Lab 2 Solutions - STA 360/601

Abbas Zaidi and Rebecca C. Steorts

January, 2016

1. Problem 1

The data can be simulated as follows:

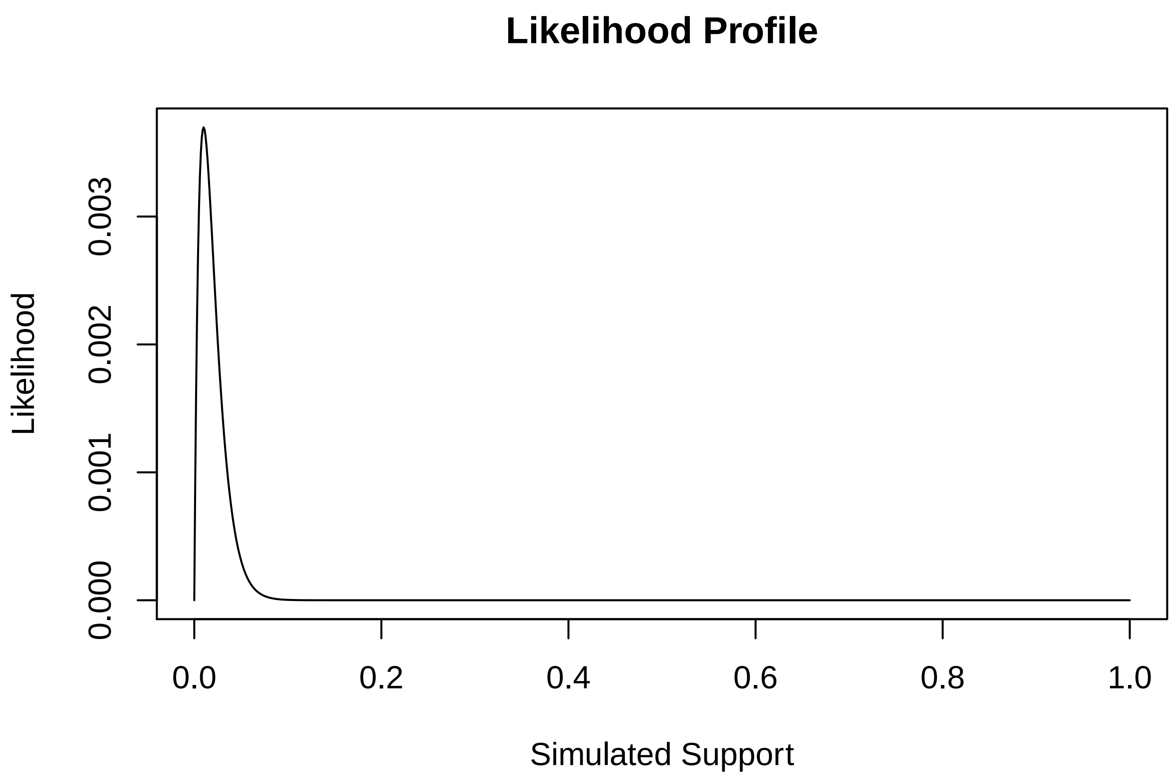
```
set.seed(123)
obs.data <- rbinom(n = 100, size = 1, prob = 0.01)
```

2. Problem 2

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.

```
### Bernoulli LH Function ###
# Input - the data, theta grid #
# Produces likelihood values #
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 #
theta.sim <- seq(from = 0, to = 1, length.out = 1000)
# Store the LH Values #
sim.LH <- myBernLH(obs.data = obs.data, theta = theta.sim)
# Create the Plot #
plot(theta.sim, sim.LH, type = 'l',
 main = 'Likelihood Profile', xlab = 'Simulated Support',
  ylab = 'Likelihood')
```



3. Problem 3

The function used to generate the paraters is given as follows. The parameters themselves are printed out and stores as well for later use.

```
### Function to determine posterior parameters based on ###
### observed data and prior assumptions ###
# Inputs - Prior Paramters, observed data #
myPosteriorParam <- function(pri.a, pri.b, obs.data){
    N <- length(obs.data)
    x <- sum(obs.data)
    post.a <- pri.a + x
    post.b <- pri.b + N - x
    post.param <- list('post.a' = post.a,
     'post.b' = post.b)
    return(post.param)
}

# Find posterior parameters for two different priors #
# a = 1, b = 1 #
non.inform <- myPosteriorParam(pri.a = 1, pri.b = 1, obs.data = obs.data)
# a = 3, b = 1 #
inform <- myPosteriorParam(pri.a = 3, pri.b = 1, obs.data = obs.data)
print(non.inform)
```

```
## $post.a
## [1] 2
##
```

```
## $post.b
## [1] 100
```

```
print(inform)
```

```
## $post.a
## [1] 4
##
## $post.b
## [1] 100
```

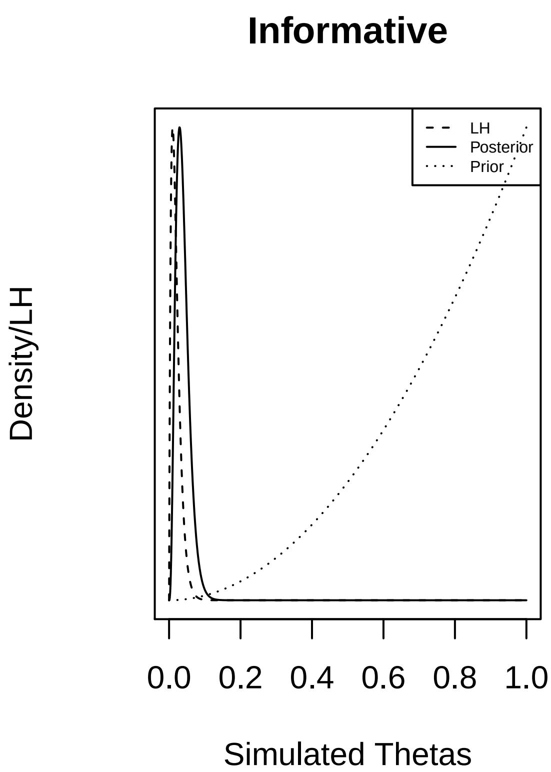
4. Problem 4

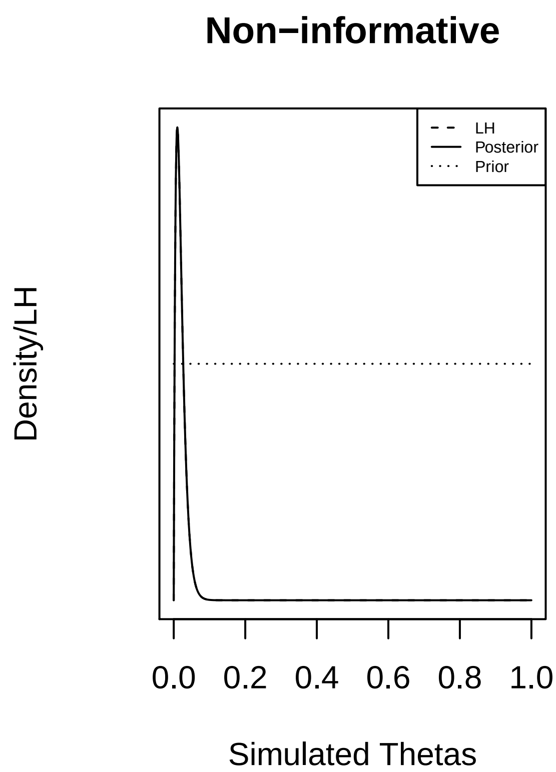
The desired plots are given below, along with the code used to generate them.

```
### Create a plot of LH, Pri, Posterior using the simulated seq ###
non.inform.den <- dbeta(x = theta.sim, shape1 = non.inform$post.a,
 shape2 = non.inform$post.b)
inform.den <- dbeta(x = theta.sim, shape1 = inform$post.a,
 shape2 = inform$post.b)
pri.inform <- dbeta(x = theta.sim, shape1 = 3,
 shape2 = 1)
pri.non.inform <- 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')
par(new = TRUE)
plot(theta.sim, inform.den, lty = 1, axes = FALSE, xlab = '', ylab = '',
 type = 'l')
par(new = TRUE)
plot(theta.sim, pri.inform, lty = 3, axes = FALSE, xlab = '', ylab = '',
 type = 'l')
legend('topright', lty=c(2,1,3), legend = c('LH', 'Posterior', 'Prior'),
cex = 0.5)

plot(theta.sim, sim.LH, lty = 2, xlab = 'Simulated Thetas',
 ylab = 'Density/LH', type = 'l', yaxt = 'n', main = 'Non-informative')
par(new = TRUE)
plot(theta.sim, non.inform.den, lty = 1, axes = FALSE, xlab = '', ylab = '',
 type = 'l')
par(new = TRUE)
plot(theta.sim, pri.non.inform, lty = 3, axes = FALSE, xlab = '', ylab = '',
 type = 'l')
legend('topright', lty=c(2,1,3), legend = c('LH', 'Posterior', 'Prior'),
cex=0.5)
```





In the plots given above, the first think to note is Shrinkage. The posterior distribution is averaged between the prior and the likelihood. As a result, when we used the flat prior, the influence of the likelihood is much, much greater than in the case where we use an informative prior.

```
require(xtable)
## Create confidence/ credible intervals - informative ##
# Credible Interval #
my.sim <- rbeta(n = 1000, shape1 = inform$post.a,
 shape2 = inform$post.b)
my.credI <- quantile(my.sim, prob = c(0.025, 0.975))
```