761 0.13871731 0.1601000 0.1501
## 3 0.08639283 0.06312204 0.07572126 0.0501000 0.0701
```

```
optimal_bw2 = bandwidth2 * sample^(-1/3)
optimal_bw2
```

```
##           V1         V2         V3        V4     V5
## 1 0.36646934 0.43110238 0.25211020 0.3403047 0.2501
## 2 0.21565891 0.19411457 0.12611810 0.1201000 0.1301
## 3 0.06484848 0.05052283 0.05052283 0.0401000 0.0601
```

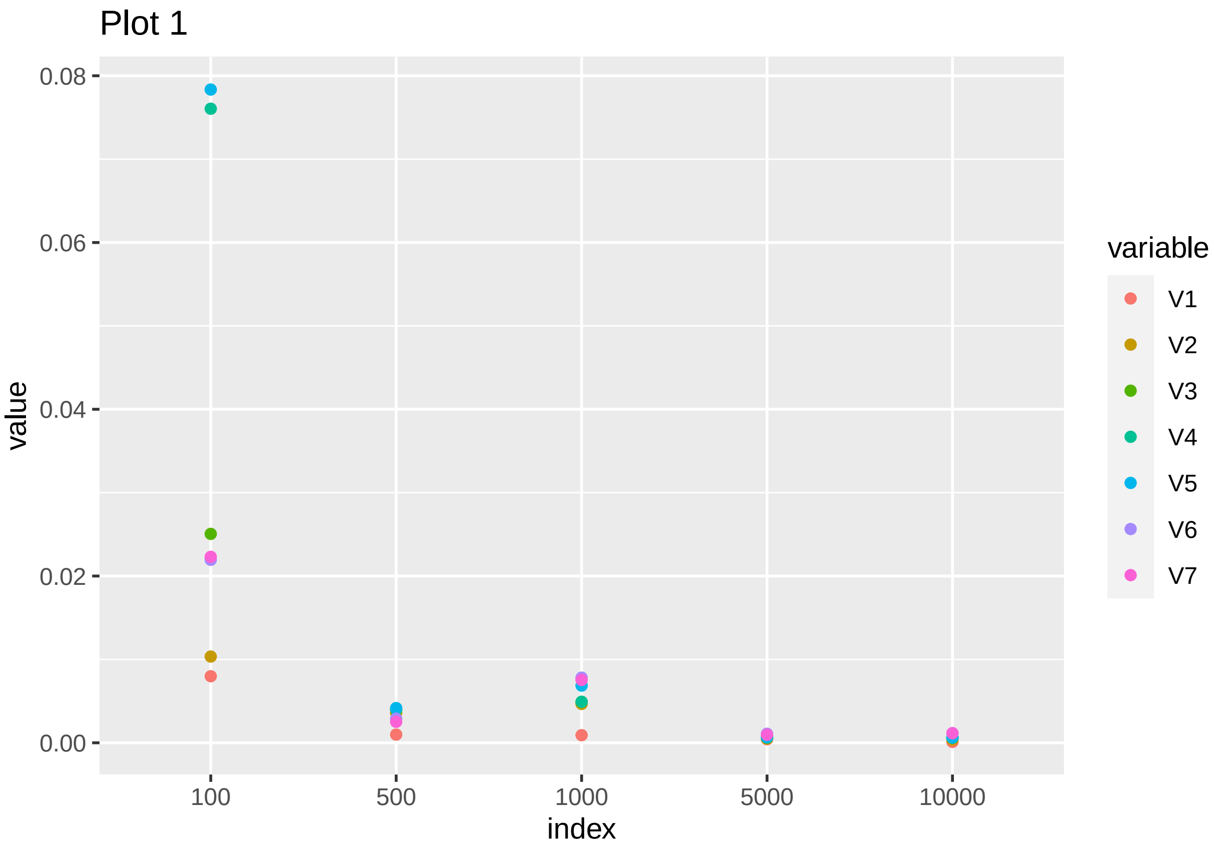
The plots are as follows:

```
df1 = melt(total_res[[1]])
```

```
## No id variables; using all as measure variables
```

```
df1$index = rep(as.factor(sample_size),7)

ggplot(df1, aes(index, value,color = variable)) +
  geom_point() +
  ggtitle("Plot 1")
```

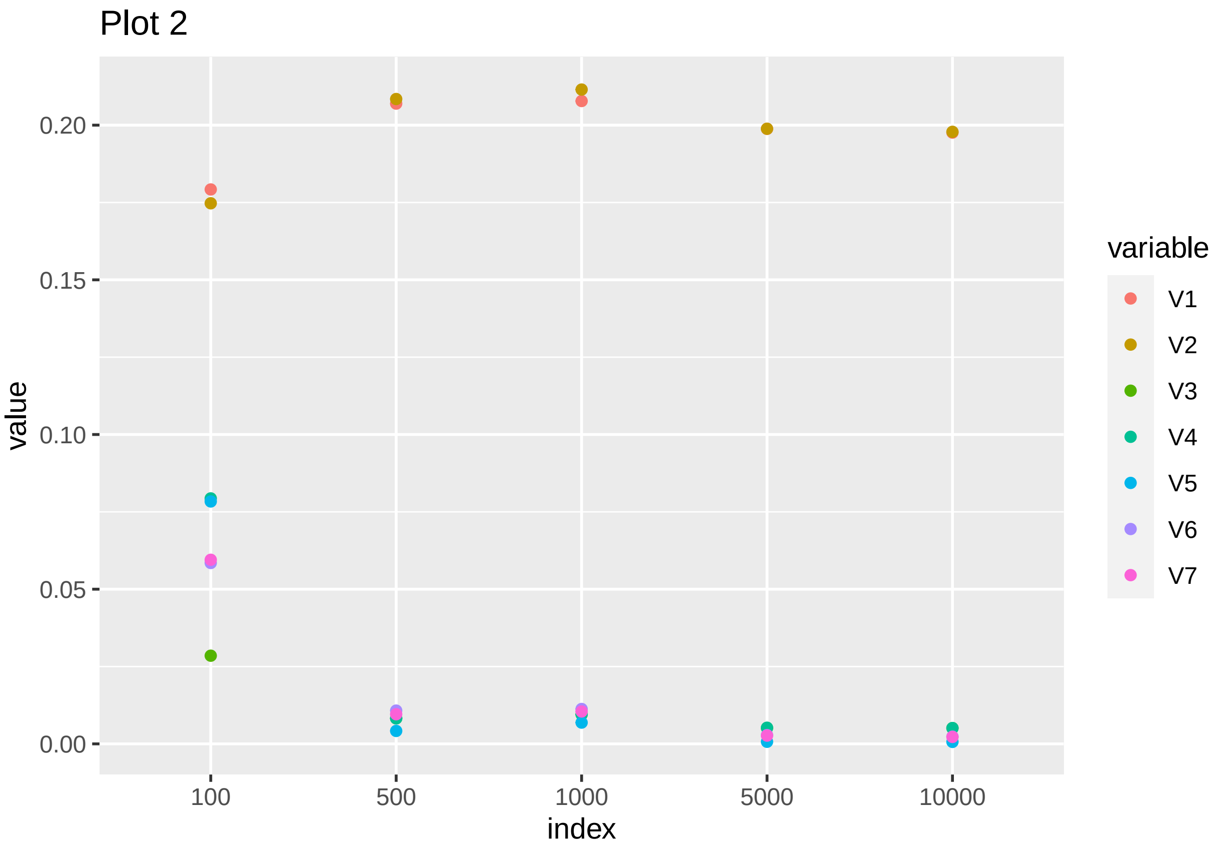


```
df2 = melt(total_res[[2]])
```

```
## No id variables; using all as measure variables
```

```
df2$index = rep(as.factor(sample_size),7)

ggplot(df2, aes(index, value,color = variable)) +
  geom_point() +
  ggtitle("Plot 2")
```

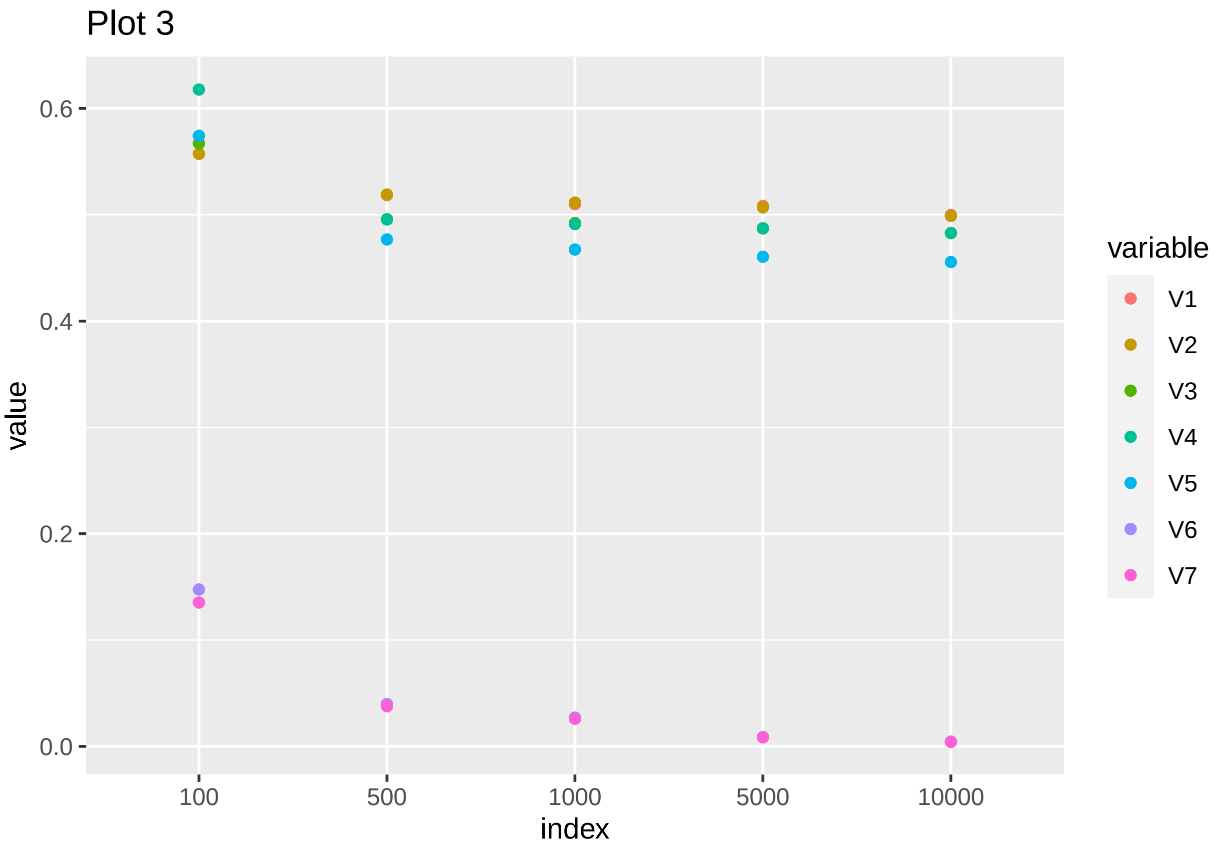


```
df3 = melt(total_res[[3]])
```

```
## No id variables; using all as measure variables
```

```
df3$index = rep(as.factor(sample_size),7)

ggplot(df3, aes(index, value,color = variable)) +
  geom_point()  +
  ggtitle("Plot 3")
```



In the above plots, Plot 1-3 each shows the MSE for different models with different datasets. V1 stands for linear regression, V2-V5 are polynomial regressions with degrees 2-5, V6 is the box kernel NW model, V7 is the gaussian kernel NW model. Plot 1 has the 2x data, plot 2 indicates $sin(2\pi x)$ data, plot 3 indicates $sin(30x)$ data.

The plots indicate that in general, an increase in sample size leads to decrease in the MSE. But it also depends on the data. For exmaple, when data is linear, all models work and the MSEs converge. However, when data is not linear, as in case 2, low-degree (degree = 2) polynomial model and simple linear regression would fail to converge to 0 and instead converge to 0.2. Similarly, in case 3, all polynomial models and the simple linear regression model would fail to converge to 0 and instead to somewhere between 0.45 and 0.5. The NW methods, however, will always converge to 0 regardless of the dataset. Moreover, sometimes more complex model (high degree polynomial) will actually increase the MSE if relationship is simple(linear).

Q4

Attempted, but not solved. So did not include the code.

Q4.a

Q4.b

Q 3

(a) Let $I\{x_{j,n} \in A_{j,n}\} = I\left\{\frac{j-1}{k(n)} < X \leq \frac{j}{K(n)}\right\}$

Sime X are deterministic, ε_i iid Normal,

$$L(f(x)) = \prod_{i=1}^{n} P(y_{i,n} : f)$$

(where $y_{i,n} \sim N(f(x_{i,n}), \sigma_\varepsilon^2)$)

$$L(f(x)) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}} e^{-\frac{(y_{i,n} - f(x_{i,n}))^2}{2\sigma_\varepsilon^2}}$$

$$\ell(f(x)) = \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}} - \frac{1}{2\sigma_\varepsilon^2}(y_{i,n} - f(x_{i,n}))^2$$

$$= \frac{n}{\sqrt{2\pi\sigma_\varepsilon^2}} - \frac{1}{2\sigma_\varepsilon^2}\sum_{i=1}^{n}(y_{i,n} - f(x_{i,n}))^2$$

To maximize $L(f(x))$ is equivalent to to minimize $\sum_{i=1}^{n}(y_{i,n} - f(x_{i,n}))^2$.

$$\operatorname*{argmin}_{C_i \in \mathbb{R}} \sum_{i=1}^{n}(y_{i,n} - f(x_{i,n}))^2$$

$$= \operatorname*{argmin}_{C_i \in \mathbb{R}} \sum_{i=1}^{n}\left(y_{i,n} - \sum_{j=1}^{k(n)} C_j I(x_{i,n} \in A_{j,n})\right)^2$$

$$= \operatorname*{argmin}_{C_1 \ldots C_{k(n)}} \left[\sum_{i=1}^{n}(y_{i,n} - C_1)^2 + \sum_{i=1}^{n}(y_{i,n} - C_2)^2 \ldots + \sum_{i=1}^{n}(y_{i,n} - C_{k(n)})^2\right]$$

$$= \sum_{j=1}^{k(n)} \left[\underbrace{\operatorname*{argmin}_{C_j \in \mathbb{R}} \sum_{i, x_{i,n} \in A_{j,n}} (y_{i,n} - C_j)^2}_{\swarrow}\right]$$

For each term,

$$f(C_i) = \sum_{i; x_{i,n} \in A_{i,n}} (y_{i,n} - C_i)^2$$

$$f'(C_i) = -2\sum_{i; x_{i,n} \in A_{i,n}} (y_{i,n} - C_i)^2 = 0$$

$$\sum \hat{C}_i = \sum y_{i,n} \cdot I(x_{i,n} \in A_{j,n})$$

$$\sum \hat{C}_i \cdot I(x_{i,n} \in A_{j,n}) = \sum y_{i,n} \cdot I(x_{i,n} \in A_{j,n})$$

$$\hat{C}_i = \frac{\sum_{i=1}^{n} y_{i,n} \cdot I(x_{i,n} \in A_{j,n})}{\sum_{i=1}^{n} I(x_{i,n} \in A_{j,n})}$$

$\Rightarrow$

Therefore,

$$\hat{f}_n(x) = \sum_{j=1}^{k(n)} \left(\operatorname*{argmin}_{C_j \in \mathbb{R}} \sum_{i, x_{i,n} \in A_{j,n}} (y_{i,n} - C_j)^2\right)$$

$$= \sum_{j=1}^{k(n)} \hat{C}_i \cdot I(x \in A_{j,n})$$

$$= \sum_{j=1}^{k(n)} \left(\frac{\sum_{i=1}^{n} y_{i,n} I(x_{i,n} \in A_{j,n})}{\sum_{i=1}^{n} I\{x_{i,n} \in A_{j,n}\}} I(x \in A_{j,n})\right)$$

(b)

$$E[(\hat{f}_n(x_0) - f(x_0))^2] = \text{Bias}^2 + \text{Variance}$$

$$|\text{Bias}| = |E(\hat{f}(x_0)) - f(x_0)|$$

$$= \left| E\left(\sum_{j=1}^{k(n)} \left(\frac{\sum_{i=1}^{n} Y_{i,n} I\{(x_{i,n} \in A_{j,n}\}}{\sum_{i=1}^{n} I\{x_{i,n} \in A_{j,n}\}}\right) I\{x \in A_{j,n}\}\right) - f(x_0)\right|$$

$$= \left| E\left(\frac{\sum_{i=1}^{n} Y_{i,n} I\{x_{i,n} \in A_{j,n}\}}{k(n)}\right) - f(x_0)\right|$$

$$= \left|\frac{\sum_{i \in R} f(x_i)}{k(n)} - f(x_0)\right|$$

$$= \left|\sum_{i \in R}\left(\frac{f(x_i) - f(x_0)}{K(n)}\right)\right|$$

$$\leq \frac{1}{K(n)} \sum_{i \in R} |f(x_i) - f(x_0)|$$

$$\leq \frac{c}{K(n)}$$

Thus,

$$\text{Bias}^2 \leq \frac{c^2}{K(n)^2}$$

~~Variance := Var($\frac{k(n)}{n} \cdot \sum_{i=1}^{n} Y_{i,n} I\{x_{i,n} \in A_{j,n}\}$)~~

$$\text{Variance} = \text{Var}\left(\sum_{j=1}^{k(n)}\left(\frac{\sum_{i=1}^{n} Y_{i,n} I\{(x_{i,n} \in A_{j,n}\}}{\sum_{i=1}^{n} I\{x_{i,n} \in A_{j,n}\}} \cdot I(x \in A_{j,n}\right)\right)$$

$$= \sum_{i=1}^{k(n)} \text{Var}\left(\frac{\sum_{i=1}^{n} Y_{i,n} I\{x_{i,n} \in A_{j,n}\}}{\sum_{i=1}^{n} I\{x_{i,n} \in A_{j,n}\}} \cdot I(x \in A_{j,n})\right)$$

~~$= K(n) \cdot \text{Var}($~~

$$= \sum_{i=1}^{K(n)} \text{Var}\left(\frac{\sum_{i=1}^{n} Y_{i,n} I\{x_{i,n} \in A_{j,n}\}}{n}\right)$$

$$= K(n) \cdot \frac{1}{n^2} \cdot \sum_{i=1}^{n} \text{Var}(Y_i)$$

$$= \frac{K(n)}{n} \cdot \sigma_\varepsilon^2$$

Thus,

$$E[(\hat{f}_n(x_0) - f(x_0))^2] = O\left(\frac{\sigma_\varepsilon^2 K(n)}{n} + \frac{c^2}{K(n)^2}\right)$$

(c) Taking derivative with respect to $K(n)$, we have

$$\frac{\sigma_\varepsilon^2}{n} \cancel{+} - \frac{2c^2}{2K(n)^3} = 0$$

Basically, $n = K(n)^3$
or $K(n) = n^{1/3}$

$$\cancel{O}\left(\frac{\sigma_\varepsilon^2 K(n)}{n} + \frac{c^2}{K(n)^2}\right) \equiv n^{-2/3}$$

and the rate is $O(n^{-2/3})$ which is identical to our box kernel.