

Decision Theory

Assignment 1

Mohammed Bakheet (mohba508)

06/10/2020

Contents

Question 1	1
Solution	1
Question 2	3
Solution:	3
References	4
Appendix	4

Question 1

This is essentially Exercise 37 in Chapter 3 of “Winkler: An Introduction to Bayesian inference and decision, 2 nd ed.” A bank official is concerned about the rate at which the bank’s tellers provide service for their customers. He feels that all of the tellers work at about the same speed, which is either 30, 40 or 50 customers per hour. Furthermore, 40 customers per hour is twice as likely as each of the two other values, which are assumed to be equally likely. In order to obtain more information, the official observes all five tellers for a two-hour period, noting that 380 customers are served during that period.

Show that the posterior probabilities for the three possible speeds are approximately 0.000045, 0.99996 and 0.00000012 respectively.

Solution

We consider that all five tellers serve (30,40,50) customers per hour.

If all tellers work at the same speed, then we can consider $5 \times (30,40,50) = (150,200,250)$ total customers per hour will be served.

And since $t = 2$, this will give us a value of $\lambda_t = (300,400,500)$

We have monitored 380 customers for two hours, this gives us 190 customers per hour and 38 customers per teller.

Since λ_t is more than 20, we can use normal approximation:

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

For the first case (30): we calculate the posterior (we use R for faster calculation)

```
#using normal approximation
options("scipen"=100, "digits"=4)
lambdat= 300
y = 380
mu = lambdat
sigma2 = lambdat #since sigma = sqrt(lambda)

result30 = (2*pi*sigma2)^(-0.5)*exp(-(y-mu)^2/(2*sigma2))
posterior1 = result30*0.25
print(posterior1)
```

```
## [1] 0.0000001342
```

For the second case (40): we calculate the posterior

```
#using normal approximation
options("scipen"=100, "digits"=4)
lambdat= 400
y = 380
mu = lambdat
sigma2 = lambdat #since sigma = sqrt(lambda)

result40 = (2*pi*sigma2)^(-0.5)*exp(-(y-mu)^2/(2*sigma2))
posterior2 = result40*0.5
print(posterior2)
```

```
## [1] 0.006049
```

For the third case (50): we calculate the posterior

```
#using normal approximation
options("scipen"=100, "digits"=4)
lambdat= 500
y = 380
mu = lambdat
sigma2 = lambdat #since sigma = sqrt(lambda)

result50 = (2*pi*sigma2)^(-0.5)*exp(-(y-mu)^2/(2*sigma2))
posterior3 = result50*0.25
print(posterior3)
```

```
## [1] 0.000000002486
```

Then we can calculate the posterior probabilities:

```
options(digits = 22)
posterior_sum = posterior1+posterior2+posterior3
posteriors = c(posterior1/posterior_sum, posterior2/posterior_sum, posterior3/posterior_sum)
lambda_vector = c(30,40,50)

df = data.frame(posteriors, lambda_vector)
knitr::kable(df, booktabs = TRUE)
```

	posteriors	lambda_vector
	0.0000221871911812087081012582	30
	0.9999774018371490047485394825	40
	0.0000004109716698483781937769	50

If we normalize the big values of lambda and use it with Poisson distribution, it will give us this output:

```
lambda1 = 3
lambda2 = 4
lambda3 = 5
plambda1 = 0.25
plambda2 = 0.5
plambda3 = 0.25
r = 3.8
t = 1

factor30 = lambda1^r * exp(-t*lambda1) * plambda1 + lambda2^r * exp(-t*lambda2)*plambda2+lambda3^r * exp(-t*lambda3)*plambda3

problambda1 = (lambda1^r)*exp(-t*lambda1) * plambda1/factor30
problambda1

## [1] 0.2416520723393723590888

problambda2 = (lambda2^r)*exp(-t*lambda2) * plambda2/factor30
problambda2

## [1] 0.5305095872114534705233

problambda3 = (lambda3^r)*exp(-t*lambda3) * plambda3/factor30
problambda3

## [1] 0.2278383404491741703879
```

Question 2

Assume that data follows a beta distribution with parameters a and b , i.e. the probability density function is $f(y|a, b) = y^{a-1}(1-y)^{b-1}/B(a, b)$ where $B(a, b)$ is the beta function also equal to $\Gamma a \Gamma b / \Gamma(a+b)$. This could be the case when data consists of the analysed purity (in percent) of a narcotic substance in a number of seized packages.

- a) Show – by using the properties of distributions belonging to the exponential class – that the probability density function of a conjugate family to the family of beta distributions can be expressed as:

$$f'(a, b|\gamma, \delta, \theta) \propto \frac{e^{\gamma*a+\delta*b}}{(B(a, b))^\theta}$$

- b) Find the corresponding expression for the probability density function of the posterior distribution when a sample of size n has been obtained.

Solution:

First we have the function:

$$f'(a, b|\gamma, \delta, \theta) \propto \frac{e^{\gamma*a+\delta*b}}{(B(a, b))^\theta}$$

We can mimic the structure of the exponential class from this equation as follows:

$$= e^{\gamma*a+\delta*b-\theta*\ln B(a,b)}$$

$$\begin{aligned}
&= e^{\gamma * a + \delta * b - \theta * \ln(\frac{\Gamma(a) \Gamma(b)}{\Gamma(a+b)})} \\
&= e^{\gamma * a + \delta * b - \theta * \ln \Gamma a + \ln \Gamma b - \ln \Gamma(a+b)}
\end{aligned}$$

So, our prior for the exponential class family is:

$$= e^{\gamma * a + \delta * b - \theta * \ln \Gamma a + \ln \Gamma b - \ln \Gamma(a+b)}$$

The Likelihood:

$$\prod_{i=1}^n f(y_i|a, b) = \prod_{i=1}^n y_i^{a-1} (1-y_i)^{b-1} \cdot \frac{\Gamma(a+b)}{\Gamma a * \Gamma b}$$

For n observations:

$$\begin{aligned}
&= \prod_{i=1}^n e^{(a-1) \ln y_i + (b-1) \ln(1-y_i) + \ln \Gamma(a+b) - \ln \Gamma a - \ln \Gamma b} \\
&= e^{(a-1) \sum_{i=1}^n \ln y_i + (b-1) \sum_{i=1}^n \ln(1-y_i) + n(\ln \Gamma(a+b) - \ln \Gamma a - \ln \Gamma b)}
\end{aligned}$$

Now we have $B'(y_1, \dots, y_n) = \sum_{i=1}^n \ln y_i$

and we have $C'(y_1, \dots, y_n) = \sum_{i=1}^n \ln(1-y_i)$

Now the posterior becomes:

$$\begin{aligned}
f''(a, b|\{y\}, \gamma, \delta, \theta) &= f''(a, b|y_1, \dots, y_n, \gamma, \delta, \theta) \\
&\propto \prod_{i=1}^n f(y_i|a, b) \cdot f'(a, b|\gamma, \delta, \theta) \\
&\propto e^{(a-1) \sum_{i=1}^n \ln y_i + (b-1) \sum_{i=1}^n \ln(1-y_i) - n \ln B(a, b) + \gamma a + \delta b - \theta \ln B(a, b)} \\
&\propto e^{(a-1) \sum_{i=1}^n \ln y_i + (b-1) \sum_{i=1}^n \ln(1-y_i) + \gamma a + \delta b - n \theta \ln B(a, b)} \\
&\propto e^{a \sum_{i=1}^n \ln y_i - \sum_{i=1}^n \ln y_i + b \sum_{i=1}^n \ln(1-y_i) - \sum_{i=1}^n \ln(1-y_i) + \gamma a + \delta b - n \theta \ln B(a, b)} \\
&\propto e^{a(\sum_{i=1}^n \ln y_i + \gamma) + b(\sum_{i=1}^n \ln(1-y_i) + \delta) - \sum_{i=1}^n \ln(y_i) - \sum_{i=1}^n \ln(1-y_i) - n \theta \ln B(a, b)}
\end{aligned}$$

From the above equation the posterior distribution has the same prior distribution form of the prior with hyperparameters:

$$\sum_{i=1}^n \ln y_i + \gamma + \sum_{i=1}^n \ln(1-y_i) + \delta - \sum_{i=1}^n \ln y_i - \sum_{i=1}^n \ln(1-y_i)$$

References

Course documents (meeting 2, 3)

http://www.stat.ucla.edu/~dinov/courses_students.dir/Applets.dir/NormalApprox2PoissonApplet.html

Appendix

```

knitr::opts_chunk$set(echo = TRUE)
#using normal approximation
options("scipen"=100, "digits"=4)
lambdat= 300
y = 380
mu = lambdat
sigma2 = lambdat #since sigma = sqrt(lambda)

result30 = (2*pi*sigma2)^(-0.5)*exp(-(y-mu)^2/(2*sigma2))
posterior1 = result30*0.25
print(posterior1)

#using normal approximation
options("scipen"=100, "digits"=4)
lambdat= 400
y = 380
mu = lambdat
sigma2 = lambdat #since sigma = sqrt(lambda)

result40 = (2*pi*sigma2)^(-0.5)*exp(-(y-mu)^2/(2*sigma2))
posterior2 = result40*0.5
print(posterior2)

#using normal approximation
options("scipen"=100, "digits"=4)
lambdat= 500
y = 380
mu = lambdat
sigma2 = lambdat #since sigma = sqrt(lambda)

result50 = (2*pi*sigma2)^(-0.5)*exp(-(y-mu)^2/(2*sigma2))
posterior3 = result50*0.25
print(posterior3)

options(digits = 22)
posterior_sum = posterior1+posterior2+posterior3
posteriors = c(posterior1/posterior_sum, posterior2/posterior_sum, posterior3/posterior_sum)
lambda_vector = c(30,40,50)

df = data.frame(posteriors, lambda_vector)
knitr::kable(df, booktabs = TRUE)

lambda1 = 3
lambda2 = 4
lambda3 = 5
plambda1 = 0.25
plambda2 = 0.5
plambda3 = 0.25
r = 3.8
t = 1

factor30 = lambda1^r * exp(-t*lambda1) * plambda1 + lambda2^r * exp(-t*lambda2)*plambda2+lambda3^r * exp(-t*lambda3)*plambda3

```

```
problambda1 = (lambda1^r)*exp(-t*lambda1) * plambda1/factor30  
problambda1  
  
problambda2 = (lambda2^r)*exp(-t*lambda2) * plambda2/factor30  
problambda2  
  
problambda3 = (lambda3^r)*exp(-t*lambda3) * plambda3/factor30  
problambda3
```