Running large databases: PIM

# Introduction

In this report, we try to fit PIM models when is large. As the estimation procecures uses the set of *pseudo-observations*, the computation time increases exponentially with : .

# Data generation

We will generate data according to the normal linear model: with . Also, are IID and . The predictor takes values uniformly within where . For instance, take . Then generating data and fitting the PIM model in R is done through:

library(pim)  
library(nleqslv)  
library(data.table)  
library(ggplot2)

# Parameters  
n <- 50  
u <- 1  
alpha <- 5  
sigma <- 1  
trueBeta <- alpha/(sqrt(2) \* sigma)  
  
# Generate predictor  
X <- runif(n = n, min = 0.1, max = u)  
# Generate data  
Y <- alpha\*X + rnorm(n = n, mean = 0, sd = sigma)  
  
# PIM package beta parameter  
BETAvalue <- try(pim(formula = Y ~ X, link = 'probit', model = 'difference')@coef, silent = TRUE)  
data.frame(BETAvalue,trueBeta)

BETAvalue trueBeta  
X 4.456021 3.535534

# Increasing N

We take the code from above, and run this while increasing initially from to .

## Locally

First, we test this locally. The vector is given by:

n <- round(seq(10,25000,length.out = 25), 0)  
n

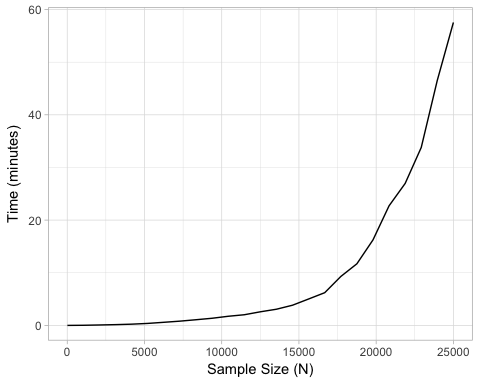
[1] 10 1051 2092 3134 4175 5216 6258 7299 8340 9381 10422  
[12] 11464 12505 13546 14588 15629 16670 17711 18752 19794 20835 21876  
[23] 22918 23959 25000

The estimated beta coefficients and time needed to estimate the model for each sample size are stored in a text file. And plotted below.

# Text files: time is measured in minutes  
estimBeta <- fread('/Users/hanbossier/Dropbox/Mastat/Thesis/ResultsTimeNeeded/beta\_501.txt', data.table = FALSE, col.names = 'EstimatedBeta')  
TimeNeeded <- fread('/Users/hanbossier/Dropbox/Mastat/Thesis/ResultsTimeNeeded/time\_501.txt', data.table = FALSE, col.names = 'TimeInMin')  
# Overview  
local <- data.frame(EstimatedBeta = estimBeta, TrueBeta = trueBeta, SampleSize = n, TimeInMin = TimeNeeded);local

EstimatedBeta TrueBeta SampleSize TimeInMin  
1 3.710759 3.535534 10 2.161145e-04  
2 3.284670 3.535534 1051 2.462467e-02  
3 3.525353 3.535534 2092 8.193035e-02  
4 3.523066 3.535534 3134 1.468487e-01  
5 3.565256 3.535534 4175 2.352749e-01  
6 3.618371 3.535534 5216 3.785054e-01  
7 3.559371 3.535534 6258 5.913721e-01  
8 3.522990 3.535534 7299 8.094707e-01  
9 3.456785 3.535534 8340 1.077774e+00  
10 3.566096 3.535534 9381 1.357088e+00  
11 3.537316 3.535534 10422 1.733767e+00  
12 3.520023 3.535534 11464 2.024379e+00  
13 3.531523 3.535534 12505 2.585303e+00  
14 3.550803 3.535534 13546 3.073942e+00  
15 3.499380 3.535534 14588 3.837159e+00  
16 3.525123 3.535534 15629 5.019652e+00  
17 3.519586 3.535534 16670 6.213987e+00  
18 3.507788 3.535534 17711 9.291136e+00  
19 3.533566 3.535534 18752 1.170329e+01  
20 3.555743 3.535534 19794 1.623747e+01  
21 3.571614 3.535534 20835 2.271294e+01  
22 3.527799 3.535534 21876 2.695617e+01  
23 3.537034 3.535534 22918 3.378866e+01  
24 3.502156 3.535534 23959 4.655588e+01  
25 3.554781 3.535534 25000 5.753793e+01

# Plot  
ggplot(local, aes(x = SampleSize, y = TimeInMin)) + geom\_line() +   
 scale\_x\_continuous(name = "Sample Size (N)") + scale\_y\_continuous(name = "Time (minutes)") + theme\_light()



## HPC

We execute the same code on the HPC infrastructure of UGent and iterated for times. The results are stored externaly in text files caled *beta\_N\_n\_K\_k.txt* and *time\_N\_n\_K\_k.txt* in which small *n* and small *k* represent the sample size and iteration respectively.

# Data directory  
dataWD <- '/Volumes/2\_TB\_WD\_Elements\_10B8\_Han/PhD/BigDataPIM/TimeNeeded/output/separate'  
# Number of simulations  
nsim <- 500  
betavalues <- array(NA, dim = c(length(n), nsim))  
recordedTime <- array(NA, dim = c(length(n), nsim))  
# Run loop over n and nsim  
for(i in 1:length(n)){  
 for(k in 1:nsim){  
 # Read in text files: estimated beta and time needed to estimate  
 beta.tmp <- try(fread(paste(dataWD, '/beta\_N\_', n[i], '\_K\_', k, '.txt', sep = '' ), data.table = FALSE, verbose = FALSE), silent = TRUE)  
 if(class(beta.tmp) == 'try-error'){   
 next  
 }else{  
 betavalues[i,k] <- as.numeric(beta.tmp)  
 }  
 time.tmp <- try(fread(paste(dataWD, '/time\_N\_', n[i], '\_K\_', k, '.txt', sep = '' ), data.table = FALSE, verbose = FALSE), silent = TRUE)  
 if(class(time.tmp) == 'try-error'){  
 next  
 }else{  
 recordedTime[i,k] <- as.numeric(time.tmp)  
 }  
 }  
}  
  
# How many values have we recorded?  
data.frame(EstimatedBeta = apply(betavalues, 1, mean), TrueBeta = trueBeta, SampleSize = n, TimeInMin = apply(recordedTime, 1, mean))

EstimatedBeta TrueBeta SampleSize TimeInMin  
1 4.271203 3.535534 10 0.0005379234  
2 3.546431 3.535534 1051 0.0253973256  
3 3.543821 3.535534 2092 0.0845617874  
4 3.534804 3.535534 3134 0.1882360769  
5 3.535431 3.535534 4175 0.3400619260  
6 3.534056 3.535534 5216 0.5492313999  
7 3.537546 3.535534 6258 0.8752335478  
8 3.535204 3.535534 7299 1.4030073936  
9 3.533692 3.535534 8340 2.0160403477  
10 NA 3.535534 9381 NA  
11 NA 3.535534 10422 NA  
12 NA 3.535534 11464 NA  
13 NA 3.535534 12505 NA  
14 NA 3.535534 13546 NA  
15 NA 3.535534 14588 NA  
16 NA 3.535534 15629 NA  
17 NA 3.535534 16670 NA  
18 NA 3.535534 17711 NA  
19 NA 3.535534 18752 NA  
20 NA 3.535534 19794 NA  
21 NA 3.535534 20835 NA  
22 NA 3.535534 21876 NA  
23 NA 3.535534 22918 NA  
24 NA 3.535534 23959 NA  
25 NA 3.535534 25000 NA

We were only able to reliably run the model when . When was higher, we ran out of memory to solve the estimating equations.