Monte Carlo error in simulation-based statistical analyses: introduction

After Elizabeth Koehler, Elizabeth Brown and Sebastien J.-P. A. Haneuse (2009)

(i) Draw attention on the importance of reporting Monte Carlo (MC) error.

(ii) Provide simple and practical tools for estimating MC error.

(iii) Provide means for determining the number of replications required to achieve a prespecified level of accuracy.

Experiment 1: estimators of operating characteristics of the log-odds ratio (1/5)

Assumed generating process: $Logit(Y = 1 \mid X) = \beta_0 + \beta_1 x$

MC error of an estimator $\hat{\phi}$: $MCE(\hat{\phi}) = \sqrt{var(\hat{\phi})}$

Setting 1:

(i) Quantification of association between Y and X, two binary r.v.

(ii) Assessment of some operating characteristics of the Maximum Likelihood Estimator (MLE) of the slope (i.e. log-odds ratio)

N (sample size) = 100

R (number of replicates) = 100, 500, 1,000, 2,500, 5,000, 10,000

M (number of simulations) = 1,000

Experiment 1: three operating characteristics (2/5)

$\hat{\phi}_R^b$, an estimate of the **percent bias** for the MLE of β_X, defined as:

$$\hat{\phi}_R^b = \frac{1}{R}\sum_{r=1}^{R}\frac{\hat{\beta}_X^r - \beta_X}{\beta_X} * 100$$

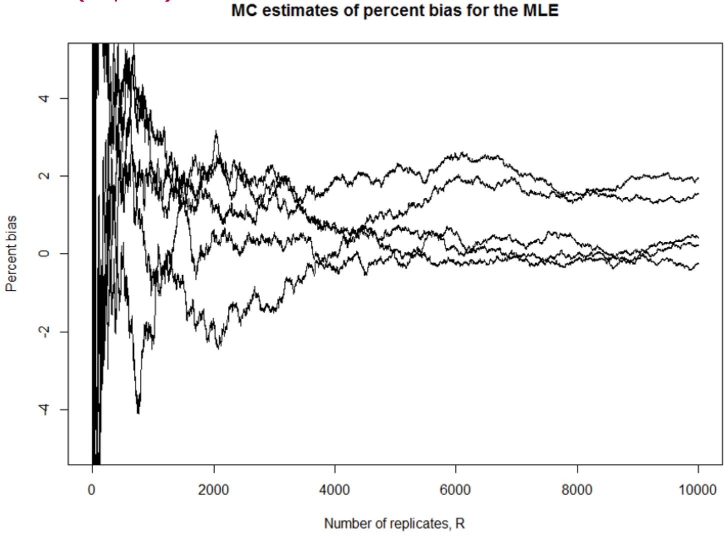
$\hat{\phi}_R^c$, an estimate of the **coverage rate of the 95 % Confidence interval**, defined as:

$$\hat{\phi}_R^c = \sum_{r=1}^{R} \mathbb{1}\left[\hat{\beta}_X^r - 1.96\ se(\hat{\beta}_X^r) \geq \beta_X \leq \hat{\beta}_X^r + 1.96\ se(\hat{\beta}_X^r)\right]$$

$\hat{\phi}_R^p$, an estimate of the **power** to detect an association, defined as:

$$\hat{\phi}_R^p = \sum_{r=1}^{R} \mathbb{1}\left[\left|\frac{\hat{\beta}_X^r}{se(\hat{\beta}_X^r)}\right| < \hat{\beta}_X^r\right]$$

Experiment 1: MC estimates of percent bias (3/5)



Experiment 1: R code to generate the graph (4/5)

```
beta11=numeric(10000)
...
beta61=numeric(10000)

beta0= -1  ; betax = log(2)    # known parameter for logistic regression
Px1=0.3 ; Px0=0.7   # P(X=1) and P(X=0)
levels=c(1,0) ; N1=100 ; X=c(rep(1,30),rep(0,70))  # fixed

set.seed(1)
for(i in 1:10000)
{  z1 = beta0 + betax *X       # linear combination with a bias
   Py1 = 1/(1+exp(-z1))    # pass through an inv-logit function
   y1 <- rbinom(N1,1,Py1)
   beta11[i] <- glm(y1 ~ X, family="binomial"(link='logit'))$coeff[2]  }
...
set.seed(6)
...
# figure 1
maxden=10000 ; den=1:maxden
hplotR61=cumsum(round((round(beta11,2) - (betax*rep(1, maxden))) /
              (betax*rep(1, maxden)),2)*100) / den
...
hplotR65=cumsum(round((round(beta51,2) - (betax*rep(1, maxden))) /
              (betax*rep(1, maxden)),2)*100) / den
plot(hplotR61,type="l", xlim=c(1,maxden), ylim=c(-5,5), main="MC estimates of bias ..."
     ,xlab="Number of replicates R", ylab="Bias", col = 'red3')
lines(hplotR62)
...
lines(hplotR65)
```

Experiment 1: results (5/5)

M=1000

Operating characteristic	Number of replications, R	Min.	Max.	Mean	MCE
Percent bias	100	-22.5	24.4	1.0	6.8
	500	-10.0	11.8	0.9	3.2
	1,000	-6.9	7.7	1.0	2.3
	2,500	-3.7	4.8	1.0	1.4
	5,000	-2.4	3.6	0.9	1.0
	10,000	-1.5	3.1	0.9	0.7
95% CI coverage	100	85.0	100.0	94.7	2.2
	500	90.8	97.4	94.5	1.0
	1,000	92.3	96.5	94.6	0.7
	2,500	93.0	96.0	94.5	0.5
	5,000	93.2	95.5	94.5	0.4
	10,000	93.6	99.4	94.5	0.3
Power	100	20.0	50.0	33.2	4.7
	500	26.2	39.4	33.0	2.2
	1,000	28.4	37.7	33.0	1.5
	2,500	29.3	36.5	33.0	1.0
	5,000	30.7	35.3	33.0	0.7
	10,000	31.4	34.4	33.0	0.5

Experiment 2: quantification of MC error (1/4)

Setting 2:

By the strong Law of Large Numbers, we have that $\hat{\phi}_R \to E[\phi(X)]$

By the Central Limit Theorem, we know that $\sqrt{R}(\hat{\phi}_R - \phi) \to N(0, \sigma^2_\phi)$

We consider the two following measures:

$$(i) \quad MC\hat{E}clt = \frac{\hat{\sigma}_\phi}{R} = \frac{1}{R}\sqrt{\sum_{r=1}^{R}\left(\phi(X) - \hat{\phi}_R\right)^2} \tag{1}$$

$$(ii) \quad MC\hat{E}boot = \frac{1}{B}\sqrt{\sum_{b=1}^{B}\left(\hat{\phi}_R(\mathbf{X}_b^*) - \overline{\hat{\phi}_R(\mathbf{X}_b^*)}\right)^2} \tag{2}$$

Experiment 2: results (2/4)

M=500,000 (in replicating 500 x M=1000)						
				MCE*boot*		
Characteristic	R	Estimated value	MCE*clt*	*B* = 100	*B* = 200	*B* = 500
Mean	**100**	1.0254	0.0096	0.0095	0.0103	0.0094
	500	0.9248	0.0045	0.0037	0.0047	0.0046
	1,000	0.9810	0.0033	0.0035	0.0032	0.0032
	2,500	0.9572	0.0019	0.0022	0.0019	0.0020
	5,000	0.9045	0.0013	0.0014	0.0012	0.0013
	10,000	0.8984	0.0009	0.0010	0.0009	0.0010
MCE	**100**	6.8191	NA	0.0067	0.0071	0.0070
	500	3.2131	NA	0.0029	0.0032	0.0030
	1,000	2.3205	NA	0.0022	0.0022	0.0023
	2,500	1.3603	NA	0.0012	0.0015	0.0013
	5,000	0.9540	NA	0.0009	0.0009	0.0009
	10,000	0.6571	NA	0.0006	0.0007	0.0007

Experiment 2: more results (3/4)

M=1,000

Characteristic	R	Estimated value	MCE*clt*	MCE*boot* $B = 100$	MCE*boot* $B = 200$	MCE*boot* $B = 500$
Mean	**100**	1.0254	0.2156	0.2210	0.2268	0.2128
	500	0.9248	0.1016	0.0958	0.1027	0.1038
	1,000	0.9810	0.0734	0.0724	0.0782	0.0727
	2,500	0.9572	0.0430	0.0416	0.0397	0.0432
	5,000	0.9045	0.0302	0.0308	0.0307	0.0297
	10,000	0.8984	0.0208	0.0209	0.0224	0.0209
MCE	**100**	6.8191	NA	0.1601	0.1548	0.1600
	500	3.2131	NA	0.0678	0.0660	0.0699
	1,000	2.3205	NA	0.0482	0.0524	0.0501
	2,500	1.3603	NA	0.0291	0.0279	0.0293
	5,000	0.9540	NA	0.0211	0.0191	0.0201
	10,000	0.6571	NA	0.0146	0.0128	0.0143

Experiment 2: partial R code (4/4)

```
betaA2=numeric(100)
...
betaF2=numeric(10000)
betamean100=numeric(1000)
...
betamean10000=numeric(1000)

# R=100
set.seed(3)
for(j in 1:1000) {
  for(i in 1:100)
  { beta0= -1     # known parameter for logistic regression
    betax = log(2)   # [1] 0.6931472  parameter for logistic regression
    levels=c(1,0)
    X=c(rep(1,30),rep(0,70))  # fixed
    z1 = beta0 + (betax *X)      # linear combination
    Py1 = 1/(1+exp(-z1))   # inv-logit function
    y1 <- as.numeric(rbinom(100,1,Py1))
    betaA2[i] <- glm(y1 ~ X, family="binomial"(link=logit))$coeff[2]      }
  betamean100[j] = mean(betaA2)       }
...
calcPB <- function(data, index, truth) (mean(data[index]) - truth) / truth * 100

calcSE <- function(data, index) sd(data[index])

mceBoot <- function(data, B, type="", truth=NULL) {...}

## for  M=500,000
# R=100
set.seed(3)
mceBoot(betamean100_2, B=100,type="PB", truth=betax)
# [1] 0.009481879
```

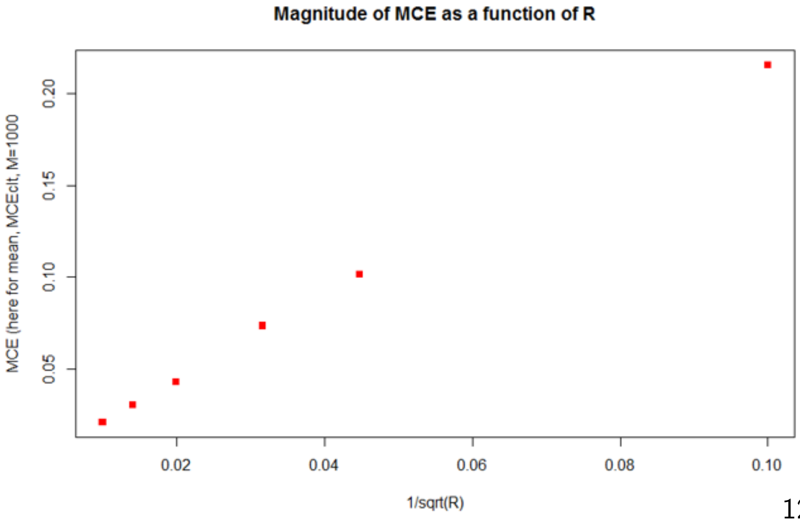
Some remarks

(i) The magnitude of the MC error seems to be linear in $1/\sqrt{R}$, (see plot next slide).

(ii) As $1/\sqrt{R}$ tends to 0, MC error tends to 0, (see plot next slide).

(iii) An idea: maybe we could use a constrained linear regression to estimate R to achieve an acceptable level of MC error.

Magnitude of the MC error as a function of R



Conclusions

(i) The MC error can be more substantial than traditionally though.

(ii) Even after $500,000$ simulations, there is residual uncertainty.

(iii) The magnitude of MC error depends on various factors, i.e. parameters, operating characteristics, variability in the data, etc.

(iv) The MC error can be drastically reduced by increasing R, the number of replicates.

References

Elizabeth KOEHLER, Elizabeth BROWN, and Sebastien J.-P. A. HANEUSE, 2009. On the Assessment of Monte Carlo Error in Simulation-Based Statistical Analyses.

Christian P. ROBERT, George CASELLA, 2005. Monte Carlo Statistical Methods.

Maria L. RIZZO, 2008. Statistical computing with R.

F. H. C. MARRIOTT, 1979. Barnard's Monte Carlo Tests: How Many Simulations?

Full R code of this reproduced study available upon request.