bayesian_stats_ch4_montecarlo_approximation

inoue jin

2022-12-03

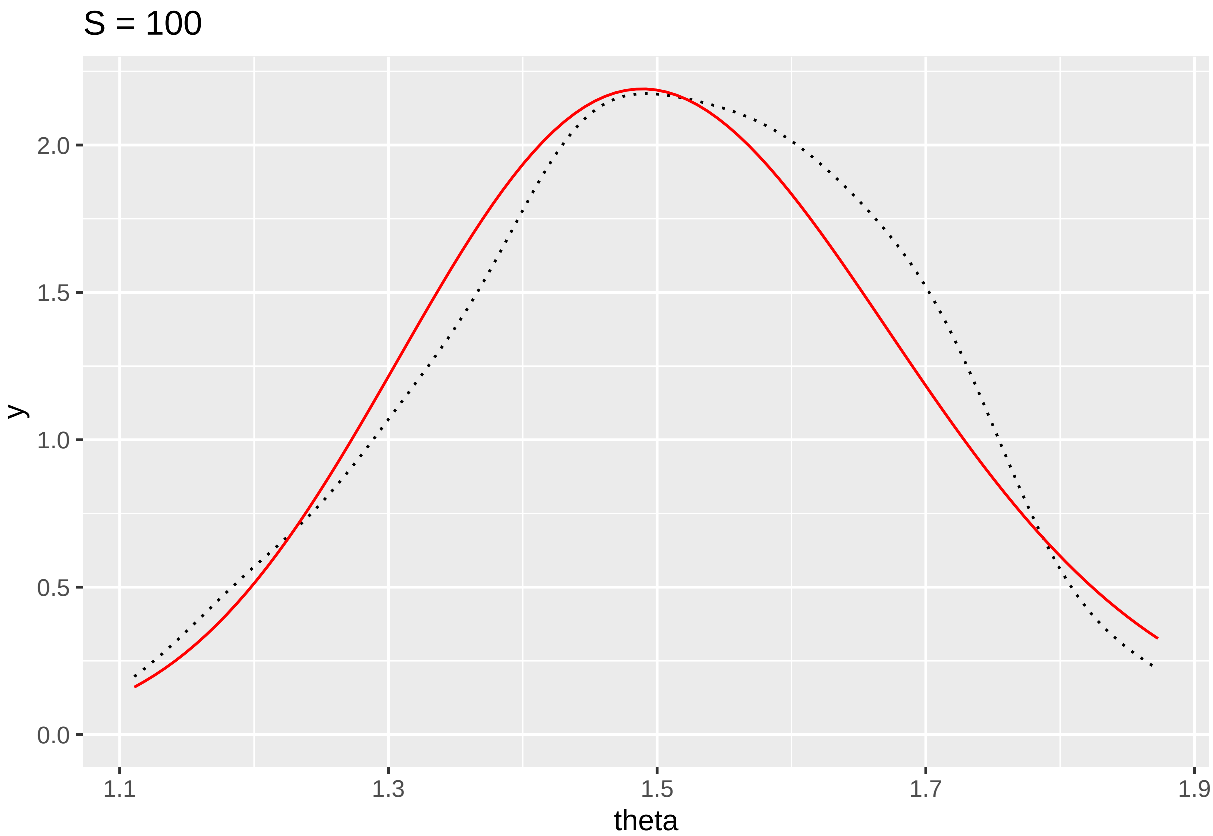
ガンマ分布の真の密度関数と経験分布によるモンテカルロ近似

```
a = 68
b = 45

s100 <- tibble(theta = rgamma(100, a, b))
s1000 <- tibble(theta = rgamma(1000, a, b))
s10000 <- tibble(theta = rgamma(10000, a, b))

s100
```

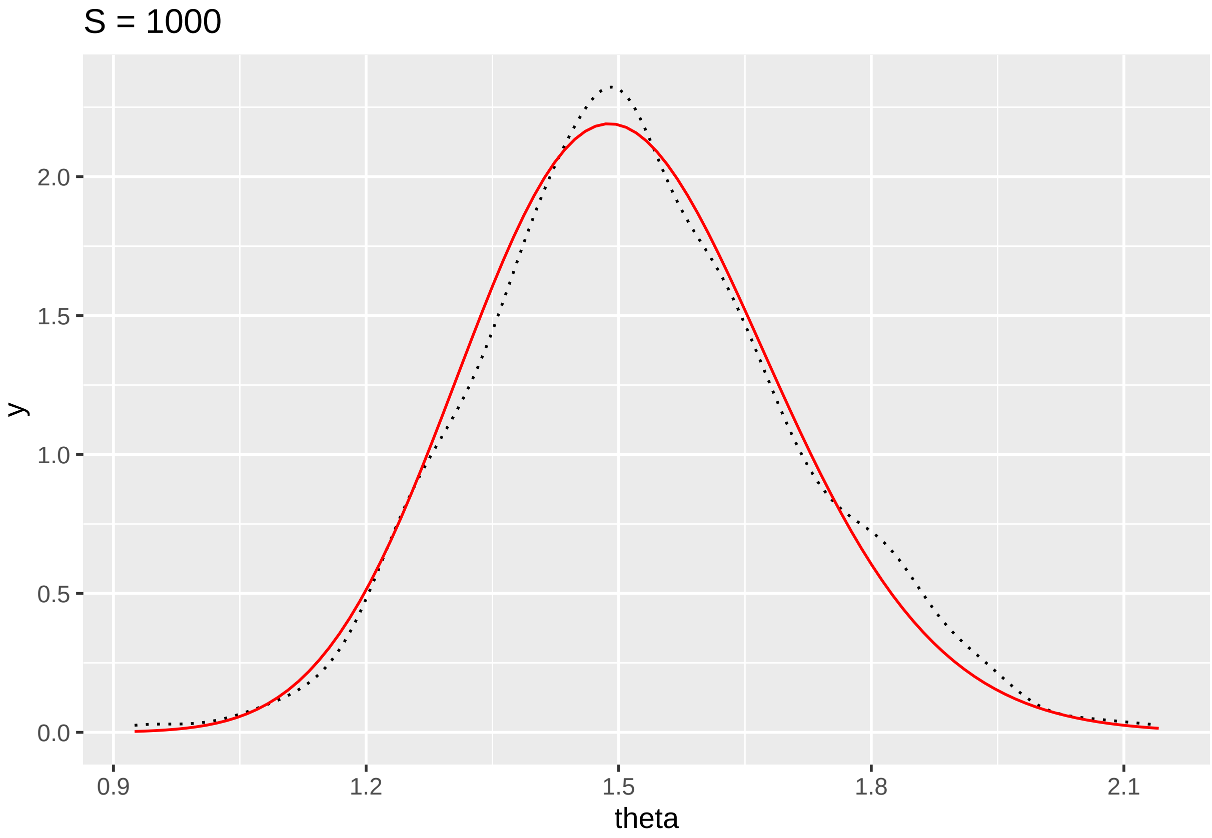
```
## # A tibble: 100 x 1
##    theta
##    <dbl>
##  1  1.28
##  2  1.65
##  3  1.55
##  4  1.47
##  5  1.60
##  6  1.62
##  7  1.49
##  8  1.35
##  9  1.43
## 10  1.38
## # ... with 90 more rows
```

```
# curve(dgamma(x, a, b), -2, 4)

gg <- ggplot(data.frame(x = c(-2, 4)), aes(x = x)) +
  xlim(c(-2, 4)) +
```

```
  ylim(c(0, 3)) +
  stat_function(fun = dgamma, args = list(a, b))

gg100 <- s100 %>%
  ggplot(aes(x = theta)) +
  geom_density(linetype = "dotted")+
  stat_function(fun = dgamma, args = list(a, b), color = "red") +
  labs(title = "S = 100") +
  xlab("theta")
gg100
```




```
gg1000 <- s1000 %>%
  ggplot(aes(x= theta)) +
  geom_density(linetype = "dotted") +
  stat_function(fun = dgamma, args = list(a, b), color = "red") +
  labs(title = "S = 1000")+
  xlab("theta")

gg1000
```



数値的な評価

まず、モンテカルロ法によって得られる近似値を、事後要約統計量の解析的に得られる解と比較する。

- モデル $Y_1, \dots, Y_n \mid \theta \sim \text{i.i.d.} Poisson(\theta)$
- θ の事前分布は $gamma(a, b)$
- この時、$Y_1 = y_1, \dots, Y_n = y_n$ を観測した時の事後分布は、$gamma(a + \Sigma y_i, b + n)$ となる
- 事後平均は $(a + \Sigma y_i)/(b + n) = 1.51$ となる

```
a <- 2
b <- 1
sy <- 66
n <- 44

theta_mc10 <- rgamma(10, a + sy, b+n)
theta_mc100 <- rgamma(100, a+sy, b+n)
theta_mc1000 <- rgamma(1000, a+sy, b+n)

mean(theta_mc10)
```

```
## [1] 1.461037
```

```r
mean(theta_mc100)
```

```
## [1] 1.520959
```

```r
mean(theta_mc1000)
```

```
## [1] 1.515606
```

```r
## theta < 1.75の確率を計算
# Pro(theta < 1.75)
#

pgamma(q = 1.75, a+sy, b+n)
```

```
## [1] 0.8998286
```

```r
mean(theta_mc10 < 1.75)# \sum 1{theta_mc10 < 1.75}*(1/n)
```

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

```r
## 事後分布による95%信用領域
qgamma(c(0.025, 0.975), a+sy, b+n)
```

```
## [1] 1.173437 1.890836
```

```r
# monte carlo標本から95%信用領域を求める

quantile(theta_mc10, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 1.280973 1.689170
```

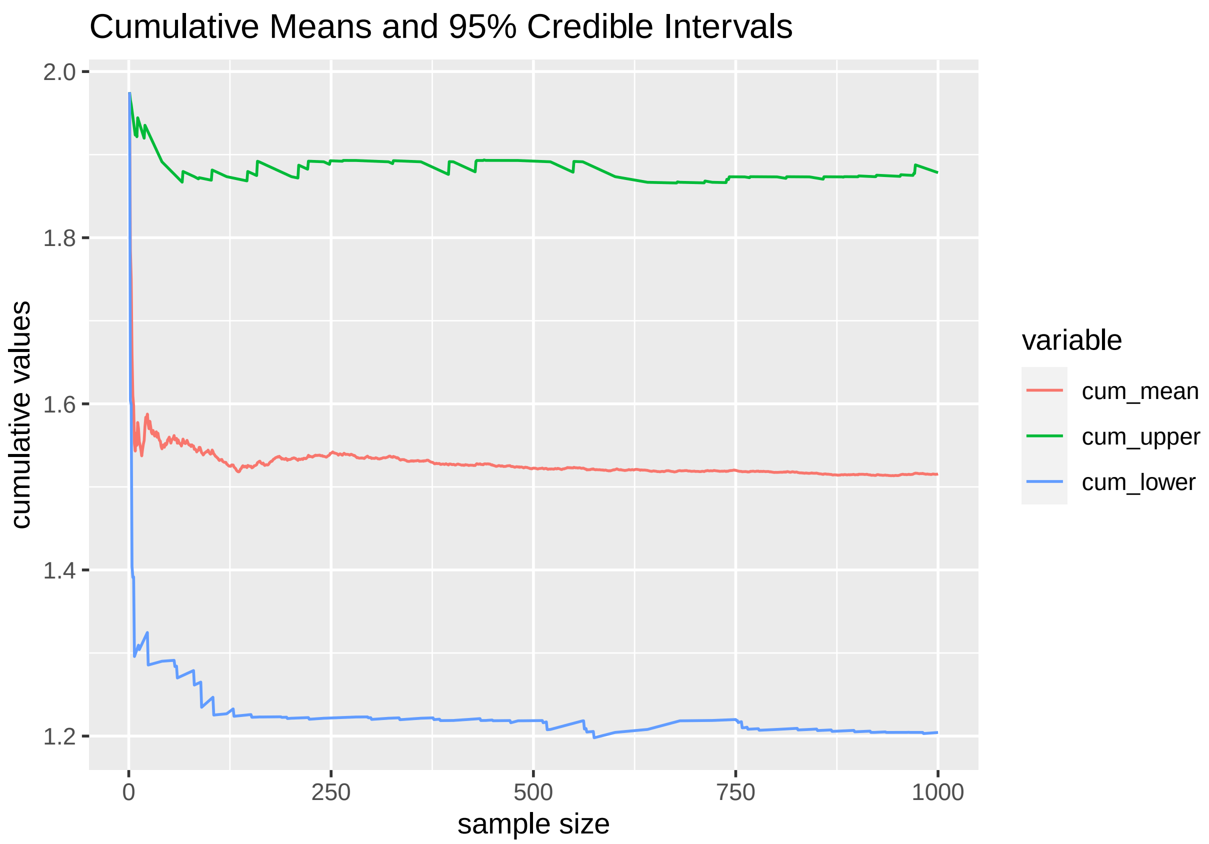
```r
# quantile(theta_mc100, 0.975) %>% as.double()
```

```r
# listで、ここの累積リストに対してquantileを適用して累積を計算する

# theta_mc1000
# theta_mc1000[1:1000]

theta_list <- list()# 1000 * 1000のリスト作成

# O(N^2)の計算時間
for(i in 1:1000){
  theta_list[[i]] <- theta_mc1000[1:i]
}
```

```r
# theta_list %>% glimpse()
# theta_list[[1]]
# map(.x = theta_list, ~as.double(quantile(.x, 0.975)))

# 累積平均、累積 97.5%分位点、累積 2.5%分位点をまとめて作成

cum_df <- tibble(n = 1:1000,
                 cum_mean = cumsum(theta_mc1000)/(1:1000),
                 cum_upper = unlist(map(.x = theta_list, ~as.double(quantile(.x, 0.975)))),
                 cum_lower = unlist(map(.x = theta_list, ~as.double(quantile(.x, 0.025)))))

# cum_df %>% summary()

# {reshape2}で long 型にする

data_long <- reshape2::melt(cum_df, id = "n")

data_long %>% glimpse()
```

```
## Rows: 3,000
## Columns: 3
## $ n        <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18~
## $ variable <fct> cum_mean, cum_mean, cum_mean, cum_mean, cum_mean, cum_mean, c~
## $ value    <dbl> 1.975160, 1.784802, 1.749877, 1.659630, 1.609776, 1.597650, 1~
```

```r
gg_cumsum <- data_long %>%
  ggplot(aes(x = n, y = value, color = variable))+
  geom_line() +
  xlab("sample size")+
  ylab("cumulative values")+
  labs(title = "Cumulative Means and 95% Credible Intervals")

gg_cumsum
```


Cumulative Means and 95% Credible Intervals
2.0
1.8
1.6
1.4
1.2
cumulative values
0
250
500
750
1000
sample size
variable
cum_mean
cum_upper
cum_lower