CSE 242 Homework 1

Sampson Mao

October 26, 2020

1 Conditional probability Review

If we remove 2 kings from a deck of 52 cards, there are 50 cards left over. Furthermore, the hearts and diamonds will have only 2 face cards each, giving them a total of 12 cards per suit.

a) $P(F) = \frac{2}{50} = \frac{1}{25}$

b) $P(B) = \frac{26}{50} = \frac{13}{25}$

c) $P(BF) = \frac{2}{50} = \frac{1}{25}$

d) $P(F|B) = \frac{2}{26} = \frac{1}{13}$

e) $P(B|F) = 1$

f) $P(BF) = P(F|B)P(B) = P(B|F)P(F)$

$\frac{1}{25} = \frac{1}{13} \cdot \frac{13}{25} = \frac{1}{25} \cdot 1$

g)

$$\begin{aligned}
P(H|\text{not a }2) &= \frac{P(\text{not a }2|\text{H})P(H)}{P(\text{not a }2)} \\
&= \frac{[1 - P(\text{is a }2|\text{H})]P(H)}{1 - P(\text{is a }2)} \\
&= \frac{\left(1 - \frac{1}{12}\right)\left(\frac{12}{50}\right)}{1 - \frac{4}{50}} \\
&= \frac{1}{46}
\end{aligned}$$

2 Linear Regression

a) I will be using Scikit-learn to solve this linear regression problem

b) Fitting the model:

[1]:

```python
import numpy as np
from sklearn import linear_model
```

[2]:

```python
data = np.array(
    [
    [3, 9, 2, 19],
    [6, 9, 1, 19],
    [7, 7, 7, 10],
    [8, 6, 4, 11],
    [1, 0, 8, -3]
    ]
              )
```

[3]:
```
X = data[:,:3]
y = data[:,3]
```

As the hint mentions, the default setting for the `LinearRegression()` object is `fit_intercept=TRUE`, so it will calculate the intercept. This means we don't need a column of ones for the first column, and `X` is just the first 3 columns in `data`.

[4]:
```
LR = linear_model.LinearRegression()
LR.fit(X, y)
```

[4]:
```
LinearRegression()
```

This model estimates the coefficients $\theta_1, \theta_2, \theta_3$ in thiat order to be

[5]:
```
LR.coef_
```

[5]:
```
array([-0.13429383,  1.84768377, -0.89658481])
```

and the intercept θ_0

[6]:
```
LR.intercept_
```

[6]:
```
4.360802923546112
```

With these coefficients, we can see how the model performs with the root-mean-square error

$RMSE = \sqrt{\frac{1}{N}\sum_i (f(x^{(i)}) - y)^2}$

[7]:
```
data_pred = LR.predict(X)
RMSE = np.sqrt(sum((data_pred-y)**2)/len(y))
RMSE
```

[7]:
```
0.1897092217331575
```

Using the built-in library function to check, we get the same result:

[8]:
```
from sklearn.metrics import mean_squared_error

np.sqrt(mean_squared_error(y, data_pred))
```

[8]:
```
0.1897092217331575
```

So on average our predictions are 0.19 from the actual value.

c) The model would predict the following label:

[9]:
```
new_X = np.array([[3, 3, 5]])
new_pred = LR.predict(new_X)
new_pred
```

[9]:
```
array([5.01804869])
```

d) If the examples are re-ordered, including the labels, there would be no change in the θ vector. This is because in the log-likelihood function, all of the data is added together (in the likelihood, they are multiplied). Order does not matter for these operations. I.e., the likelihood function is the same regardless of how we permute the rows. We are not changing the label assignments.

3 More Probability Review

First, we have

$$P(H) = \lambda$$

a) Obtaining the first head at the $(k+1)$th toss means that the k tosses before it were tails.

$$\begin{aligned} P(\{T\}_{i=1}^{k} H) &= P(T)^k P(H) \quad \text{(Independent tosses)} \\ &= (1-\lambda)^k \lambda \end{aligned}$$

b) Here we are saying the number of tosses K before the first heads is a random variable, so the expectation is

$$\begin{aligned} E(K) &= \sum_{k \in \mathbb{Z}^+} kP(K=k) \\ &= \sum_{k=1}^{\infty} k(1-\lambda)^k \lambda \\ &= \lambda \frac{1-\lambda}{[(1-\lambda)-1]^2} \quad \text{(Wolfram Alpha)} \\ &= \frac{1-\lambda}{\lambda} \end{aligned}$$

c) Now the random variable is the number of heads Y. If we toss the coin one time, then it has a probability mass function (pmf) of

$$P(Y=y) = \lambda^y (1-\lambda)^{1-y} \quad y \in \{0,1\}$$

If we have N independent tosses, then we have to multiply the probability of success y times, and multiply the probability of failure the remaining $N-y$ times. Also, the order of heads does not matter when we just want the number of heads, so the pmf is

$$P(Y=y) = \binom{N}{y}\lambda^y(1-\lambda)^{N-y} \quad y \in \{0,1,2,...\}$$

Thus, the expected value of the number of heads Y is

$$\begin{aligned}
E(Y) &= \sum_y yP(Y=y) \\
&= \sum_{y=0}^{\infty} y\binom{N}{y}\lambda^y(1-\lambda)^{N-y} \\
&= \sum_{y=0}^{\infty} y\frac{N!}{y!(N-y)!}\lambda^y(1-\lambda)^{N-y} \\
&= \sum_{y=0}^{\infty} \frac{N(N-1)!}{(y-1)!(N-y)!}\lambda^y(1-\lambda)^{N-y} \\
&= N\sum_{y=0}^{\infty} \binom{N-1}{y-1}\lambda^y(1-\lambda)^{N-y} && (N-1)-(y-1) = N-y \\
&= N\lambda\sum_{y=0}^{\infty} \binom{N-1}{y-1}\lambda^{y-1}(1-\lambda)^{(N-1)-(y-1)} \\
&= N\lambda
\end{aligned}$$

The last step comes from the fact that pmfs sum to 1 and the expression inside the sum was the pmf of an experiment with $N-1$ independent tosses with $y-1$ heads.

4 A Continuous Variable plus Bayes's Rule

a) If my assigned probability density of x is $f(x) = (n+1)x^n$ for $x \in (0,1]$, then my expected value is

$$\begin{aligned}
E(X) &= \int_{-\infty}^{\infty} xf(x)\,dx \\
&= \int_0^1 x(n+1)x^n\,dx \\
&= \frac{n+1}{n+2}x^{n+2}\Big|_0^1 \\
&= \frac{n+1}{n+2}
\end{aligned}$$

b) Denote the random variable Y for whether I heard thunder or not. $y=0$ means that I did not hear thunder and $y=1$ means that I did. Then the pmf for the probability of hearing thunder is

$$P(Y=y|x) = (1-x)^y x^{1-y} \quad y \in \{0,1\}$$

Then by Bayes' rule,

$$\begin{aligned} f(x|Y=y) &\propto P(Y=y|x)f(x) \\ &\propto (1-x)^y x^{1-y}(n+1)x^n \\ &\propto (1-x)^y x^{1-y+n} \end{aligned}$$

So if we did not hear thunder, i.e. $y = 0$, then the posterior density is proportional to

$$f(x|y=0) \propto x^{n+1} \quad x \in (0,1]$$

The normalizing factor is a factor c that when multiplied with the pdf, the pdf sums to 1 on its support set.

$$\int_0^1 cx^{n+1}\,dx = c\frac{x^{n+2}}{n+2}\bigg|_0^1 = \frac{c}{n+2} = 1 \implies c = n+2$$

So the posterior density is

$$f(x|y=1) = (n+2)x^{n+1} \quad x \in (0,1]$$