Lab 2 – Beta-Binomial Distribution

Rebecca C. Steorts

January 2018

In class, you saw the Binomial-Beta model. We will now use this to solve a very real problem! Suppose I wish to determine whether the probability that a worker will fake an illness is truly 1%. Your task is to assist me! Tasks 1–3 will be completed in lab and tasks 3–5 should be completed in your weekly homework assignment. You should still upload task 3 even though this will be worked through in lab!

Task 1

Let's start by quickly deriving the Beta-Binomial distribution.

We assume that

$$X \mid \theta \sim \text{Binomial}(\theta)$$

,

$$\theta \sim \text{Beta}(a, b),$$

where a, b are assumed to be known parameters. What is the posterior distribution of $\theta \mid X$?

$$\begin{aligned} p(\theta \mid X) &\propto p(X \mid \theta) p(\theta) && (1) \\ &\propto \theta^x (1-\theta)^{(n-x)} \times \theta^{(a-1)} (1-\theta)^{(b-1)} && (2) \\ &\propto \theta^{x+a-1} (1-\theta)^{(n-x+b-1)}. && (3) \end{aligned}$$

This implies that

$$\theta \mid X \sim \text{Beta}(x + a, n - x + b).$$

Task 2

Simulate some data using the rbinom function of size $n = 100$ and probability equal to 1%. Remember to set.seed(123) so that you can replicate your results.

The data can be simulated as follows:

```
# set a seed
set.seed(123)
# create the observed data
obs.data <- rbinom(n = 100, size = 1, prob = 0.01)
# inspect the observed data
head(obs.data)
```

```
## [1] 0 0 0 0 0 0
```

```
tail(obs.data)
```

```
## [1] 0 0 0 0 0 0
```

```r
length(obs.data)
```

```
## [1] 100
```

Task 3

Write a function that takes as its inputs that data you simulated (or any data of the same type) and a sequence of θ values of length 1000 and produces Likelihood values based on the Binomial Likelihood. Plot your sequence and its corresponding Likelihood function.

The likelihood function is given below. Since this is a probability and is only valid over the interval from $[0, 1]$ we generate a sequence over that interval of length 1000.

You have a rough sketch of what you should do for this part of the assignment. Try this out in lab on your own.

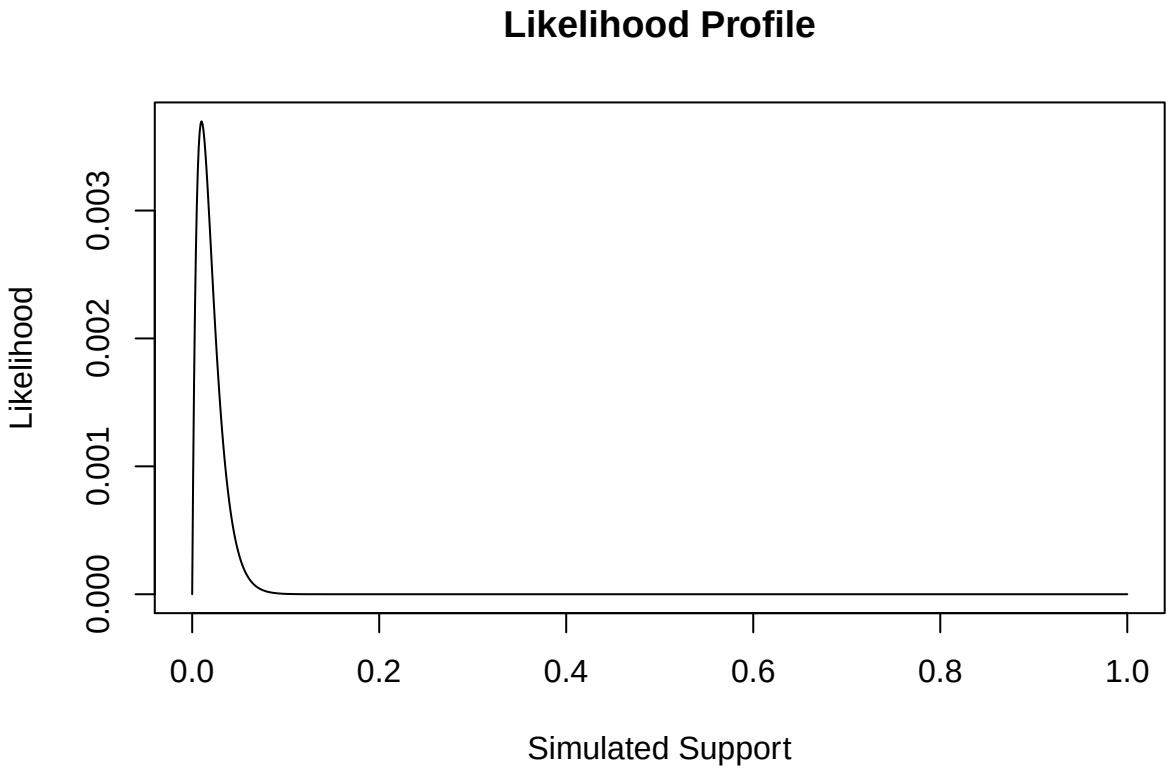
```r
### Bernoulli LH Function ###

# Input: obs.data, theta
# Output: bernoulli likelihood

myBernLH <- function(obs.data, theta){
  N <- length(obs.data)
  x <- sum(obs.data)
  LH <- (theta^x) * ((1-theta)^(N-x))
  return(LH)
}


### Plot LH for a grid of theta values ###
# Create the grid #
# Store the LH values
# Create the Plot
theta.sim <- seq(from = 0, to = 1, length.out = 1000)
sim.LH <- myBernLH(obs.data, theta = theta.sim)
plot(theta.sim, sim.LH, type ="line", main = "Likelihood Profile", xlab = "Simulated Support", ylab = "
```

```
## Warning in plot.xy(xy, type, ...): plot type 'line' will be truncated to first
## character
```



Task 4 (To be completed for homework)

Write a function that takes as its inputs prior parameters a and b for the Beta-Bernoulli model and the observed data, and produces the posterior parameters you need for the model. **Generate and print** the posterior parameters for a non-informative prior i.e. (a,b) = (1,1) and for an informative case (a,b) = (3,1)}.

```
posterior <- function(priorA, priorB, obs.data)
{
  N <- length(obs.data)
  x <- sum(obs.data)
  postA <- priorA + x
    postB <- priorB + N - x
    postparam <- list('postA' = postA,
     'postB' = postB)
    return(postparam)
}
```

```
nonInformative <- posterior(priorA = 1, priorB = 1, obs.data = obs.data)
informative <- posterior(priorA = 3, priorB = 1, obs.data = obs.data)
print(c(nonInformative, informative))
```

```
## $postA
## [1] 2
##
## $postB
## [1] 100
##
## $postA
## [1] 4
```

```
##
## $postB
## [1] 100
```

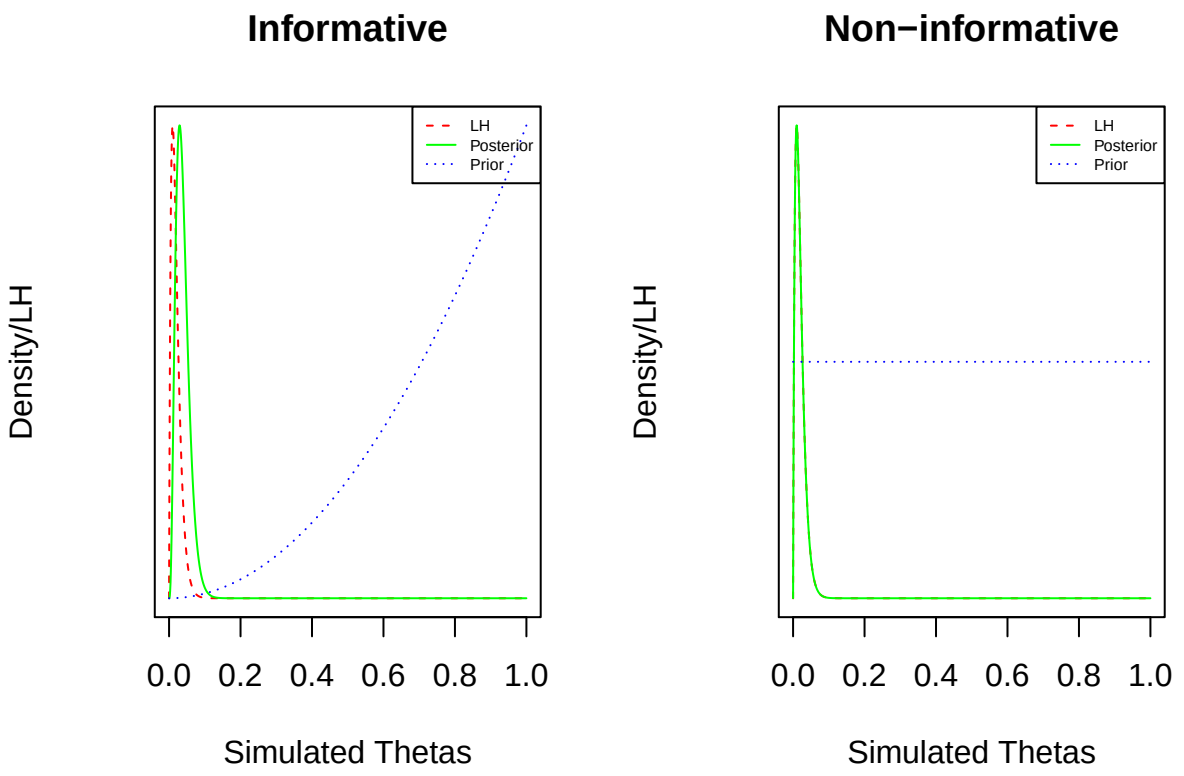
Task 5 (To be completed for homework)

Create two plots, one for the informative and one for the non-informative case to show the posterior distribution and superimpose the prior distributions on each along with the likelihood. What do you see? Remember to turn the y-axis ticks off since superimposing may make the scale non-sense.

```
nonInformativeDen <- dbeta(x = theta.sim, shape1 = nonInformative$postA,
 shape2 =nonInformative$postB)
informativeDen <- dbeta(x = theta.sim, shape1 = informative$postA,
 shape2 = informative$postB)
priorInform <- dbeta(x = theta.sim, shape1 = 3,
 shape2 = 1)
priorNonInform <- dbeta(x = theta.sim, shape1 = 1,
 shape2 = 1)

par(mfrow=c(1, 2))
plot(theta.sim, sim.LH, lty = 2, xlab = 'Simulated Thetas',
 ylab = 'Density/LH', type = 'l', yaxt = 'n', main = 'Informative',
 , col = c("red"))
par(new = TRUE)
plot(theta.sim, informativeDen, lty = 1, axes = FALSE, xlab = '', ylab = '',
 type = 'l', , col = c("green"))
par(new = TRUE)
plot(theta.sim, priorInform, lty = 3, axes = FALSE, xlab = '', ylab = '',
 type = 'l', col = c("blue"))
legend('topright', lty=c(2,1,3), legend = c('LH', 'Posterior', 'Prior'),
       col = c("red", "green", "blue"), cex = 0.5)

plot(theta.sim, sim.LH, lty = 2, xlab = 'Simulated Thetas',
 ylab = 'Density/LH', type = 'l', yaxt = 'n', main = 'Non-informative',
 col = c("red"))
par(new = TRUE)
plot(theta.sim, nonInformativeDen, lty = 1, axes = FALSE, xlab = '', ylab = '',
 type = 'l', col = c("green"))
par(new = TRUE)
plot(theta.sim, priorNonInform, lty = 3, axes = FALSE, xlab = '', ylab = '',
 type = 'l', col = c("blue"))
legend('topright', lty=c(2,1,3), legend = c('LH', 'Posterior', 'Prior'),
       col = c("red", "green", "blue"), cex=0.5)
```



These graphs both show the posterior distribution is a result of combining the Likelihood and the Prior. It is graphically seen as an average between the Likelihood and Prior. A non-informative prior is graphically seen as a flat line and seems to have less effect on the posterior than a prior which is informative, which is graphically a upwards sloping curve, and is shown to have a larger effect on moving the posterior rightwards.