STAT3023 Statiscal Inference

Lab Week 11

Tutor: Wen Dai

SID: 470408326

School of Mathematics and Statistics

The University of Sydney

Semester 2, 2021

1 (a) We write a function that computes the MLE based interval

```
mle.int = function(x, C) {
    # x is a random sample:
    thetaML = 1/mean(x)
    # C is half with width of the intervals
    d1 = max(thetaML, C)
    c(d1 - C, d1 + C)
}
```

(b) We test it by generating a random sample:

```
n = 4
C = 1.5
th0 = 2
x = rexp(n, rate = th0)
mle_interval = mle.int(x, C)
mle_interval
```

```
[1] 1.742083 4.742083
```

2 (a) We now write a function called `bayes.int()` which computes the Bayes interval based on the flat prior.

```
bayes.int = function(x, C) {
    d2 = C * (exp(2 * mean(x) * C) + 1)/(exp(2 * mean(x) * C) - 1)
    c(d2 - C, d2 + C)
}
```

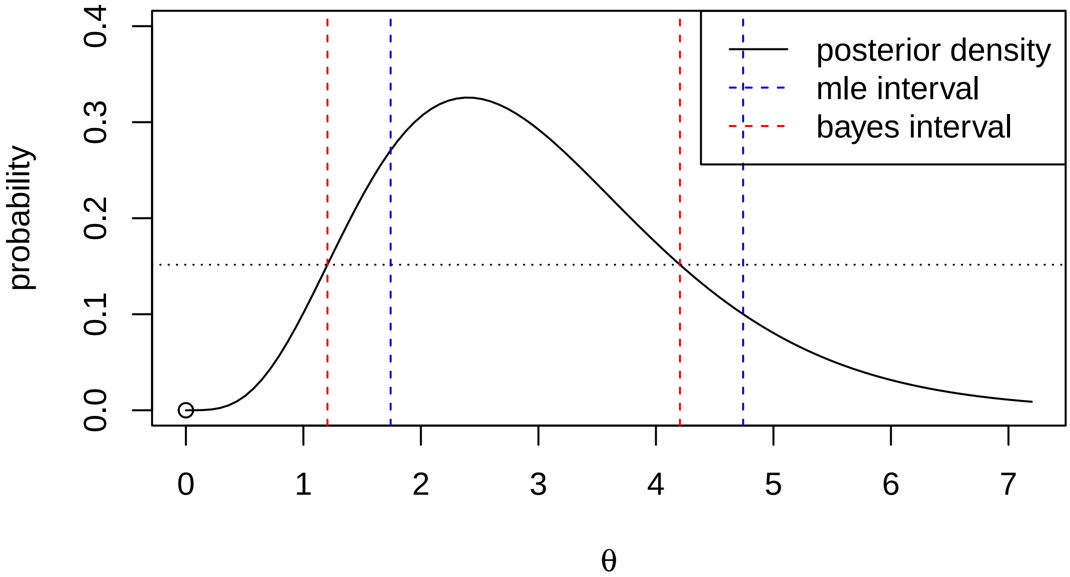
(b) Again we test it by testing the same sample

```
n = 4
C = 1.5
th0 = 2
x = rexp(n, rate = th0)
bayes_interval = bayes.int(x, C)
bayes_interval
```

```
[1] 1.204336 4.204336
```

(c) We visualize the interval by plotting the posterior curve and interval on the same plot.

```
m = mean(x)
# label the Origin
plot(0, 0, xlim = c(0, 3/m), ylim = c(0, 0.4), xlab = expression(theta), ylab
    = "probability")
curve(dgamma(x, shape = n + 1, rate = n * m), from = 0, to = 3/m, add = TRUE)
abline(v = mle_interval, col = "blue", lty = 2)
abline(v = bayes_interval, col = "red", lty = 2)
abline(h = dgamma(bayes_interval[1], shape = n + 1, rate = n * m), lty = 3)
legend("topright", legend = c("posterior density", "mle interval", "bayes
    interval"), col = c("black",
    "blue", "red"), lty = c(1, 2, 2))
```



3 We plot the non-coverage probabilities of θ values for the MLE based interval vs the Bayes interval.

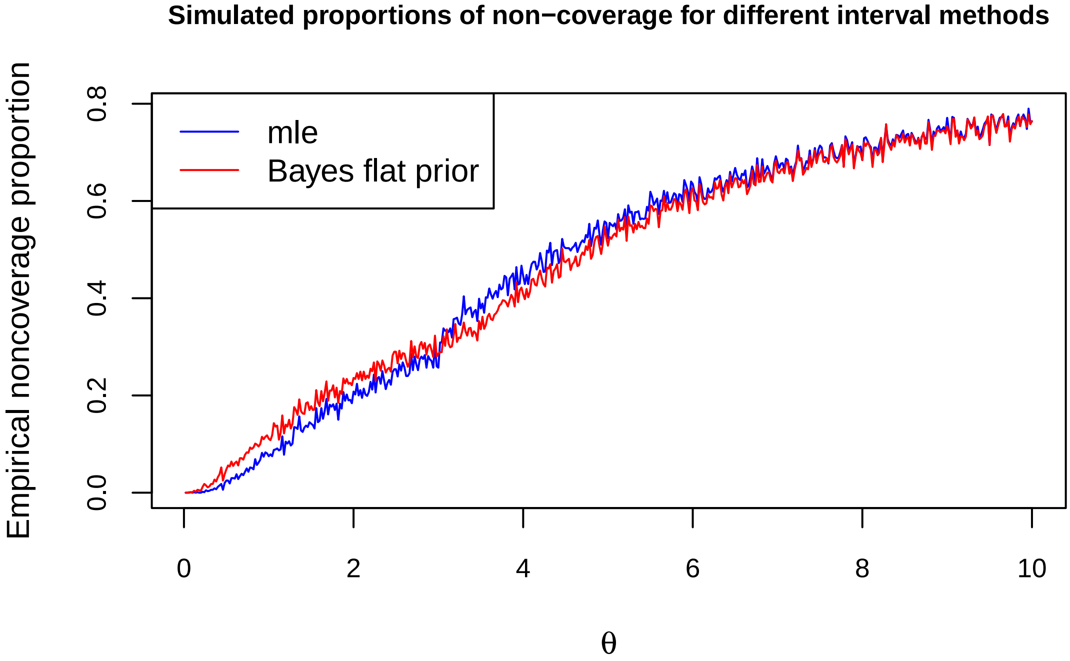
```
th = (1:500)/50
L = length(th)
B = 1000
noncoverage.mle = noncoverage.bayes = 0
c = 1.5

# we fix the theta value
for (i in 1:L) {
    mle.mat = matrix(0, B, 2)
    bayes.mat = matrix(0, B, 2)

    # we fix the row in our matrix.
    for (j in 1:B) {
        x = rexp(4, rate = th[i])  # draw 4 random numbers from the exp distn.
        mle.mat[j, ] = mle.int(x, c)  # constructing intervals
        bayes.mat[j, ] = bayes.int(x, c)
    }

    # count the number of intervals not containing theta
    noncoverage.mle[i] = sum(th[i] < mle.mat[, 1]) + sum(th[i] > mle.mat[, 2])
    noncoverage.bayes[i] = sum(th[i] < bayes.mat[, 1]) + sum(th[i] > bayes.mat[,
        2])
}
plot(th, noncoverage.mle/B, type = "l", col = "blue", main = "Simulated
    proportions of non-coverage for different interval methods",
    xlab = expression(theta), ylab = "Empirical noncoverage proportion", cex.main
        = 0.85, cex.axis = 0.8)
```

```
lines(th, noncoverage.bayes/B, type = "l", col = "red")
legend("topleft", legend = c("mle", "Bayes flat prior"), col = c("blue", "red"),
    lty = c(1, 1))
```



4 We compute the risk for the MLE based interval for each value of θ from our vector `th` and we save it in a vector called `m.risk`.

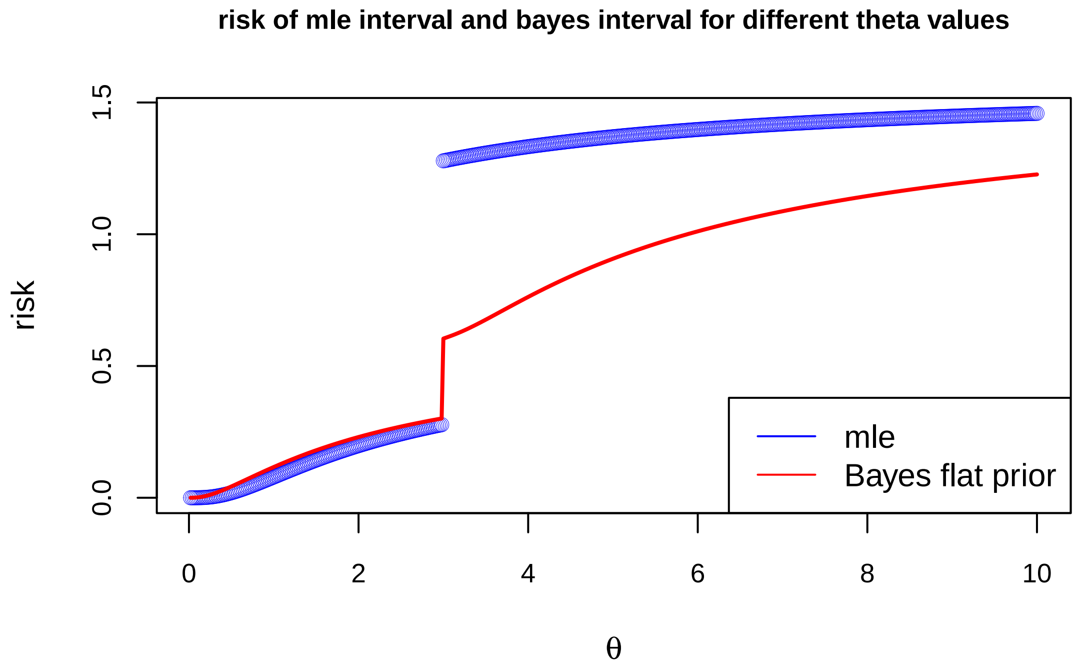
```
risk.overest = pgamma(q = 1/(th + c), shape = n, rate = n * th)  # this applies to
     all values in th
big = (th ≥ (2 * C))  # find the theta values ≥ 2C
risk.underest = 0 * risk.overest  # Start with a vector of zeroes
risk.underest[big] = pgamma(q = 1/(th[big] + c), shape = n, rate = n * th[big]) +
    pgamma(q = 1/(th[big] +
     c), shape = n, rate = n * th[big], lower.tail = FALSE)
m.risk = risk.overest + risk.underest
```

5 We compute the risk for the Bayes interval for each values of θ from our vector `th` and we save it in a vector called `m.risk`.

```
risk.overest = pgamma(q = 1/(2 * c) * log(1 + 2 * c/th), shape = n, rate = n * th)
risk.underest = 0 * risk.overest  # Start with a vector of zeroes
big = th ≥ 2 * c
risk.underest[big] = pgamma(q = 1/(2 * c) * log(1 + 2 * c/th[big]), shape = n,
    rate = n * th[big]) +
     pgamma(q = 1/(2 * c) * log(th[big]/(th[big] - 2 * c)), shape = n, rate = n *
         th[big], lower.tail = FALSE)
b.risk = risk.overest + risk.underest
```

We now plot our curves:

```
plot(th, m.risk, col = "blue", lwd = 0.2, main = "risk of mle interval and bayes
    interval for different theta values",
     cex.main = 0.85, cex.axis = 0.8, xlab = expression(theta), ylab = "risk")
lines(th, b.risk, col = "red", lwd = 2)
legend("bottomright", legend = c("mle", "Bayes flat prior"), col = c("blue", "red"
    ), lty = c(1, 1))
```



The values of θ for which the mle interval does better are the θ values **less than** 3. The interval in which the bayes interval does better are the θ values **greater than** 3.

The two intervals have somewhat similar performances for $\theta < 3$ but the bayes interval is noticably better for $\theta > 3$.

Near the value of $\theta = 3$ there is a discontinuity for both the mle and the bayes intervals. This is because of the fact that the intervals are different depending on if you're in the region of $0 < \theta < 2C$ or the region $\theta > 2C$. In our example, $C = 1.5$ and hence the intervals are drastically different on the boundary of $2C = 2(1.5) = 3$