

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 $\theta \mid X$?

$$p(\theta \mid X) \propto p(X \mid \theta)p(\theta) \tag{1}$$

$$\propto \theta^x (1 - \theta)^{(n-x)} \times \theta^{(a-1)} (1 - \theta)^{(b-1)} \tag{2}$$

$$\propto \theta^{x+a-1} (1 - \theta)^{(n-x+b-1)}. \tag{3}$$

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
```

```
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.

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

```
# Input: obs.data, theta
```

```
# Output: bernoulli likelihood
```

```
bernoulliLHFunction <- function(obs.data, theta) {  
  x <- sum(obs.data)  
  n <- length(obs.data)  
  like <- dbeta(theta, x+1, n-x+1)  
  return(like)  
}
```

```
### Plot LH for a grid of theta values ###
```

```
# Create the grid #
```

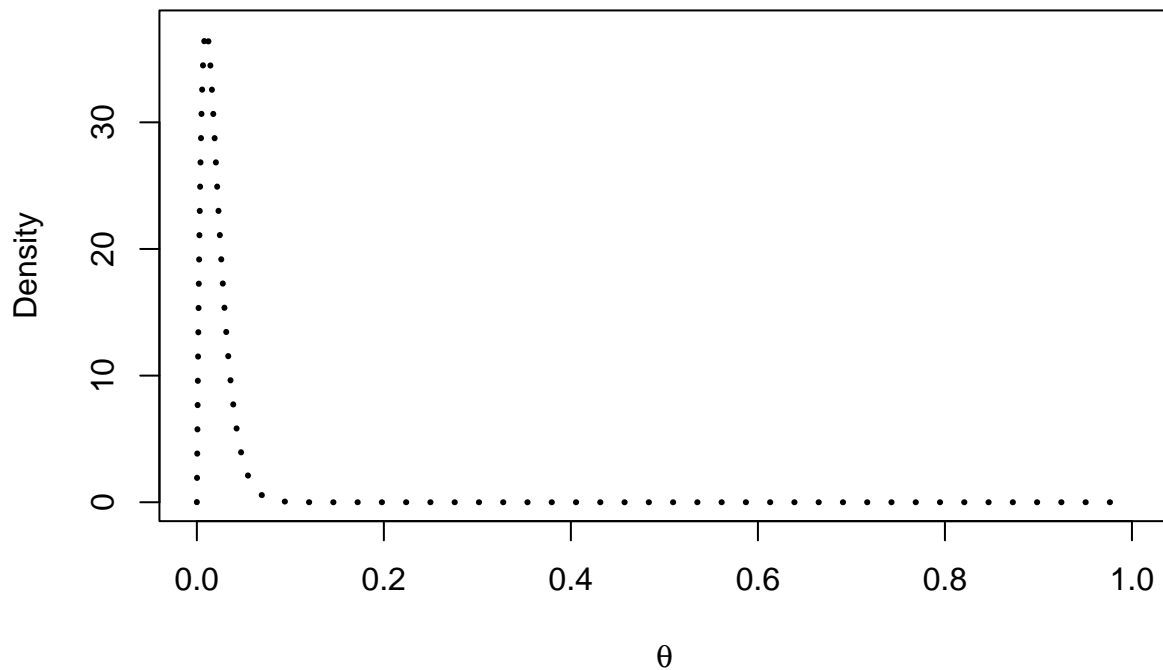
```
# Store the LH values
```

```
# Create the Plot
```

```
theta <- seq(0, 1, length = 1000)
```

```
like <- bernoulliLHFunction(obs.data, theta)
```

```
plot(theta, like, type = "l", ylab = "Density", lty = 3, lwd = 3, xlab = expression(theta))
```



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 $(a,b) = (3,1)$.

```
betabernoulliPosteriorFunction <- function(a, b, obs.data) {
  x <- sum(obs.data)
  n <- length(obs.data)
  posterior.a <- x + a
  posterior.b <- n - x + b
  return(c(posterior.a, posterior.b))
}
```

Non-Informative Prior

```
betabernoulliPosteriorFunction(1, 1, obs.data)
```

```
## [1] 2 100
```

Informative Prior

```
betabernoulliPosteriorFunction(3, 1, obs.data)
```

```
## [1] 4 100
```

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.

```
informative.posterior.a <- betabernoulliPosteriorFunction(3, 1, obs.data)[1]
informative.posterior.b <- betabernoulliPosteriorFunction(3, 1, obs.data)[2]
noninformative.posterior.a <- betabernoulliPosteriorFunction(1, 1, obs.data)[1]
noninformative.posterior.b <- betabernoulliPosteriorFunction(1, 1, obs.data)[2]
informative.posterior <- dbeta(theta, informative.posterior.a, informative.posterior.b)
noninformative.posterior <- dbeta(theta, noninformative.posterior.a, noninformative.posterior.b)
informative.prior <- dbeta(theta, 3, 1)
noninformative.prior <- dbeta(theta, 1, 1)
plot(theta, like, type = "l", ylab = "Density", lty = 3, lwd = 3, xlab = expression(theta))
points(theta, informative.prior)
```

