

$$1a. P(X=k|\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

$$\begin{aligned} L(\lambda) &= \sum_{i=1}^n (x_i \log \lambda - \lambda - \log x_i!) \\ &= \log \lambda \sum_{i=1}^n x_i - n\lambda - \sum_{i=1}^n \log x_i! \end{aligned}$$

find max by setting log lik to zero

$$L'(\lambda) = \frac{1}{\lambda} \sum_{i=1}^n k_i - n = 0$$

$$\boxed{\hat{\lambda} = \bar{x}}$$

$$\begin{aligned} 1b. \text{Bias}(\hat{\lambda} - \lambda) &= E[\hat{\lambda} - \lambda | \lambda] \\ &= E(\hat{\lambda} | \lambda) - \lambda \\ &= \frac{\sum_{i=1}^n E(x_i | \lambda)}{n} - \lambda \\ &= \frac{\sum_{i=1}^n \lambda}{n} - \lambda \\ &= \lambda - \lambda \end{aligned}$$

$= 0$ thus, the estimate is unbiased]

$$1c \quad \text{var}(\hat{\lambda} | \lambda) = \text{var}\left(\frac{1}{n} \sum_{i=1}^n x_i | \lambda\right)$$

$$\begin{aligned} &= \frac{1}{n^2} \sum_{i=1}^n \text{var}(x_i) \\ &= \frac{n\lambda}{n^2} \\ &= \boxed{\frac{\lambda}{n}} \end{aligned}$$

$$\text{Id. } E[(\lambda - \hat{\lambda})^2 | \lambda] = \text{var}(\hat{\lambda} | \lambda)$$

$$= \frac{\lambda}{n}$$

for $\frac{\lambda}{n} < 0.1, \lambda \in [10, 20]$

for $n > \frac{20}{0.1} = \boxed{200 \text{ samples needed}}$

- 3a i) not linear \rightarrow it's quadratic
ii) if $B_0 = 1$, $B_1 = 2$, and $B_2 = 0$, then
no undermodeling
- 3b i) not linear \rightarrow linear divided by linear \neq linear
ii) there is undermodeling because a, x can
never collapse to just "1"
- 3c i) not linear \rightarrow it's quadratic
ii) there is undermodeling since the term
" $-2x_1 x_2$ " is never represented in the
model class.

4a Let $cv = \text{cancer volume}$

let $age = \text{age}$

let $ct = \text{cancer type}$

$$\text{model 1: } \hat{y} = \beta_0 + cv\beta_1$$

$$\text{model 2: } \hat{y} = \beta_0 + cv\beta_1 + age\beta_2$$

$$\text{model 3: } \hat{y}_{\text{typeI}} = \beta_{\text{typeI}} + cv\beta_1 + age\beta_2$$

} depends on
which type of
cancer

$$\hat{y}_{\text{typeII}} = \beta_{\text{typeII}} + cv\beta_1 + age\beta_2$$

4b. model 1 = 2 parameters

model 2 = 3 parameters

model 3 = 4 parameters, so model 3 the most complex

4c. model 1: $A = \begin{bmatrix} 1 & 0.7 \\ 1 & 1.3 \\ 1 & 1.6 \end{bmatrix}$

model 2: $A = \begin{bmatrix} 1 & 0.7 & 55 \\ 1 & 1.3 & 65 \\ 1 & 1.6 & 70 \end{bmatrix}$

model 3: $A = \begin{bmatrix} 1 & 0.7 & 55 \\ 1 & 1.3 & 65 \\ 1 & 1.6 & 70 \end{bmatrix}$

111

4d Based on lowest test error, model 3 should be chosen since it has the lowest mean test and training MSE.

Based on the one standard error rule, model 2 should be chosen

Homework 2

Kitu Komya

May 2, 2018

2a

```
set.seed(2018)

# create B0 and B1
B <- c(rep(1, 500), rep(10, 500))
B <- as.data.frame(B)

# generate Ntest = 1000 iids
for (i in 1:nrow(B))
{
  B$y[i] <- rexp(n = 1, rate = B$B[i])
}

# map classifier: calculating posterior probabilities
for (i in 1:nrow(B))
{
  B$y0[i] <- 0.5 * rexp(n = 1, rate = B$B[i])
  B$y1[i] <- 0.5 * rexp(n = 1, rate = B$B[i])
  B$choose[i] <- ifelse(B$y0[i] > B$y1[i], 0, 1)
}

# classification error rate
a <- length(which(B$choose[1:500] == 1)) # how many guessed 1 when should have been 0
b <- length(which(B$choose[501:1000] == 0)) # how many guessed 0 when should have been 1
error <- (a + b)/1000
error*100

## [1] 47.4
```

Classification error rate is 47.4%

2c

```
# create y
y <- c(rep(0, 500), rep(1, 500))
y <- as.data.frame(y)

# map classifier: calculating posterior probabilities
for (i in 1:500)
{
  y$x[i] <- rexp(n = 1, rate = 1)
```