# Bag of Little Bootstrap

Wenting Zhao Jingru Mu
Yueying Wang Xingche Guo

Iowa State University

*Dept. of Statistics*

April 19, 2017

# Introduction

The *empirical bootstrap* is a statistical technique popularized by Bradley Efron in 1979.

In statistics, bootstrapping is any test or metric that relies on random sampling with replacement. It allows assigning measures of accuracy to sample estimates and estimation of the sampling distribution of almost any statistic using random sampling methods.

# Problem Setting and Notation

- Sample observation $X_1, X_2, \cdots, X_n$ are drawn i.i.d from some unknown underlying population $P \in \mathcal{P}$
- Empirical distribution: $\mathbb{P}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$
- $\theta \in \Theta$ is some (unknown) population value associated with $P$, we compute its estimator and denote by $\hat{\theta}_n$
- $\hat{\theta}_n \sim Q_n(P)$, which is its true underlying distribution.
- Estimator quality assessment $\xi(Q_n(P))$, which depend directly on $P$ and $Q_n(P)$.
    - For example, quantile, confidence interval/region, standard error, and bias.

# Problem Setting and Notation(Cont'd)

- Goal:
  Estimate $\xi(Q_n(P))$ (the estimator quality assessment of $\hat{\theta}_n$) based only on:
  - The observed data $X_1, X_2, \cdots, X_n$
  - Knowledge of the form of the estimator $\hat{\theta}_n$.
- Difficulty:
  Distribution $P$ and $Q_n(P)$ unknown.

# Basic Idea - Bootstrap

- Repeatedly resample $n$ points i.i.d. from $\mathbb{P}_n$.
- Compute the estimate on each resample: $\hat{\theta}_n^{(1)}, \hat{\theta}_n^{(2)}, \ldots$.
- Form the empirical distribution $\mathbb{Q}_n^*$ of $\hat{\theta}_n^{(k)}, k = 1, 2, \ldots$.
- Approximate $\xi(Q_n(P)) \approx \xi(\mathbb{Q}_n^*)$

# Bootstrap in R

- The *boot* package provides extensive facilities for bootstrapping and related resampling methods
- A simple example:

```
bootobject <- boot(data= , statistic= , R=, ...)
```

The bootstrap porvides a simple and powerful means of assesing the quality of estimators.

# Basic Idea - Bootstrap

Fact:

- Need repeated computation of the estimate on resamples having size comparable to $n$, which can be really large.
- Each bootstrap resample contains approximately $0.632n$ distinct points(Efron and Tibshirani, 1993) - higher cost of computing and storage.

However, in setting involving large datasets– which are increasingly prevalent the computation of bootstrap-based quantities can be prohibitively demanding computationally.

**Advantages:**

- Simplicity
- control and check the stability of the results.

**Disadvantages:**

- It doesn't provide general finite-sample guarantees.
- May conceal the fact that important assumptions are being made when undertaking the bootstrap analysis where these would be more formally stated in other approaches.

# How to improve it?

In recent decades, people have developed several methods that could perform better than traditional bootstrap.

- subsampling
- m out of n bootstrap
- Parallel computation
- Bag of little Bootstraps (BLB)
- BLFRB (A development of BLB)

Subsampling and the m out of n bootstraps often require use of more prior information. As an alternative, BLB is a procedure which incorporates features of both the bootstrap and subsampling to yield a robust, computationally efficient means of assessing the quality of estimators.

# Subsampling & m out of N

- Subsampling
  Without replacement, uses a smaller resample size.
- M Out of N Bootstrap
  Resample size is m, (bootstrap sample size is n)
  where $m \to \infty$, $\frac{m}{n} \to 0$. (The choice of m is an important matter.)

# Basic Idea - $m$ out of $n$ Bootstrap/Subsampling

For $m < n$,

- Repeatedly resample $m$ points i.i.d. from $\mathbb{P}_n$ (subsample $m$ points without replacement from $X_1, \cdots, X_n$).
- Compute the estimate on each resample(subsample): $\hat{\theta}_m^{(1)}, \hat{\theta}_m^{(2)}, \ldots$.
- Form the empirical distribution $\mathbb{Q}_m^*$ of $\hat{\theta}_m^{(k)}, k = 1, 2, \ldots$.
- Approximate $\xi(Q_m(P)) \approx \xi(\mathbb{Q}_m^*)$
- Apply an analytical correction to in turn approximate $\xi(Q_n(P))$.

# Pros and Cons

- Advantage
  - Repeat computation under consideration on subsamples that can be significantly smaller than the original dataset.
- Disadvantages
  - Sensitive to the choice of resample size.
  - Greater computation on selection of an optimal resample size.
  - Need perform a rescaling of their output, requires prior knowledge of the convergence rate of estimator.

# Parallel Computing

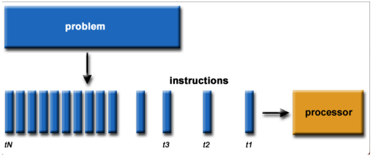



Figure: Serial Computing

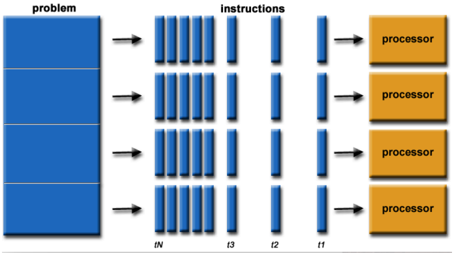



Figure: Parallel Computing

# Parallel Computing in Bootstrap

- embarassingly parallel.
- does not require exchange of information between workers.

# Boostrap using and without using 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
```

# Modern Computing Technology

- Multi-core Server.
- Cloud Computing.
- GPU Computing.

# Basic Idea - Bag of Little Bootstraps(BLB)

For subset size $b < n$,

- Sample $s$ (disjoint) subsets of size $b$ from the original $n$ data points, uniformly at random.
  - Corresponding index multisets: $\mathcal{I}_1, \cdots, \mathcal{I}_s$
  - Empirical distribution: $\mathbb{P}_{n,b}^{(j)} = \frac{1}{b}\sum_{i\in\mathcal{I}_j}\delta_{X_i}$
- For each term $j$,
  - Repeatedly resample $n$ points i.i.d. from $\mathbb{P}_{n,b}^{(j)}$.
  - Compute the estimate on each resample.
  - Form the empirical distribution $\mathbb{Q}_{n,j}^*$ of the computed estimates.
  - Approximate $\xi(Q_n(\mathbb{P}_{n,b}^{(j)})) \approx \xi(\mathbb{Q}_{n,j}^*)$
- $\xi(Q_n(P)) \approx \frac{1}{s}\sum_{j=1}^{s}\xi(Q_n(\mathbb{P}_{n,b}^{(j)})) \approx \frac{1}{s}\sum_{j=1}^{s}\xi(\mathbb{Q}_{n,j}^*)$

# BLB Algorithm

**Input**: Data $X_1, \ldots, X_n$
$\hat{\theta}$: estimator of interest
$\xi$: estimator quality assessment
$b$: subset size
$s$: number of sampled subsets
$r$: number of Monte Carlo iterations
**Output**: An estimate of $\xi(Q_n(P))$

**for** $j \leftarrow 1$ **to** $s$ **do**
  `// Subsample the data`
  Randomly sample a set $\mathcal{I} = \{i_1, \ldots, i_b\}$ of $b$ indices from $\{1, \ldots, n\}$ without replacement
  [or, choose $\mathcal{I}$ to be a disjoint subset of size $b$ from a predefined random partition of $\{1, \ldots, n\}$]
  `// Approximate` $\xi(Q_n(\mathbb{P}_{n,b}^{(j)}))$
  **for** $k \leftarrow 1$ **to** $r$ **do**
    Sample $(n_1, \ldots, n_b) \sim \text{Multinomial}(n, \mathbf{1}_b/b)$
    $\mathbb{P}_{n,k}^* \leftarrow n^{-1} \sum_{a=1}^{b} n_a \delta_{X_{i_a}}$
    $\hat{\theta}_{n,k}^* \leftarrow \hat{\theta}(\mathbb{P}_{n,k}^*)$
  **end**
  $\mathbb{Q}_{n,j}^* \leftarrow r^{-1} \sum_{k=1}^{r} \delta_{\hat{\theta}_{n,k}^*}$
  $\xi_{n,j}^* \leftarrow \xi(\mathbb{Q}_{n,j}^*)$
**end**
`// Average values of` $\xi(Q_n(\mathbb{P}_{n,b}^{(j)}))$ `computed for different data subsets`
**return** $s^{-1} \sum_{j=1}^{s} \xi_{n,j}^*$

# BLB Diagram

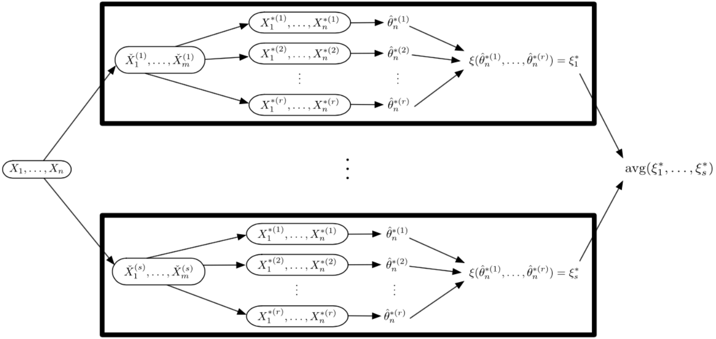



**Figure 1.** The BLB procedure. From the original dataset, $\{X_1, \ldots, X_n\}$, $s$ subsamples of size $m$ are formed. From each of these subsamples, $r$ bootstrap resamples are formed, each of which are conceptually of size $n$ (but would generally be stored as weighted samples of size $m$). The resulting bootstrap estimates of risk are averaged. In a parallel implementation of BLB, the boxes in the diagram would correspond to separate processors; moreover, the bootstrap resampling within a box could also be parallelized.

# Why BLB is fast?

```r
n <- 100000
b <- 100
N <- rnorm(n)
B <- sample(N,b)

mean0 <- function(N,n){
  X <- sample(N, n, replace = TRUE)
  return(mean(X))
}

mean1 <- function(B,n){
X <- sample(B, n, replace = TRUE)
return(mean(X))
}

mean2 <- function(B,n,b){
  X <- as.numeric( rmultinom(1,n,rep(1/b,b)) )
  return(sum(X*B)/n)
}

mean0(N,n)
```

```
## [1] 0.0001172181
```

```r
mean1(B,n)
```

```
## [1] 0.1169493
```

```r
mean2(B,n,b)
```

```
## [1] 0.1211508
```

```r
microbenchmark::microbenchmark(mean0(N,n),mean1(B,n), mean2(B,n,b))
```

```
## Unit: microseconds
##             expr      min       lq        mean    median        uq       max
##      mean0(N, n) 1955.048 2096.614 3192.13140 2447.9895 2911.3800 40233.297
##      mean1(B, n) 1848.265 1870.685 2787.35440 2020.5600 2565.5080 34120.396
##   mean2(B, n, b)   16.077   20.072   33.19767   34.9485   42.2525    55.461
```

# R Package

### Compare BLB & bootstrap

```r
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
```

# R Package

```
###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)
```

# R Package

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

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

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

```r
coefs
```

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

# R Package

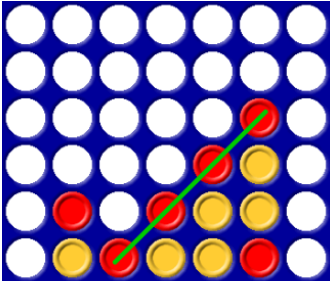
```
### compared with bootstrap
library(boot)
coef_adult <- function(x,d){
  coef(glm(incomebin ~ educationnum,
           data = x[d,], family = binomial()))[2]
}

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
```

# Real Data - Connect-4

Object: Connect four of your checkers in a row while preventing your opponent from doing the same. (Milton Bradley, 1977).
https://en.wikipedia.org/wiki/Connect_Four



# Data Description

This database contains all legal 8-ply positions in the game of connect-4 in which neither player has won yet, and in which the next move is not forced along with their theoretical result.

Number of Instances: 67557
Number of Attributes: 42
Each attribute is corresponding to 1 of the 42 connect-4 squares.
x=player x has taken
o=player o has taken
b=blank
Theoretical Result of first player(win/lose/draw).

The board is numbered like:

| | | | | | | |
|---|---|---|---|---|---|---|
| a6 | b6 | c6 | d6 | e6 | f6 | g6 |
| a5 | b5 | c5 | d5 | e5 | f5 | g5 |
| a4 | b4 | c4 | d4 | e4 | f4 | g4 |
| a3 | b3 | c3 | d3 | e3 | f3 | g3 |
| a2 | b2 | c2 | d2 | e2 | f2 | g2 |
| a1 | b1 | c1 | d1 | e1 | f1 | g1 |

# Simulation(A Kleiner, 2012)

Consider classification model using logistic regression.

- $n = 67557$ observations: $(\mathbf{X_i}, Y_i)$, $\mathbf{X_i} \in \mathbf{R}^{42}$ $Y_i \in \{0, 1\}$.
- $\hat{\theta}_n$ estimates parameter vector in the logistic regression model.
- $\xi$ as a procedure that computes a set of marginal 95% confidence intervals, one for each element of the $\hat{\theta}_n$.
- $b = n^{\gamma}$, $\gamma = 0.6, 0.7, 0.8, 0.9$
- Hyperparameters r and s using adaptive method.
- Average (across dimensions) absolute confidence interval width yielded by each procedure is reported.

# Performance

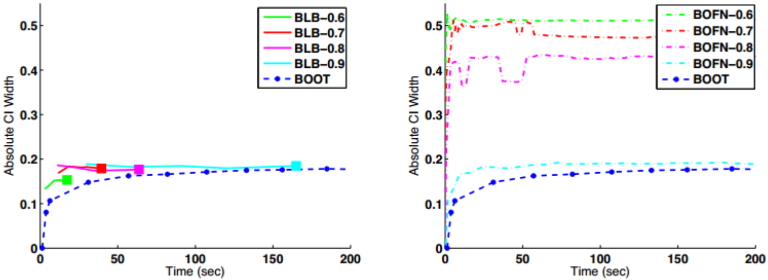



Figure: BLB: Bag of Little Bootstrap, BOOT: Bootstrap, BOFN: b out of n bootstrap

# BLB Advantage

- Retains the generic applicability and statistical efficiency of the bootstrap.
- Suited to modern parallel and distributed computing architecture, computationally efficiency.
- Robust.

# References


- A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan. Scalable Bootstrap for Massive Data, June 29, 2012.
- S. Basiri, E. Ollila, and V. Koivunen, Robust, scalable and fast bootstrap method for analyzing large scale data, *IEEE Transactions on Signal Processing,* 64:1007-1017, 2016.
- A. Kleiner, A. Talwalkar, P. Sarkar, and M. Jordan, The Big Data Bootstrap, Jun 27 2012.
- D. Politis, J. Romano, and M. Wolf. *Subsampling.* Springer, 1999.
- P. J. Bickel, F. Gotze, and W. van Zwet. Resampling fewer than n observations: Gains, losses, and remedies for losses. *Statistica Sinica,* 7:1-31, 1997.
- P. J. Bickel, and A. Sakov. On the Choice of m in the m Out of n Bootstrap and its Application to Confidence Bounds for Extreme Percentiles, June 27, 2005.
- A. Frank and A. Asuncion. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2010. URL

Thank you!