

1a.  $\begin{pmatrix} H \\ T \end{pmatrix} \begin{pmatrix} H \\ T \end{pmatrix} \begin{pmatrix} H \\ T \end{pmatrix} \begin{pmatrix} H \\ H \end{pmatrix} \begin{pmatrix} T \\ T \end{pmatrix}$

$$\begin{aligned} P(\text{head}) &= P(\text{double head})P(\text{head}|\text{double head}) + P(\text{double tail})P(\text{head}|\text{double tail}) \\ &\quad + P(\text{unbiased coin})P(\text{head}|\text{unbiased coin}) \\ &= (1/5 \times 1/2) + (1/5 \times 0/2) + (3/5 \times 1/2) \\ &= 5/10 = \boxed{0.5} \end{aligned}$$

1b. 
$$\begin{aligned} P(\text{double head}|\text{head}) &= \frac{P(\text{double head})P(\text{head}|\text{double head})}{P(\text{head})} \\ &= \frac{(1/5) \times (1/2)}{(1/2)} \\ &= 2/5 \\ &= \boxed{0.4} \end{aligned}$$

2a. 
$$\frac{p(x|y=1)}{p(x|y=0)} = \frac{\frac{3}{2}x^2}{\frac{1}{2}} = 3x^2$$

when  $3x^2 < 1$ , LRT says  $\hat{y}=0$   
when  $3x^2 > 1$ , LRT says  $\hat{y}=1$

2b. 
$$\begin{aligned} x < \sqrt{1/3} &\rightarrow x < 1/\sqrt{3} \\ x > \sqrt{1/3} &\rightarrow x > 1/\sqrt{3} \end{aligned}$$

so, 
$$\hat{y} = \begin{cases} 0 & \text{when } -1/\sqrt{3} \leq x \leq 1/\sqrt{3} \\ 1 & \text{when } 1/\sqrt{3} < x < 1 \text{ or } -1 \leq x \leq -1/\sqrt{3} \end{cases}$$

2b.  $P_{MD}$  = type II error,  $P_{FA}$  = type I error

$$P_{MD} = P(\hat{y}=0|y=1) = \int_{-1/\sqrt{3}}^{1/\sqrt{3}} \frac{3}{2}x^2 dx = \boxed{0.19245}$$

$$\begin{aligned} P_{FA} &= P(\hat{y}=1|y=0) = \int_{-1}^{-1/\sqrt{3}} \frac{1}{2} dx + \int_{1/\sqrt{3}}^1 \frac{1}{2} dx \\ &= \frac{1}{6}(3-\sqrt{3}) + \frac{1}{6}(3-\sqrt{3}) = \boxed{0.422} \end{aligned}$$

$$\begin{aligned}
 2c. \quad E(L(\hat{y}, y)) &= L(0, 1) p(\hat{y}=0 | y=1) p(y=1) \\
 &\quad + L(1, 0) p(\hat{y}=1 | y=0) p(y=0) \\
 &= c_2 0.19245 \cdot q + c_1 0.422 \cdot (1-q) \\
 &= \boxed{0.19245 q c_2 - 0.422 q c_1 + 0.422 c_1}
 \end{aligned}$$

$$4a. \quad P(y=1) = 0.3$$

$$P(y=0) = 0.7$$

$$P(x|y=0) = N(0, 0.25) = \frac{1}{0.25\sqrt{2\pi}} e^{-\frac{(x-0)^2}{2(0.25)^2}}$$

$$P(x|y=1) = N(1, 0.25) = \frac{1}{0.25\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2(0.25)^2}}$$

$$\begin{aligned}
 \frac{P(x|y=1)}{P(x|y=0)} &= \frac{e^{-\frac{(x-1)^2}{2(0.25)^2}}}{e^{-\frac{(x-0)^2}{2(0.25)^2}}} = \frac{e^{-8(x-1)^2}}{e^{-8x^2}}
 \end{aligned}$$

$$\log \left( \frac{P(x|y=1)}{P(x|y=0)} \right) = \log \left( e^{8x^2 - 8(x-1)^2} \right)$$

$$= \log(e^{16x-8})$$

$$= 16x - 8, \text{ assuming } x > 0$$

$$16x - 8 > 0 \text{ when}$$

$$x > 1/2$$

thus,

$$\hat{y}_{ML} = \begin{cases} 0 & \text{when } x < 1/2 \\ 1 & \text{when } x > 1/2 \end{cases}$$

$$4b. \quad p(y=0/x) = \frac{p(x|y=0)p(y=0)}{p(x)}$$

$$= \frac{\frac{4}{\sqrt{2\pi}} e^{-x^2/2(0.25)^2} \cdot 0.7}{p(x)}$$

$$p(y=1/x) = \frac{p(x|y=1)p(y=1)}{p(x)}$$

$$= \frac{\frac{4}{\sqrt{2\pi}} e^{-(x-1)^2/2(0.25)^2} \cdot 0.3}{p(x)}$$

$$\text{check } \frac{p(y=1/x)}{p(y=0/x)} = \frac{\frac{4}{\sqrt{2\pi}} e^{-(x-1)^2/2(0.25)^2} \cdot 0.3}{\frac{4}{\sqrt{2\pi}} e^{-x^2/2(0.25)^2} \cdot 0.7}$$

$$= e^{-\frac{x^2}{2(0.25)^2} + \frac{(x-1)^2}{2(0.25)^2}} > \frac{3}{7}$$

$$= \frac{-x^2}{2(0.25)^2} + \frac{(x-1)^2}{2(0.25)^2} > \ln\left(\frac{3}{7}\right)$$

$$= -x^2 + x^2 - 2x + 1 > (\ln 3 - \ln 7) 2(0.25)^2$$

$$-2x > -1.106$$

$$x < 0.553$$

$$\hat{y}_{\text{MAP}} = \begin{cases} 0 & \text{when } x < 0.553 \\ 1 & \text{when } x > 0.553 \end{cases}$$

# Stats C161/C261 - Homework 1

*Kitu Komya*

*April 17, 2018*

## Problem 3

### part a

```
# set seed
set.seed(2018)

# initialize 1000 x and y datapoints
x <- c(1:1000)
y <- numeric(1000)

# create slope coefficients
B0 <- rnorm(n = 1000, mean = 0, sd = 75) # wi

# create the 1000 y datapoints
y <- x + B0

# merge variables into a dataframe
df <- data.frame(x, y)
```

### part b

```
# divide into training and testing set
train <- df[sample(nrow(df), 500), ]
test <- df[!df$x %in% train$x, ]
```

### part c

```
# linear regression on training
linear <- lm(formula = y ~ x, data = train)

# compute error on testing set: MSE is 6138.18
mean((test$y - predict.lm(linear, test))^2)
```

```
## [1] 6138.18
```

### part d

```
# quadratic regression on training
quadratic <- lm(formula = y ~ poly(x, 2), data = train)
```

```

# compute error on testing set: MSE is 6100.209
mean((test$y - predict.lm(quadratic, test))^2)

## [1] 6100.209

# cubic regression on training
cubic <- lm(formula = y ~ poly(x, 3), data = train)

# compute error on testing set: MSE is 6201.891
mean((test$y - predict.lm(cubic, test))^2)

## [1] 6201.891

```

## part e

```

### re-doing steps for flow_data.csv ###
flow <- read.csv(file = "flow_data.csv")

# divide into training and testing set
train <- flow[sample(nrow(flow), 50), ]
test <- flow[!flow$X %in% train$X, ]

# linear regression on training
linear <- lm(formula = flow ~ radius, data = train)

# compute error on testing set: MSE is 0.1831462
mean((test$flow - predict.lm(linear, test))^2)

## [1] 0.1831462

# quadratic regression on training
quadratic <- lm(formula = flow ~ poly(radius, 2), data = train)

# compute error on testing set: MSE is 0.06036196
mean((test$flow - predict.lm(quadratic, test))^2)

## [1] 0.06036196

# cubic regression on training
cubic <- lm(formula = flow ~ poly(radius, 3), data = train)

# compute error on testing set: MSE is 0.06608234
mean((test$flow - predict.lm(cubic, test))^2)

## [1] 0.06608234

```

The quadratic model fits the data the best, since its testing MSE is the lowest.

## part f

Yes, this makes intuitive sense since the volume of any cylinder is proportionate to/varies with  $r$  squared. Thus, a quadratic relationship makes sense to our data. This homework helps us to realize that whenever we choose models, we should allow the physical principles to help us in making our model selection in order to avoid underfitting and overfitting.

## Problem 4

### part c

```
# generate 10^4 samples total. 7000 for y = 0 and 3000 for y = 1
y0 <- rnorm(n = 7000, mean = 0, sd = sqrt(0.25))
y1 <- rnorm(n = 3000, mean = 1, sd = sqrt(0.25))
```

### part d

```
# run ML classifier and find error rate: 0.1585
(length(y0) - length(which(y0 < 0.5)) + length(which(y1 < 0.5)))/10000
```

```
## [1] 0.1547
```

```
# run MAP classifier and find error rate: 0.1512
(length(y0) - length(which(y0 < 0.553)) + length(which(y1 < 0.553)))/10000
```

```
## [1] 0.1477
```

ML has an error rate of 0.1585, while MAP has an error rate of 0.1512. The error rate is calculated summing the number of missed detections and number of false alarms. These numbers lead us to believe that MAP classified the data better than ML.

### part e

```
# plot error rates from between 0.2 and 0.7 arbitrarily chosen
x <- seq(0.1, 0.9, by = 0.05)
y <- numeric(length(x))

# add in error rate for each step
for (i in 1:17)
{
  y[i] <- (length(y0) - length(which(y0 < x[i])) + length(which(y1 < x[i])))/10000
}
```

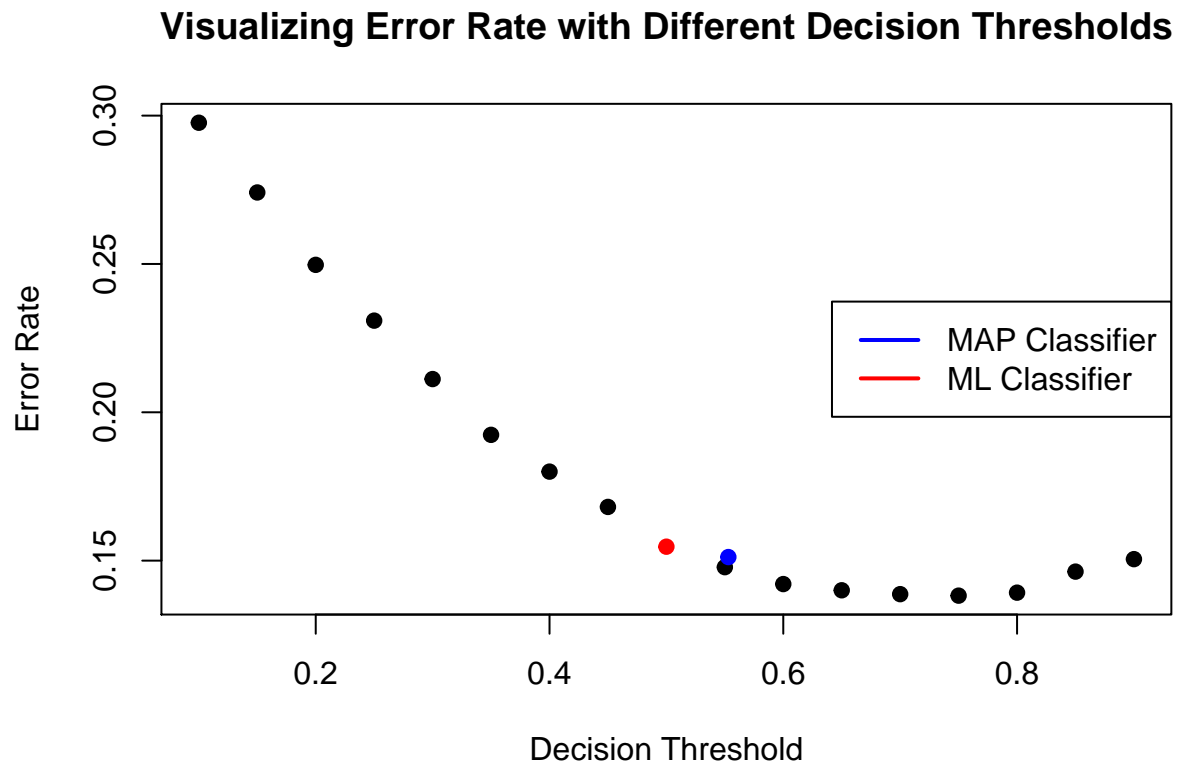
```
# create dataframe
xy <- data.frame(x, y)
```

```
# add MAP classifier (ML already added in)
ML <- data.frame(0.553, 0.1512)
names(ML) <- c("x", "y")
xy <- rbind(xy, ML)
```

```
# add colors
xy$color <- "black"
xy$color <- ifelse(xy$x == 0.500, "red", xy$color) # for ML classifier
xy$color <- ifelse(xy$x == 0.553, "blue", xy$color) # for MAP classifier
```

```
# plot the rates
plot(xy$x, xy$y, main = "Visualizing Error Rate with Different Decision Thresholds", xlab = "Decision Threshold", ylab = "Error Rate", col = xy$color)
```

```
legend("right", legend = c("MAP Classifier", "ML Classifier"),  
      lty = c(1,1), lwd = c(2, 2), col = c("blue", "red"))
```



In visualizing the error rates among varying decision thresholds, we see that the ML and MAP classifiers are close to the minima, but not entirely there. It seems like a decision threshold near 0.7 will work best with our data. In that case, the MAP classifier still did a better job than the ML classifier.