HW 03

Samuel Swain

2022-11-22

Question 1

1(a)

Likelihood and log(Likelihoods):

$$L(\theta_i \mid n,\, y_i) = \binom{n}{y_i}\Pi_{i=1}^{6}\theta_i^{y_i};\; n = 100$$
$$ln(L_i) = ln(\binom{n}{y_i}) + \Sigma_{i=1}^{6} y_i\theta_i$$
$$y_6 = n - \Sigma_{i=1}^{5} y_i = 19,\; \theta_6 = 1 - \Sigma_{i=1}^{5}\theta_i$$
$$\alpha = ln(L_i) = ln(\binom{n}{y_i}) + \Sigma_{i=1}^{5} y_i\theta_i + 19(\theta_6)$$

Differentiate w.r.t. $\theta' s$:

$$\frac{d\alpha}{d\theta_1} = \frac{y_1}{\theta_1} - \frac{19}{\theta_6} = \frac{18}{\theta_1} - \frac{19}{\theta_6} = 0$$
$$\frac{d\alpha}{d\theta_2} = \frac{11}{\theta_2} - \frac{19}{\theta_6} = 0$$
$$\frac{d\alpha}{d\theta_3} = \frac{9}{\theta_3} - \frac{19}{\theta_6} = 0$$
$$\frac{d\alpha}{d\theta_4} = \frac{25}{\theta_4} - \frac{19}{\theta_6} = 0$$
$$\frac{d\alpha}{d\theta_5} = \frac{18}{\theta_5} - \frac{19}{\theta_6} = 0$$

After solving:

$$18\theta_6 = 19\theta_1$$
$$11\theta_6 = 19\theta_2$$
$$9\theta_6 = 19\theta_3$$
$$25\theta_6 = 19\theta_4$$
$$18\theta_6 = 19\theta_5$$
$$\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 + \theta_6 = 1$$

We attain:

$$\theta_1 = \frac{18}{100},\, \theta_2 = \frac{11}{100},\, \theta_3 = \frac{9}{100}$$
$$\theta_4 = \frac{25}{100},\, \theta_5 = \frac{18}{100},\, \theta_6 = \frac{19}{100}$$

1(b)

Prior:

$$P(\overrightarrow{\theta} \mid \overrightarrow{\alpha} = 1)\frac{1}{B(\overrightarrow{\alpha} = 1)}\Pi_{i=1}^{12}\theta_i^{\alpha_i - 1},\; \alpha = 1$$

Posterior:

$$P(\overrightarrow{\theta} \mid delta) \propto P(delta \mid \overrightarrow{\theta}) * P(\overrightarrow{\theta} \mid \alpha = 1)$$
$$P(delta \mid \overrightarrow{\theta}) = \Pi_{i=1}^{6}\binom{100}{y_i}\theta^{y_i}$$
$$P(\overrightarrow{\theta} \mid \alpha = 1) = \frac{1}{B(\overrightarrow{\alpha} = 1)}$$

log(Posterior):

$$\alpha = log(P(delta \mid \overrightarrow{\theta})) \propto \frac{1}{B(\alpha = 1)} + \Sigma_{i=1}^{6}(\frac{100}{y_i}) + y_i * log(\theta_i)$$

Differentiate w.r.t. $\theta' s$:

$$\theta_1 = \frac{18}{100}, \theta_2 = \frac{11}{100}, \theta_3 = \frac{9}{100}$$
$$\theta_4 = \frac{25}{100}, \theta_5 = \frac{18}{100}, \theta_6 = \frac{19}{100}$$

Question 2

2(a)

Biased

$$P(x_0,\, x_1,\, \ldots,\, x_n : \pi_0,\, \pi_1,\, \ldots,\, \pi_n)$$
$$= P(x_n \mid \pi_n = bias) * P(\pi_n = bias \mid \pi_{n-1} = bias) *$$
$$\ldots$$
$$* P(x_1 \mid \pi_1 = bias) * P(\pi_1 = bias \mid \pi_0 = bias)$$

- The biased probability is 6.2929774^{-6}

Fair Likelihood:

$$P(x_0,\, x_1,\, \ldots,\, x_n : \pi_0,\, \pi_1,\, \ldots,\, \pi_n)$$
$$= P(x_n \mid \pi_n = fair) * P(\pi_n = fair \mid \pi_{n-1} = fair) *$$
$$\ldots$$
$$* P(x_1 \mid \pi_1 = fair) * P(\pi_1 = fair \mid \pi_0 = fair)$$

- The fair probability is 2.5431315^{-6}

As we can see, the biased probability is higher. Thus, it's more likely that the hidden states are not being fair.

2(b)

Get probability vectors

```
states = expand.grid(0:1, 0:1, 0:1, 0:1, 0:1, 0:1)
n_2b = nrow(states)

unfair_probs = c(4/13, 4/13, 2/13, 2/13, 2/13, 2/13)
fair_probs = c(rep(1/6, 6))
```

Calc probs

```
probs = c()
for (i in 1:n_2b) {
  p_now = 1 * 0.5

  if(states[i,1]==0){
    p_now = p_now*fair_probs[1]
  }
  else{
    p_now = p_now*unfair_probs[1]
  }

  for(j in 2:6){
    if(states[i,j]!=states[i,j-1]){
      p_now = p_now * 0.25
    }
    else{
      p_now = p_now *0.75
    }

    if(states[i,j]==0){
      p_now = p_now * fair_probs[j]
    }
    else{
      p_now = p_now * unfair_probs[j]
    }
  }
  probs = c(probs,p_now)
}
```

Get and print max probs

The combination of probabilities with the maximum likelihood is the following:

```
print(states[which.max(probs), ])
```

```
##    Var1 Var2 Var3 Var4 Var5 Var6
## 64    1    1    1    1    1    1
```

- The maximum likelihood estimate is 6.2929774^{-6}

Question 3

3(a)

Read in data, get train/test sets

```
df <- read.csv(file = 'gradAdmit.csv', header = T)

set.seed(400)
n = nrow(df)
# Get 20% for test set
sample = sample.int(n = n, size = floor(.2*n), replace = F)
train_wrong_index = df[-sample,]
test = df[sample,]
train = train_wrong_index
rownames(train) = 1:nrow(train_wrong_index)
```

Calculate balance of each dataset

30.9375 percent of the students in the training data were admitted and 35 in the testing data set were admitted.

3(b)

Train best model, get test predictions

```
library(e1071)
library(MLmetrics)
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

```
svm_final <- svm(factor(admit) ~ gre + gpa + rank,
                data = train,
                scale = T,
                probability = T,
                kernel = "polynomial",
                cost = 1,
                degree = 5,
                gamma = 1,
                coef0 = 1
                )

pred_final <- predict(svm_final,
                     test,
                     decision.values = F,
                     probability = F)
```

Get requested metrics

```
precision = Precision(test$admit, pred_final)
recall = Recall(test$admit, pred_final)
specificity = Specificity(test$admit, pred_final)
```

- Precision: 0.6521739
- Recall: 0.8653846
- Specificity: 0.1428571

3(c)

Get percent increase needed

```
diff_train = sum(train$admit==0)-sum(train$admit==1)
pct_over = diff_train/abs(sum(train$admit==1))*100
```

We need 123.2323232 percent more admitted observations to have a balanced data set.

Oversample using SMOTe

```
library("DMwR")
```

```
## Loading required package: lattice
```

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
train$admit = as.factor(train$admit)
train_SMOTe = SMOTE(admit ~ ., train, perc.over = pct_over)

table(train_SMOTe$admit)
```

```
##
##   0   1
## 198 198
```

3(d)

Train model and calculate metrics

```
# Model
svm_final <- svm(factor(admit) ~ gre + gpa + rank,
                data = train_SMOTe,
                scale = T,
                probability = T,
                kernel = "polynomial",
                cost = 1,
                degree = 5,
                gamma = 1,
                coef0 = 1
                )

# Predictions
pred_final_3d <- predict(svm_final,
                     test,
                     decision.values = F,
                     probability = F)

# Metrics
precision_3d = Precision(test$admit, pred_final_3d)
recall_3d = Recall(test$admit, pred_final_3d)
specificity_3d = Specificity(test$admit, pred_final_3d)
```

- Precision: 0.7291667
- Recall: 0.6730769
- Specificity: 0.5357143

Question 4

4(a)

```
set.seed(400)
lambdas = c(1)
results = c()
results_real = c()

for (i in lambdas){

  n = 10^8/i^2
  x = runif(n, 0, 1)
  y = -log(x)/i
  g = sin(y)/i

  results = append(results, sum(g)/n)
  results_real = append(results_real, (1 / (1+i^2)))
}
```

The probability of drawing a sample $x \geq 10\pi$ is 0

4(b)

From assignment 1 question 2 we get the following integral:

$$\int_0^\infty e^{-\lambda x} sin(x)\; dx = \frac{1}{1+\lambda^2} \Rightarrow \frac{1}{2}\; when\; \lambda = 1$$

We also know:

$$\int_0^{10\pi} e^{-\lambda x} sin(x)\; dx = \frac{1}{2} - \frac{e^{-10\pi}}{2}$$

Thus:

$$\int_{10\pi}^\infty e^{-\lambda x} sin(x)\; dx = \frac{e^{-10\pi}}{2}$$

4(c)

To find a $p^*(x)$ larger than $p(x)$ when $x \geq 10\pi$ and 0 when $x < 10\pi$, we can set $p^*(x)$ equal to $p(x)$ but shifted to the right. Therefore we end up with $p^*(x) = e^{10\pi - x}$.

4(d)

```
set.seed(400)
n = 10^6
d = rexp(n, 1)

total = 0
for (i in d) {
  total = total + sin(i) * exp(-10*pi)
}

estimate = total/n
```

The estimate is 1.1351975^{-14}. This is very close to the real value of 1.1355505^{-14}.