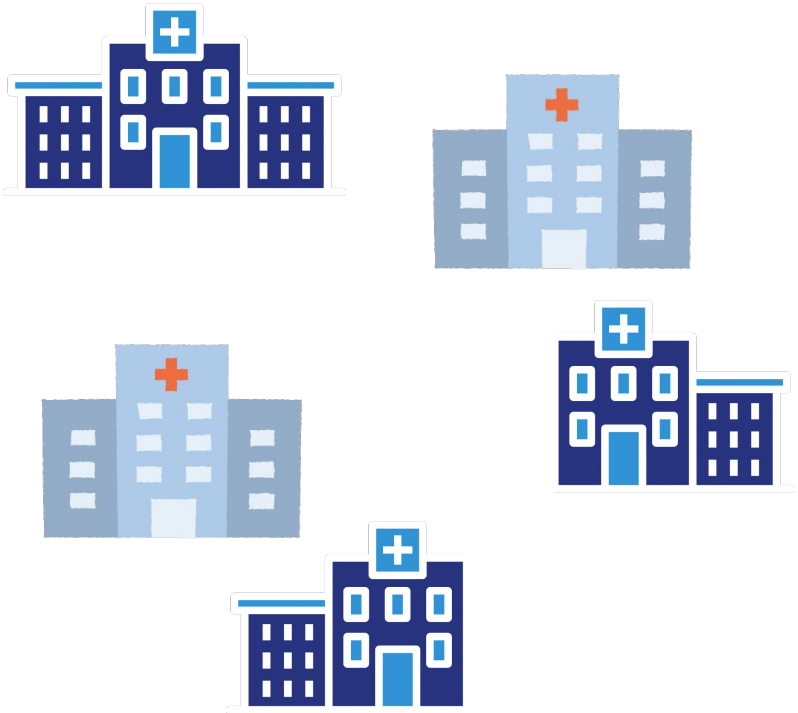
The Hierarchical Logistic Model for Multicenter Clinical Trials

Group 3A: Kate Colvin, Xuanyu Guo, Dang Lin, and Boxiang Tang

A multicenter clinical trial is a trial that enrolls participants at multiple different medical institutions.



Benefits include:

- being able to enroll more patients into the trial
- greater generalizability of trial results.

Different trial sites can have different characteristics, such as patient demographics or treatment practices, that can bias outcomes and complicate the statistical analysis

Additional challenges in evaluating clinical trial methodology are the cost of conducting a trial and the accessibility of this data to external researchers.

Jo, Daehyun. “The interpretation bias and trap of multicenter clinical research.” The Korean journal of pain. vol. 33,3 (2020): 199-200. doi:10.3344/kjp.2020.33.3.199

Simulation studies can play a key role in designing and analyzing clinical trials by improving their efficiency and reliability

Pharmaceutical Research, Vol. 17, No. 3, 2000

Commentary

Clinical Trial Simulation in Drug Development

Peter L. Bonate[1,2]

Received July 6, 1999; accepted December 11, 1999

Clinical trial simulation is the application of old technologies, e.g., Monte Carlo simulation, to a new problem, that problem being how to maximize the information content obtained during the drug development process with an intent to have the greatest chance of "success" in a clinical trial. When the information content of the drug is high, then simulation provides a method to synthesize that information into a coherent package that indicates the sponsor has good control over the pharmacology of the drug. From a purely financial point of view, what simulation offers pharmaceutical companies is the possibility of reducing the number of required studies, maximizing the chances for success in a clinical trial, and possibly shortening development time; all outcomes which will reduce drug development costs. The purpose of this paper is to introduce clinical trial simulation to the reader by discussing its potential in drug development, to briefly review the literature, and to make recommendations and caveats regarding its use.

KEY WORDS: Monte Carlo; computer assisted trial design; modeling.

INTRODUCTION

Although we are not even aware of it, everyday we reap the benefits of simulation. In doing a very brief search on the Internet for uses of simulation in everyday life, one can find billion dollars over the next ten years to develop simulation technology to validate the reliability of nuclear bombs without actually detonating them. Recently, it has been alleged that China has stolen state secrets from the United States. Part of

We designed a simulation study to determine optimal Monte Carlo sampling strategies to estimate the population-level probability of adverse events in a hypothetical multicenter clinical trial.

Method	Estimated_Probability	Bias	Variance	CPU_Time
Simple Monte Carlo Control Variates Importance Sampling	?	?	?	?

Probability of An Adverse Event

Probability of an adverse event for patient i in clinic j:

$$p_{i,j} = logit^{-1}(\alpha + b_j + \beta X_i),$$

Overall (population-level) probability of an adverse event:

$$P(\text{Adverse Event}) = \int \int logit^{-1}(\alpha + b + \beta x) f(b) f(x) dbdx$$

Parameters in the Equation

$$p_{i,j} = logit^{-1}(\alpha + b_j + \beta X_i),$$

X_i : Each patient i has a continuous risk factor Xi (disease severity score)

b_j : Random effects at the clinic level

α : The baseline log-odds of an adverse event when Xi = 0 and there is no clinic-level random effect

β : The effect size of the patient-level covariate Xi, controlling for clinic-level random effects

Define α and β

Let α = -2

$$p_{\text{baseline}} = \frac{e^{-2}}{1 + e^{-2}} \approx \frac{0.1353}{1.1353} \approx 0.1192$$

The baseline probability of an adverse event is about 11.92%

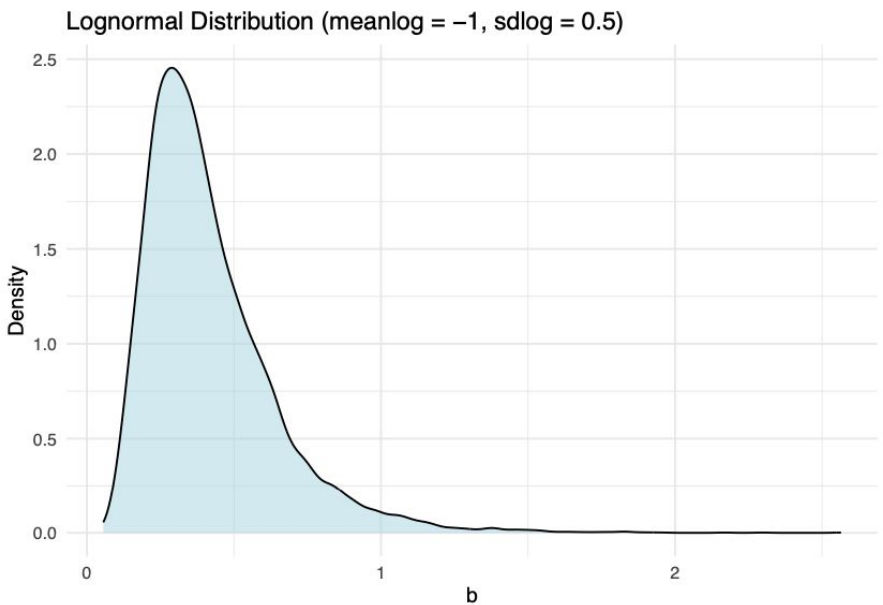
Let β = 0.5

$$e^{0.5} \approx 1.65$$

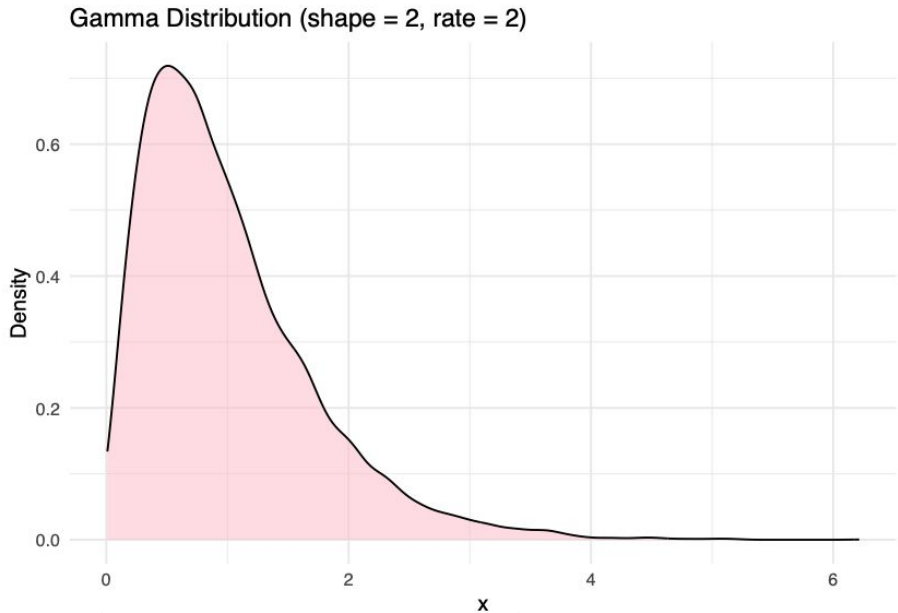
Each unit increase in Xi increases the odds of an adverse event by 65%

Specify the Distributions for b and x

$$b \sim \text{Lognormal}(-1, 0.5)$$



$$x \sim \text{Gamma}(2, 2)$$



highly skewed and heavy-tailed (sample size: 10000)

Simple Monte Carlo Integration

$$P_{Simple}(AdverseEvent) = \frac{1}{N}\sum_{i=1}^{N} logit^{-1}(f(x_i, b_i))$$

```
logit_inverse_function <- function(x) {
  return(exp(x) / (1 + exp(x)))
}

## Calculating the probability of an adverse event based on pre-sampled b and x
p_ij <- logit_inverse_function(alpha + b_samples + beta * x_samples)

p_adverse_event_smc <- mean(p_ij)
var_smc <- var(p_ij)
```

The Control Variate method improves efficiency by reducing estimate variance through the use of a correlated control variate.

Simple MC

$$\int f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Control Variate MC

$$\int f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} [f(U_i) - c^*(U_i - \bar{U})]$$

Introducing control variate (U) correlated with f(x) and has known mean

$$c^* = \frac{Cov(f(U), U)}{Var(U)}$$

Applying the control variate method to estimate P(Adverse Event) leads to a 0.278% correction to the simple Monte Carlo estimation.

Probability of an adverse event

$$f(x, b) = logit^{-1}(\alpha + b + \beta x)$$

$$b_j \sim Lognormal(\mu = -1, \sigma = 0.5)$$

$$x_i \sim Gamma(\alpha = 2, \lambda = 2)$$

Chosen control variate and exact mean

$$U = \alpha + b + \beta x$$

$$\bar{U} = E[\alpha + b + \beta x] = \alpha + e^{\frac{\mu+\sigma^2}{2}} + \beta\left(\frac{\alpha}{\lambda}\right)$$

$$P(Adverse\ Event) \approx \frac{1}{N}\sum_{i=1}^{N}[logit^{-1}(\alpha + b_i + \beta x_i) - c^*(\alpha + b_i + \beta x_i - \bar{U})]$$

0.2021 -1.0795 -1.0831

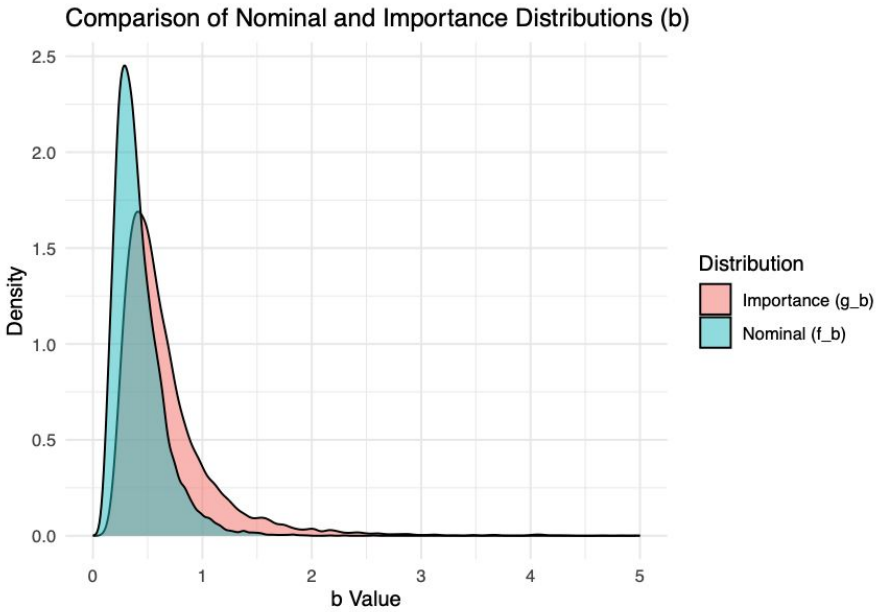
$$Prediction\ Variance = Var\big(f(x, b)\big) - \frac{Cov(f(U), U)^2}{Var(U)}$$

```r
Y <- logit_inverse_function(alpha + b_samples + beta * x_samples)
E_Y <- mean(Y)
var_Y <- var(Y)

## U is the logit transformed Y (i.e. the original argument to the inverse logit)
U <- alpha + b_samples + beta * x_samples
E_U <- mean(U)
var_U <- var(U)

## E(Gamma(2,2)) = 1, E(Log-Normal(-1, 0.5)) = exp(-1 + (0.5^2)/2)
E_U_exact <- alpha + exp(-1 + (0.5^2)/2) + beta*1

cov_Y_U <- cov(Y, U)
c_star <- -cov_Y_U / var_U

## Applying covariate correction
Y_cv <- Y + c_star * (U - E_U_exact)
E_Y_cv <- mean(Y_cv)
var_Y_cv <- var_Y - (cov_Y_U ^ 2) / var_U

cpu_time_cv <- Sys.time() - start_time_cv

control_variate_df <- data.frame(
  Estimates = c("Mean", "Variance"),
  Without_Control_Variate = c(E_Y, var_Y),
  With_Control_Variate = c(E_Y_cv, var_Y_cv)
)
```

Importance Sampling

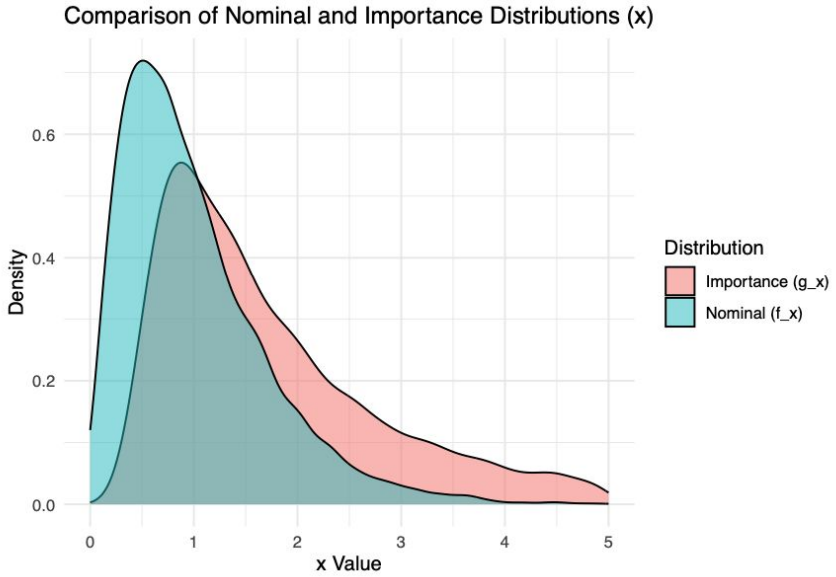
$$b \sim \text{Lognormal}(-1, 0.5)$$

$$b \sim \text{Inverse-Gamma}(4,\ 2)$$



$$X \sim \text{Gamma}(2, 2)$$

$$X \sim \text{Inverse-Gamma}(2,\ 2.6)$$



Importance Sampling

$$P_{IS}(AdverseEvent) = \frac{1}{N}\sum_{i=1}^{N}\frac{f_X(x_i)f_B(b_i)}{g_X(x_i)g_B(b_i))}logit^{-1}(f(x_i,b_i))$$

```r
# Nominal Distribution
f_b <- function(b) {
  return(dlnorm(b, meanlog = -1, sdlog = 0.5))
  }
f_x <- function(x) {
  return(dgamma(x, shape = 2, rate = 2))
  }

# Importance Distribution
g_b <- function(b) {
  return(dinvgamma(n, alpha = 4, beta = 2))
}
g_x <- function(x) {
  return(dinvgamma(n, alpha = 2, beta = 2.6))
}
```

Importance Sampling

$$P_{IS}(AdverseEvent) = \frac{1}{N}\sum_{i=1}^{N} \frac{f_X(x_i)f_B(b_i)}{g_X(x_i)g_B(b_i))} logit^{-1}(f(x_i, b_i))$$

```
weights <- function(b, x) {
  return((f_b(b) * f_x(x)) / (g_b(b) * g_x(x)))
}

b_samples_importance <- rinvgamma(n, alpha = 4, beta = 2)
x_samples_importance <- rinvgamma(n, alpha = 2, beta = 2.6)

p_ij_is <- logit_inverse_function(alpha + b_samples_importance + beta *
x_samples_importance)
w <- weights(b_samples_importance, x_samples_importance)

E_is <- sum(w * p_ij_is) / sum(w)
## Variance expression
var_is <- sum(w * (p_ij_is - E_is)^2) / sum(w)
```

Results

Table 1: Comparison of Sampling Methods (Bias, Variance, and CPU Time)

Method	Estimated_Probability	Bias	Variance	CPU_Time
Simple Monte Carlo	0.2606976	0.0006902084	0.0072318122	0.007517099 secs
Control Variates	0.2599719	-0.0000355375	0.0000800636	0.008687019 secs
Importance Sampling	0.2601710	0.0001635989	0.0032549553	0.016755819 secs

Simple Monte Carlo: fastest method with highest variance

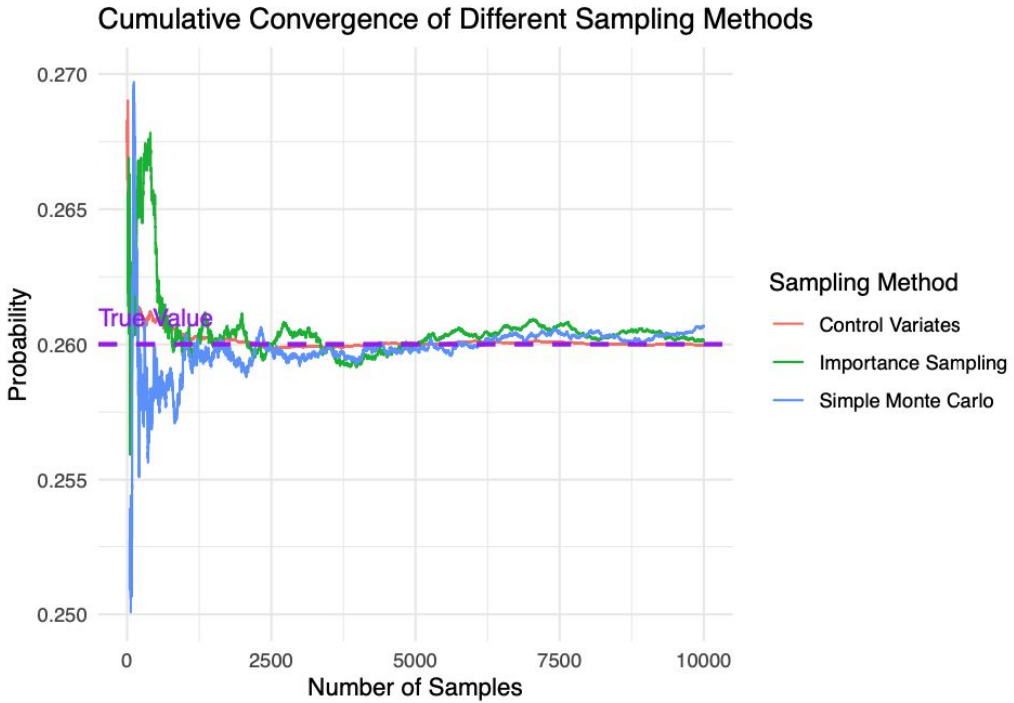
Control Variates:

lowest bias and variance, slightly more computation time than SMC

Importance Sampling:

significantly reduces the bias and variance compared to SMC, most computationally expensive

Results

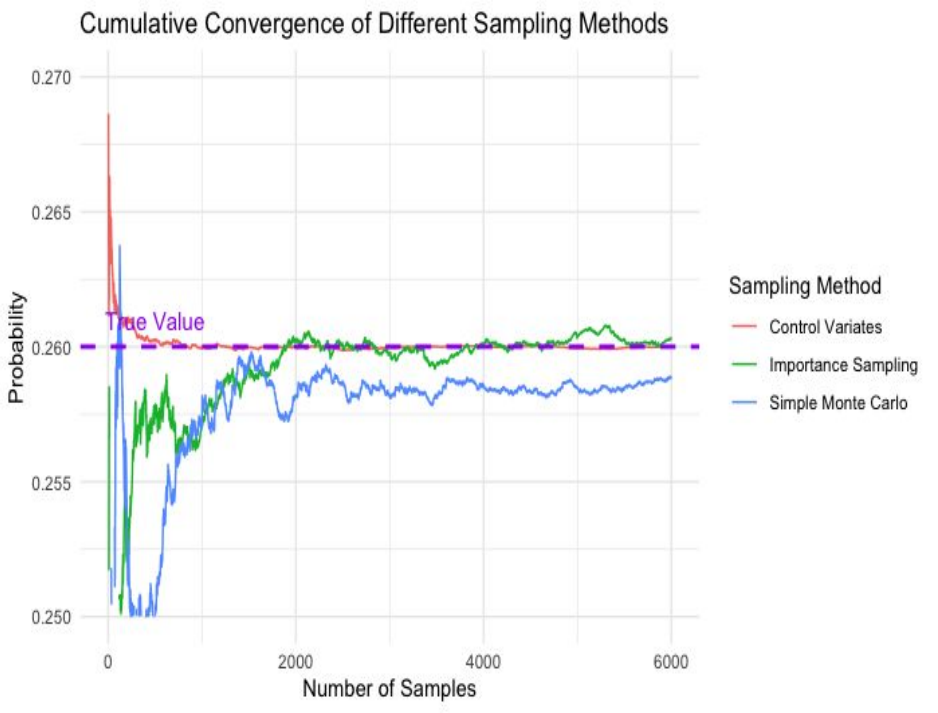


Simple Monte Carlo (blue line):
Large fluctuations, especially in the first 2,500 samples, and takes longer to stabilize.

Control Variates (red line):
The smoothest and fastest to converge. It stabilizes around the true value within the first 1,000 samples.

Importance Sampling (green line):
Shows some initial fluctuations but becomes much more stable over time.

Results



Convergence Speed :

IS converges faster than SMC

CV converges the fastest

Trade-offs in Monte Carlo Sampling

🌟 **Control Variates (CV) is the best method overall, balancing accuracy, variance reduction, and computational efficiency.**

🌟 **Importance Sampling (IS) is also effective but introduces a slight bias and requires more computation.**

🌟 **Simple Monte Carlo (SMC) is useful for quick estimates but has the highest variance and slowest convergence.**

Practical Applications in Clinical Trials

- **Optimizing Multicenter Clinical Trials**

  -> Test how changes in patient demographics or site-specific factors impact adverse event probability.
  -> Enhance trial design, reduce statistical errors, and improve result reliability.

- **Reducing Trial Costs & Accelerating Decision-Making**

  -> Fewer required samples → Lower costs → Faster regulatory approval.
  -> Can assist in selecting appropriate patients or clinical sites for trials, improving trial success rates.

- **Broader Healthcare Applications (Resource Allocation & Personalized Medicine)**

  -> Optimize hospital resource distribution, improving healthcare system efficiency.
  -> Predict patient risk and personalize treatment plans.

Conclusions

- Both the Control Variates (CV) method and Importance Sampling (IS) effectively reduce variance compared to Simple Monte Carlo (SMC), with a moderate increase in computational cost.
- CV stabilizes more quickly and smoothly around the true value, making it the most accurate and efficient method.
- Computational cost trade-off:
  -> IS > CV > SMC
- Variance reduction performance:
  -> CV achieves a 90× variance reduction while maintaining near-zero bias.
  -> IS also reduces variance but has a slightly higher bias and greater computational cost, which may limit its advantage.
- Method selection depends on balancing accuracy and computational efficiency.

Method	Variance Reduction	Convergence Speed	Computational Cost
Simple MC	High (Baseline)	Slow	Low (Baseline)
Control Variates	90× Reduction	Fastest	Moderate
Importance Sampling	Lower than SMC, but higher bias	Fast	High