BLB Code

Xingche Guo

4/6/2017

Bootstrap

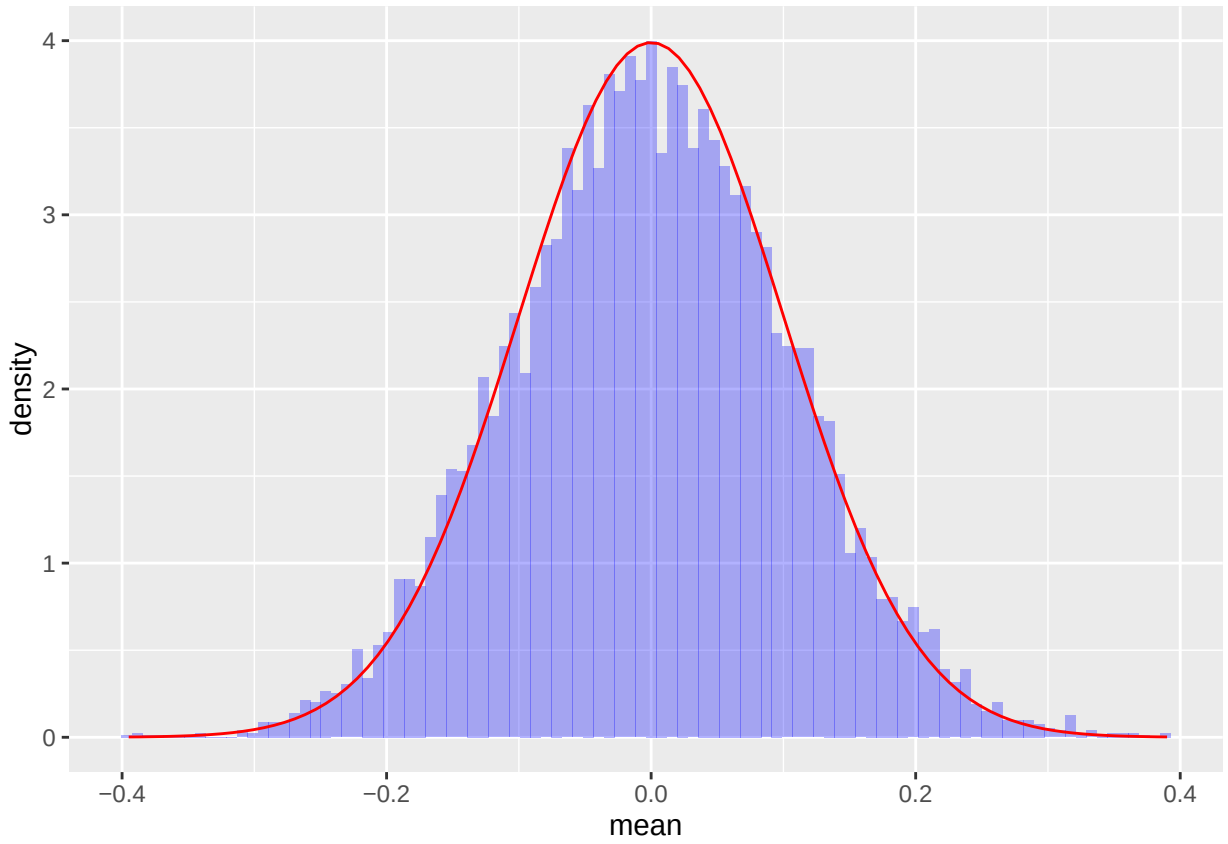
Nonparametric bootstrap

```
library(boot)
x <- rnorm(100)
samplemean <- function(x, d) {
  return(mean(x[d]))
}
b <- boot(x, samplemean, R=10000)
b
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = x, statistic = samplemean, R = 10000)
##
##
## Bootstrap Statistics :
##       original        bias    std. error
## t1* -0.1521754 -2.030644e-05    0.105298
```

```
ci <- boot.ci(b, conf = 0.95, type = c("norm", "basic", "perc"))
ci
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = b, conf = 0.95, type = c("norm", "basic",
##     "perc"))
##
## Intervals :
## Level      Normal              Basic              Percentile
## 95%   (-0.3585,  0.0542 )   (-0.3614,  0.0511 )   (-0.3554,  0.0571 )
## Calculations and Intervals on Original Scale
```



Compare Nonparametric bootstrap & Parametric bootstrap

```
head(cd4)
```

```
##   baseline oneyear
## 1     2.12    2.47
## 2     4.35    4.61
## 3     3.39    5.26
## 4     2.51    3.02
## 5     4.04    6.36
## 6     5.10    5.93
```

```
samplecor <- function(x,d){
  x1 <- x[d,1]
  x2 <- x[d,2]
  return(cor(x1,x2))
}
cd4.boot.nop <- boot(data = cd4, statistic = samplecor, R = 1000)
boot.ci(cd4.boot.nop,  type = c("norm", "basic", "perc"),
        conf = 0.9, h = atanh, hinv = tanh)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = cd4.boot.nop, conf = 0.9, type = c("norm",
```

```
##     "basic", "perc"), h = atanh, hinv = tanh)
##
## Intervals :
## Level      Normal              Basic              Percentile
## 90%   ( 0.5039,  0.8384 )   ( 0.5053,  0.8359 )   ( 0.5520,  0.8544 )
## Calculations on Transformed Scale;  Intervals on Original Scale
```

```
cd4.mle <- list(m = colMeans(cd4), v = var(cd4))
cd4.rg <- function(data, mle) MASS::mvrnorm(nrow(data), mle$m, mle$v)
cd4.boot <- boot(cd4, corr, R = 1000, sim = "parametric",
                 ran.gen = cd4.rg, mle = cd4.mle)
boot.ci(cd4.boot,  type = c("norm", "basic", "perc"),
        conf = 0.9, h = atanh, hinv = tanh)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = cd4.boot, conf = 0.9, type = c("norm", "basic",
##     "perc"), h = atanh, hinv = tanh)
##
## Intervals :
## Level      Normal              Basic              Percentile
## 90%   ( 0.4660,  0.8572 )   ( 0.4575,  0.8589 )   ( 0.4925,  0.8703 )
## Calculations on Transformed Scale;  Intervals on Original Scale
```

Compare bootstrap with and without parallel computing

```
para<-function(...){
  library(doParallel)
  cl <- makeCluster( detectCores() - 1  )
  registerDoParallel(cl)
  cd4.mle <- list(m = colMeans(boot::cd4), v = var(boot::cd4))
  cd4.rg <- function(data, mle) MASS::mvrnorm(nrow(data), mle$m, mle$v)

  cd4.boot <- foreach(i=1:500, .combine = c) %dopar% {
  boot::boot(boot::cd4, boot::corr, R = 200, sim = "parametric",
         ran.gen = cd4.rg, mle = cd4.mle)
  }
  stopCluster(cl)
  boot::boot.ci(cd4.boot, type = c("norm", "basic", "perc"),conf = 0.9, h = atanh, hinv = tanh)
}
```

```
set.seed(580580)
system.time(para())
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
##    user  system elapsed
##   0.621   0.075   4.087
```

```
system.time(no_para())
```

```
##    user  system elapsed
##  10.531   0.148  10.681
```

Compare BLB & bootstrap

```
library(datadr)
head(adult)
```

```
##   age        workclass fnlwgt education educationnum           marital
## 1  39        State-gov  77516 Bachelors           13     Never-married
## 2  50 Self-emp-not-inc  83311 Bachelors           13 Married-civ-spouse
## 3  38          Private 215646   HS-grad            9          Divorced
## 4  53          Private 234721      11th            7 Married-civ-spouse
## 5  28          Private 338409 Bachelors           13 Married-civ-spouse
## 6  37          Private 284582   Masters           14 Married-civ-spouse
##          occupation  relationship  race    sex capgain caploss
## 1      Adm-clerical Not-in-family White   Male    2174       0
## 2   Exec-managerial       Husband White   Male       0       0
## 3 Handlers-cleaners Not-in-family White   Male       0       0
## 4 Handlers-cleaners       Husband Black   Male       0       0
## 5    Prof-specialty          Wife Black Female       0       0
## 6   Exec-managerial          Wife White Female       0       0
##   hoursperweek nativecountry income incomebin
## 1           40 United-States  <=50K         0
## 2           13 United-States  <=50K         0
## 3           40 United-States  <=50K         0
## 4           40 United-States  <=50K         0
## 5           40          Cuba  <=50K         0
## 6           40 United-States  <=50K         0
```

```
###BLB --- drBLB
rrAdult <- divide(adult, by = rrDiv(1000), update = TRUE)
```

```
## * Input data is not 'ddf' - attempting to cast it as such
```

```
## * Verifying parameters...
```

```
## * Applying division...
```

```
## * Running map/reduce to get missing attributes...
```

```
BLB <- function(x) {
  drBLB(x,
        statistic = function(x, weights)
          coef(glm(incomebin ~ educationnum,
                   data = x, weights = weights, family = binomial()))[2],
        metric = function(x)
          quantile(x, c(0.05, 0.95)),
        R = 100,
        n = nrow(rrAdult)
  )
}
adultBlb <- addTransform(rrAdult, BLB)
```

```
## *** finding global variables used in 'fn'...
##
##   found: rrAdult
##   package dependencies: datadr, stats
## *** testing 'fn' on a subset...
##  ok
```

```
coefs <- recombine(adultBlb, combMean)
```

```
## * Applying recombination...
```

```
coefs
```

```
## [1] 0.3557908 0.3759363
```

```
### compared with bootstrap
library(boot)
coef_adult <- function(x,d){
  coef(glm(incomebin ~ educationnum,
           data = x[d,], family = binomial()))[2]
}

BOOT <- boot(adult, coef_adult, 100)
CI <- boot.ci(BOOT, conf = 0.90, type = "basic")
CI
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 100 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = BOOT, conf = 0.9, type = "basic")
##
## Intervals :
## Level      Basic
## 90%   ( 0.3528,  0.3745 )
## Calculations and Intervals on Original Scale
## Some basic intervals may be unstable
```