CANNeCTIN
Canadian Network and Centre for Trials Internationally

Seminar Series on Advanced Issues in Clinical Trials Methodology

The challenge of simulating high-dimensional data in methodological research

Tibor Schuster, Ph.D.

December 13, 2013

Department of Epidemiology, Biostatistics and Occupational Health
Tibor.Schuster@mcgill.ca
McGill UNIVERSITY

Centre For Clinical Epidemiology
Lady Davis Institute for Medical Research

CNODES
CANADIAN NETWORK FOR OBSERVATIONAL DRUG EFFECT STUDIES

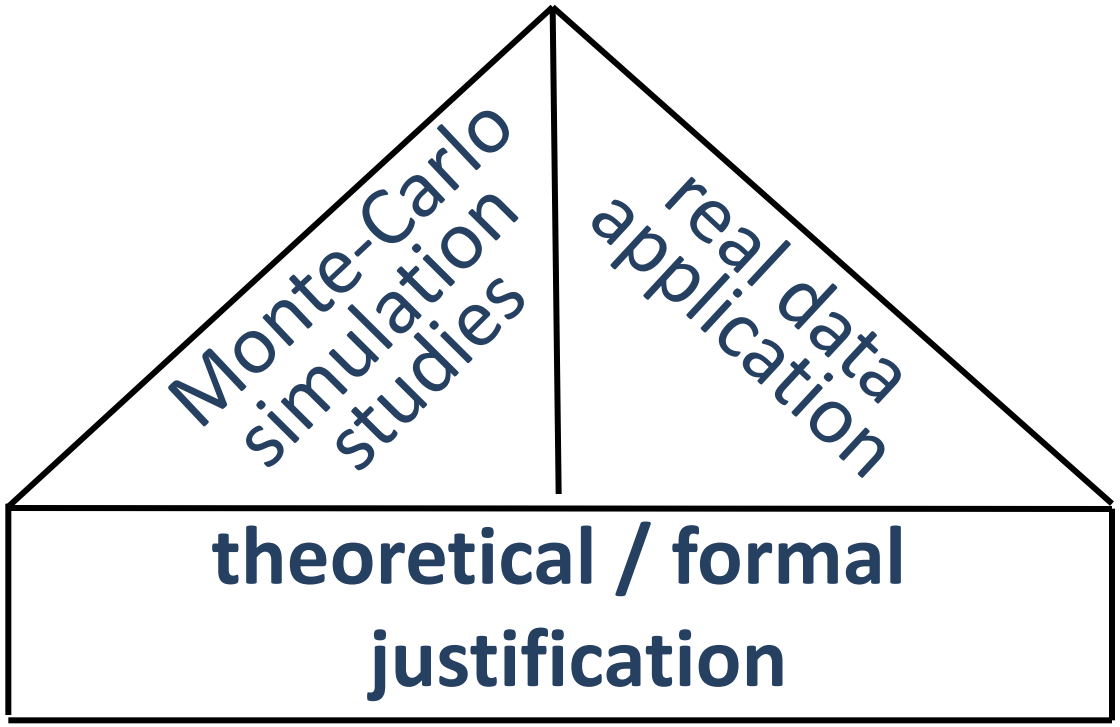
Outline

- Methods research – common practice
- Monte Carlo Simulation studies
- The curse of dimension
- Some solutions
- What else to consider?
- Some useful tools in R

Methods research – common practice

Methodological research in Biostatistics and Epidemiology:

inference, prediction, variable selection, confounder adjustment, missing data …



Monte Carlo Simulation Studies

- repeated random sampling from
  defined (known) target domains / populations

- assuming multivariate variable distributions
  and/or stochastical data generating models

- heuristical calculation (estimation) of interesting
  statistical quantities such as probabilities and
  parameters

Monte Carlo Simulation Studies

empirical assessment of..

- (parameter) estimators → bias, standard error
  mean squared error

- test performance → type I and type II error, coverage

- prediction / classification error

- variable selection performance

Example: Treatment Effect Study

Interest:

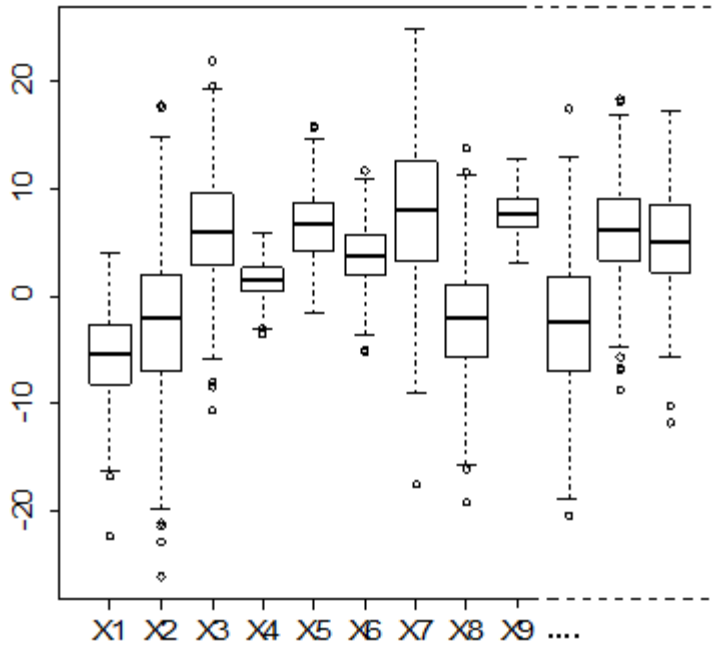
Performance assessment of method *M*
in estimating the treatment effect θ_E on the outcome Y.

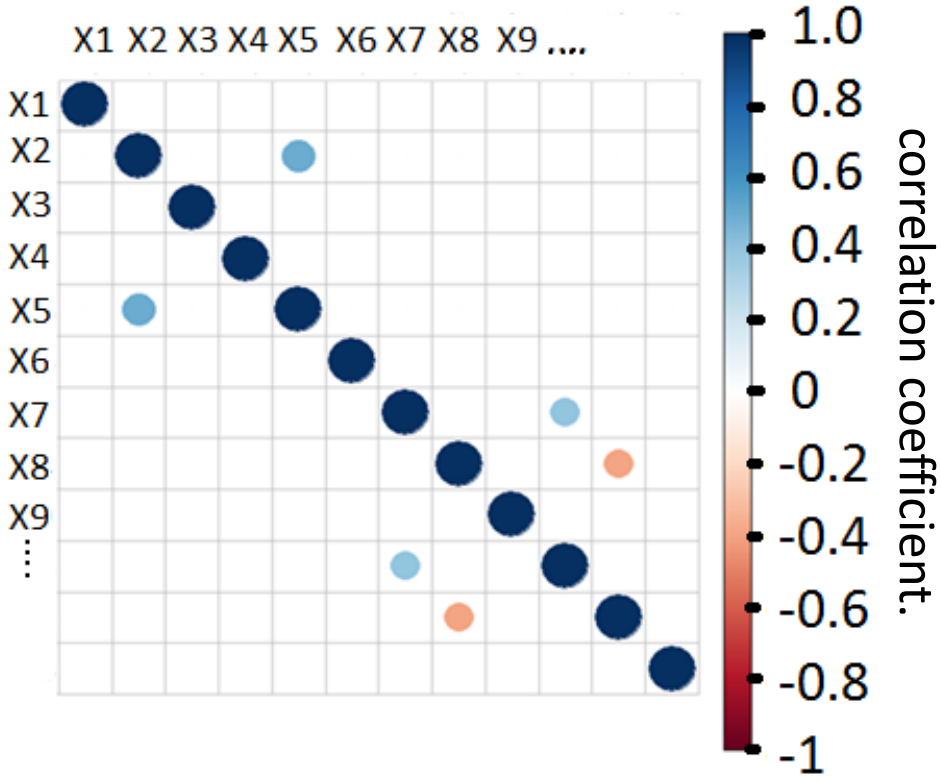
Variables:

Y binary outcome
E binary treatment/exposure
X covariates, associated with E and/or Y

Step 1 – covariate sampling

X : random sample from a multivariate distrib.
e.g. mv normal, mv binomial, mixture, **real data**





Step 2 - model specification

i) treatment model

For example, a generalized linear model

$$P(E=1) = g(a + \beta_{E1} X_1 + \ldots + \beta_{Ep} X_p)$$

ii) outcome model:

$$P(Y=1) = g(b + \theta_E E + \beta_{Y1} X_1 + \ldots + \beta_{Yp} X_p)$$

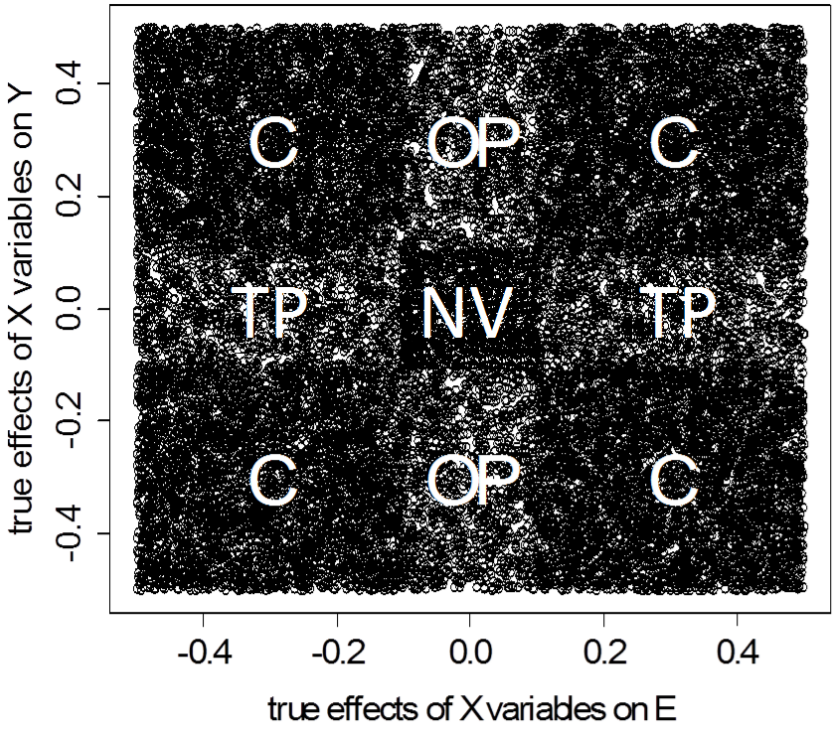
Step 3 - parameter specification

if $\beta_{Ej} = 0$ and $\beta_{Yj} \neq 0 \Rightarrow X_j$ **outcome predictor (OP)**

if $\beta_{Ej} \neq 0$ and $\beta_{Yj} = 0 \Rightarrow X_j$ **treatment predictor (TP)**

if $\beta_{Ej} \neq 0$ and $\beta_{Yj} \neq 0 \Rightarrow X_j$ **confounder (C)**

if $\beta_{Ej} = 0$ and $\beta_{Yj} = 0 \Rightarrow X_j$ **noise variable (NV)**



$$Y = f(E, \mathbf{X}, \theta, \beta)$$
$$E = g(\mathbf{X}, \beta)$$

$$Y \sim F_Y, E \sim F_E, \mathbf{X} \sim F_X,$$

Don't forget: even if θ and β are assumed to be fixed – we have sampling error:

E	X	Y
0	1	1
0	0	1
1	1	0

E	X	Y
0	1	0
0	0	1
1	1	1

E	X	Y
0	1	0
0	0	0
1	1	1

E	X	Y
0	1	0
0	0	0
1	1	0

E	X	Y
0	1	1
0	0	0
1	1	0

E	X	Y
0	1	0
0	0	1
1	1	0

E	X	Y
0	1	1
0	0	1
1	1	1

Therefore, we are dealing with sample distributions:

$$F_Y^s, F_E^s, F_{\mathbf{X}}^s, F_\theta^s, F_\beta^s$$

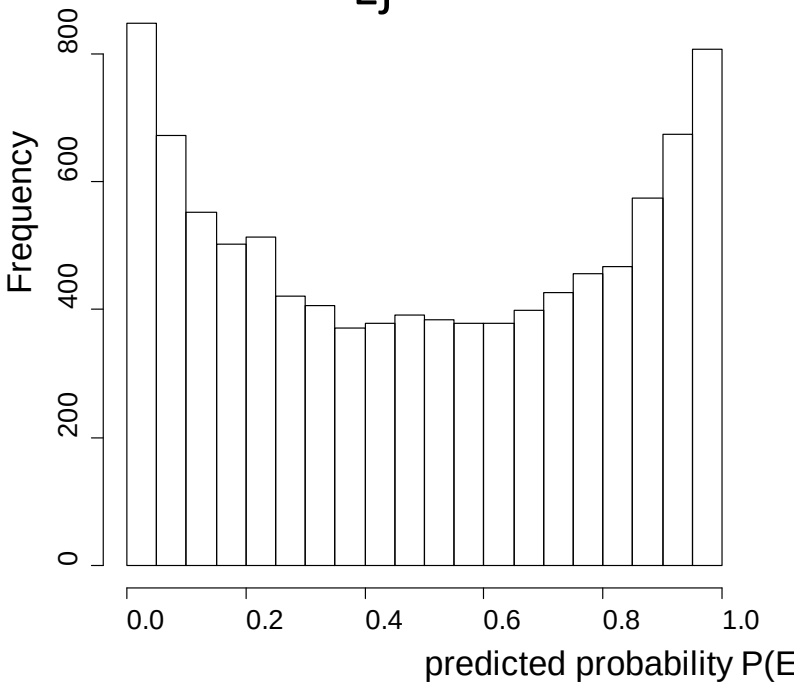
The curse of dimension

Let's try to generate two datasets:

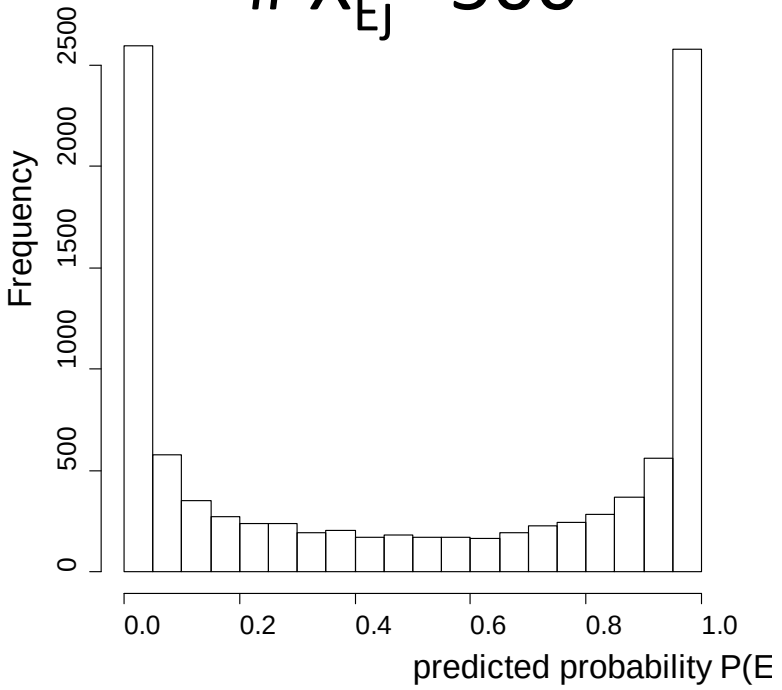
n=10,000; 100 & 500 independent noise variables

$X_{Ej} \sim N(0,1)$, $\beta_{Ej} \sim N(0,0.2)$

X_{Ej} =100



X_{Ej} =500



- with increasing dimensionality, individual absolute prediction error $|\hat{P}_{i|\boldsymbol{x}} - P_i|$ increases → variance inflation

- in absence of over fitting, predicted probabilities for *E* or *Y* close to 0 or 1 are uncommon in real data situations (unless we collected strong predictors, but noise variables are not predictors)

Do such simulation studies represent reality?

Some solutions

specification of appropriate covariance matrices for $\boldsymbol{X}$ and $\boldsymbol{\beta}$

i) decrease value of $diag[cov(\boldsymbol{\beta})]$ *if* $\#X_j$ *increases*

ii) decrease value of $diag[cov(\boldsymbol{X})]$ *if* $\#X_j$ *increases*

iii) if $cov(X_j, X_k) > 0$ *set* $cov(\beta_j, \beta_k) < 0$
if $cov(X_j, X_k) < 0$ *set* $cov(\beta_j, \beta_k) > 0$

iv) strength of correlation of X_j *and* X_k *should be proportional to strength of correlation of* β_j *and* β_k

Some solutions

- sampling of $\boldsymbol{X}$ and β from multivariate distributions with constraints on the marginal densities of E and Y:

→ sample β and verify for each consecutively sampled $\boldsymbol{X}_i$ that

$\gamma_{lower} < P(E_i) < \gamma_{upper}$ and $\xi_{lower} < P(Y_i) < \xi_{upper}$

Some solutions

I. using large N (e.g. 100,000),

sample $\boldsymbol{X}$ and $\boldsymbol{\beta}$.

Then, find the scaling parameter ϕ so that for:

$$p_E = P(E=1) = \boldsymbol{g}(a + \phi\,\beta_{E1}\,X_1 + \ldots + \phi\,\beta_{Ep}\,X_p)$$

$$P(p_E < \gamma_{\text{lower}}) < c_{tol} \quad \text{and} \quad P(p_E > \gamma_{\text{upper}}) < c_{tol}$$

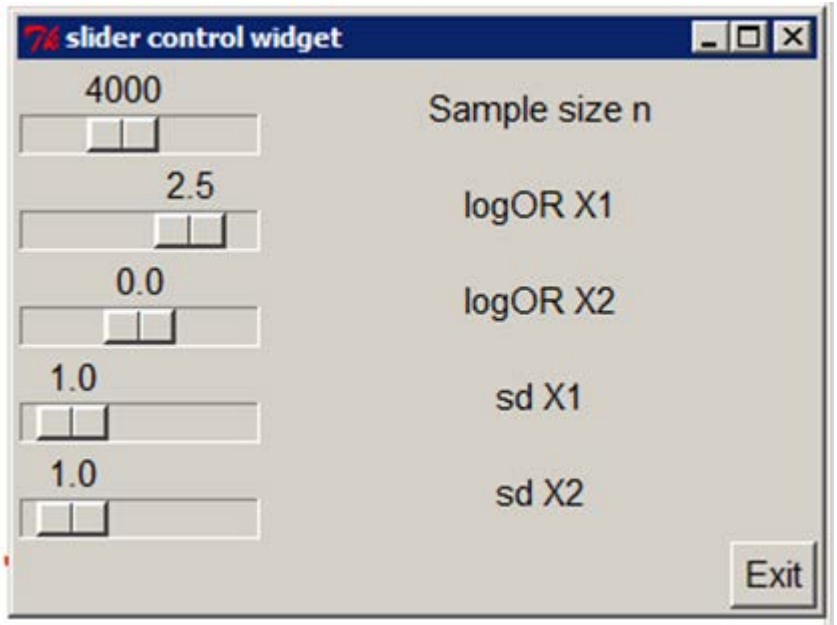
II. repeat sampling of $\boldsymbol{X}$ and $\boldsymbol{\beta}$ for desired sample size and rescale either $\boldsymbol{\beta}$ or $\boldsymbol{X}$ with ϕ

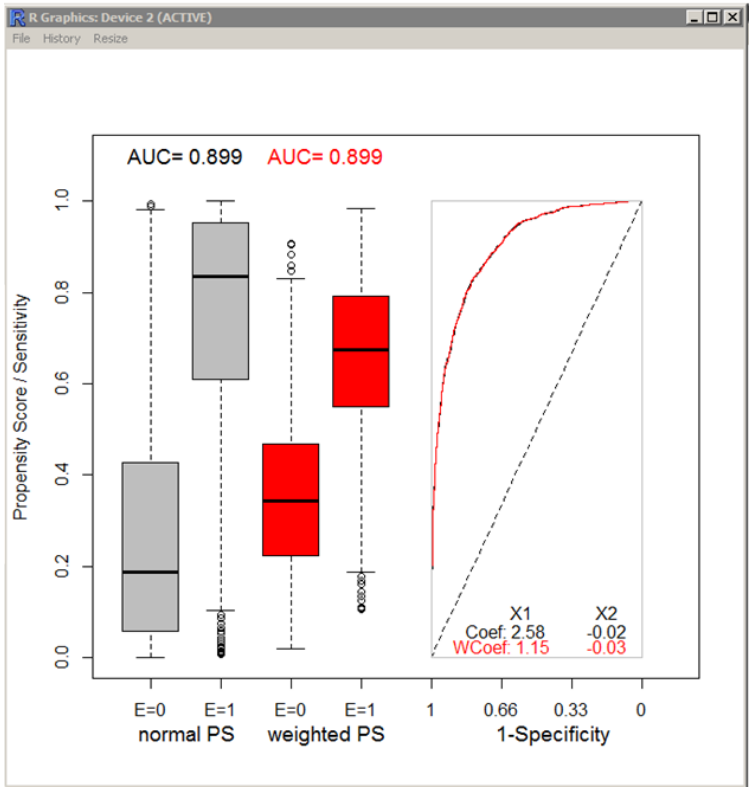
What else to consider?

Before starting a simulation project:

- do we really need simulation?
- parameters varying among the simulation runs (e.g. prevalences, effects, sample sizes, variable types, model types..)
- how to monitor simulation results?
- how to store simulation results?
- what data need to be stored?
- setting seed values to ensure reproducibility

Small pre-simulation studies can help to avoid time inefficiency and answer principal questions...





R function “slider”, Greg Snow (2012). TeachingDemos: Demonstrations for teaching and learning. R package version 2.8. http://CRAN.R-project.org/package=TeachingDemos

Some useful tools in 

R Core Team.
R: A language and environment for statistical computing.
R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

avoiding loops..

Three simulation parameters:

```
> a<-c(0.2,0.4)
> b<-c(1,2,3)
> c<-c("A","B")
```

2 x 3 x 2 = 12 simulation settings

```
> expand.grid(a,b,c)
```

1	0.2	1	A
2	0.4	1	A
3	0.2	2	A
4	0.4	2	A
5	0.2	3	A
6	0.4	3	A
7	0.2	1	B
8	0.4	1	B
9	0.2	2	B
10	0.4	2	B
11	0.2	3	B
12	0.4	3	B

Monitoring..

```
starttime<-proc.time()[3] )    # initiating start time
for (i in 1:Nsim)              # begin simulation loop
  {
   ...program code …
    eHrs<-floor((proc.time()[3]- starttime)/3600)
    eMin<-floor((eHrs -floor(eHrs))*60)
    eSec<-floor((eMin -floor(eMin))*60)
    rHrs<-floor((eHrs /i)*(Nsim-i))
    rMin<-floor((rHrs -floor(rHrs))*60)
    rSek<-floor((rMin -floor(rMin))*60)
sink("c:\\folder\\simulationTime.txt", append=F)
print(list(run=paste(i,Nsim,sep="/"),elapsed=paste(eHrs,eMin,eSec)
        remaining=paste(rHrs,rMin,rSec)))
sink()
  } # end simulation loop
```

Other useful R functions..

optim: General-purpose optimization based on Nelder–Mead, quasi-Newton and conjugate-gradient algorithms.

optimize: one dimensional optimization

corrplot: Visualization of a correlation matrix. Taiyun Wei (2012). http://CRAN.R-project.org/package=corrplot

memory problems?

"Package ***bigmemory*** supports the creation, storage, access, and manipulation of massive matrices. These matrices are allocated to shared memory and may use memory-mapped files".

source: http://www.bigmemory.org/
(accessed: Dec 11, 2013)