Data Generation Diagnostics

William Ruth

Introduction

The goal of this document is to check the validity of a Monte Carlo sample for my Trust Study Analysis. To this end, I check various conditional and marginal moments of the response variables by comparing them against group-wise and global sample means.

In each section, I will identify an expected value (conditional or marginal), and derive a theoretical expression for this expected value. I then identify an estimator of this expectation by averaging suitable quantities from a typical Trust Study dataset. We treat covariates as random, so typically we can treat these averages using standard approaches for iid samples. We then compare our estimator with the theoretical quantity to verify that our simulated data don't obviously violate any assumptions. We start with expressions for M, the mediatior, then move to Y, the outcome. This is natural, since the dependence of Y on M makes theoretical calculations a bit more intricate.

Many of the theoretical quantities can be expressed in terms of the so-called "logit-normal' ' integral. That is, the mean of a random variable whose logit is normally distributed. We open this document with a brief exploration of these logit-normal integrals and some corresponding notation. We then move on to the various means, their theoretical analysis, and comparison of theory with numerics.

The Logit-Normal Integral

Let X be normally distributed and $Y = expit(X) = exp(X)/(1 + exp(X)) = 1/(1 + exp(-X))$. In the analysis that follows, we will often encounter quantities of the form $\mathbb{E}Y$. There is no closed-form expression for this expected value, but we can write it as

$$\begin{aligned} \mathbb{E}Y &= \int_{-\infty}^{\infty} \frac{\phi(x; \mu, \sigma)}{1 + exp(-x)} dx \\ &= \int_{-\infty}^{\infty} \frac{\phi(z; 0, 1)}{1 + exp(-\mu - \sigma z)} dz \end{aligned}$$

This integral can be approximated using numerical quadrature. Specifically, we use the `integrate` function in `R`. Note: When implementing this expression, be sure to use `-Int` and `Int` as the range for `integrate`, as just choosing large finite values can cause inaccuracies. I have implemented this function in my `MultiMedUQ` package as `phi` in the file `Exact_Asymptotic_Helpers.r`.

Mediator

We start granular, conditioning on everything relevant to the mean of M, then gradually marginalize out these dependencies until we get an expression for the marginal mean of M.

First, we set some notation. Let X, Z be covariates, with $X \in \mathbb{R}^{n \times p}$ and $Z \in \mathbb{R}^{n \times q}$. We take all covariates' entries to be iid Bernoulli(0.5). Write $D = \{X, Z\}$ for the data. Next, let $\alpha \in \mathbb{R}^p$ be the vector of fixed effects and $U \sim N(0, \Sigma_M) \in \mathbb{R}^q$ be the random effects. Taken together, we write $g[\mathbb{E}(M|U, D)] = X^T\alpha + Z^TU$, where $g = \text{logit}$ is the link function. Note that we have specified this relationship for a single observation.

Since covariates are taken to be random, each observation within a group is iid, given the random effect for that group.

$\mathbb{E}(M|U,D)$

This quantity corresponds to the mean of a particular observed M. There's nothing really to average here, so we don't have an empirical reference to compare against our theory. Nevertheless, we will present the theory as a foundation for more interesting comparisons.

Applying the inverse link function to both sides, we get a simple expression for this section's conditional expectation:

$$\begin{aligned}\mathbb{E}(M|U,D) &= g^{-1}(X^T\alpha + Z^TU)\\ &=: g^{-1}[\eta(U,D)]\end{aligned}$$

where η is the linear predictor, $\eta(U,D) = X^T\alpha + Z^TU$.

$\mathbb{E}(M|U)$

This is where things start to get interesting. Marginalizing over the data while conditioning on the random effect corresponds to averaging over values of M within a single group.

Theory

This theoretical representation takes a bit more work. We will need to take expectation over all covariates. Fortunately, these covariates are all binary, so each only takes two possible values. We represent this averaging process by a new operator. I'm going to be a bit informal here to avoid having to be too pedantic about the presence of an intercept or possible overlap between X and Z in D.

First, let $\mathcal{C}$ be the corners of the unit cube with dimension equal to the number of unique random covariates in D. I.e. The set of all vectors of zeros and ones of the appropriate length. For some conformal function, f, define the operators T and $\bar{T}$ as

$$\begin{aligned}Tf &= \sum_{d\in\mathcal{C}} f(d)\\ \bar{T}f &= |\mathcal{C}|^{-1}Tf.\end{aligned}$$

When applying $\bar{T}$ to functions with multiple arguments, we will use subscripts to denote which variables are being summed over.

The operator $\bar{T}$ corresponds exactly to taking the expected value over D. The conditional expectation in this section can thus be easily represented using this operator.

$$\begin{aligned}\mathbb{E}(M|U) &= \mathbb{E}_D\mathbb{E}_M(M|U,D)\\ &= \bar{T}_D\mathbb{E}_M(M|U,D)\\ &= \bar{T}_D g^{-1}[\eta(U,D)]\end{aligned}$$

Numerics

We estimate the conditional expectation in this section by averaging over values of M within the same group. Note that, in order to compute the theoretical value of this conditional expectation, we need to know the value of the random effect, U. This is easily done by making a minor modification to my sampling function. In fact, I even included an argument to return each group's random effect.

```r
#! WARNING: The order of groups in data is not the same as the order of groups in all_REs

all_E_Ms = c()
all_M_bars = c()

for(i in seq_along(all_REs)){
    # if (i %% 50 == 0) print(paste0("i = ", i, " of ", length(all_REs)))

    # i=1
    this_group = all_REs[[i]][["data"]]
    # this_group = filter(data, group == paste0("G", i))
    this_REs = all_REs[[i]][["M"]]

    #* Compute conditional expectation of M given U

    data_all_combs = cbind(1, expand.grid(c(0,1), c(0,1), c(0,1)))
    lin_pred_all_combs = lin_preds_from_vecs(data_all_combs, b_M, this_REs, c(1,2))
    probs_all_combs = expit(lin_pred_all_combs)

    this_E_M = mean(probs_all_combs)
    all_E_Ms[i] = this_E_M

    #* Estimate conditional expectation of M given U using MC sample
    this_M_bar = mean(this_group$M)
    all_M_bars[i] = this_M_bar
    # this_M_SD = sd(this_group$M)
}

# Compare analytical and numerical values of E(M | U)
data_M_given_U = data.frame(theory = all_E_Ms, numerical = all_M_bars)
print(head(data_M_given_U))
```

```
##      theory numerical
## 1 0.6121352     0.587
## 2 0.6215573     0.643
## 3 0.5784946     0.574
## 4 0.3966781     0.401
## 5 0.4524328     0.446
## 6 0.4174304     0.452
```

```r
# Mean and SD of relative error (normalized by theoretical value)
rel_errs = abs(all_M_bars - all_E_Ms)/ all_E_Ms
print(mean(rel_errs))
```

```
## [1] 0.02722246
```

```r
print(sd(rel_errs))
```

```
## [1] 0.02159393
```

$\mathbb{E}(M)$

We now proceed to the marginal mean of M. This one is estimated by averaging the groups' means. Here we see our first example of a logit-normal integral.

Theory

We obtain this conditional expectation by taking the expected value with respect to U of $\mathbb{E}(M|U)$. To save me a bit of typing, I'm going to omit the bounds of integration. They're all over an appropriate dimensional real space.

$$\begin{aligned}
\mathbb{E}(M) &= \mathbb{E}_U \mathbb{E}_M(M|U) && (1)\\
&= \int \mathbb{E}_M(M|U=u)\phi(u;0,\Sigma_M) && (2)\\
&= \int \bar{T}_D g^{-1}[\eta(U,D)]\phi(u;0,\Sigma_M) && (3)\\
&= \bar{T}_D \int g^{-1}[\eta(U,D)]\phi(u;0,\Sigma_M) && (4)\\
&= \bar{T}_D \int g^{-1}[X^T\alpha + Z^T u]\phi(u;0,\Sigma_M) && (5)\\
&= \bar{T}_D \int g^{-1}[X^T\alpha + Z^T \Gamma_M z]\phi(z;0,I_q) && (6)\\
&= \bar{T}_D \varphi(X^T\alpha, Z^T\Sigma_M Z), && (7)
\end{aligned}$$

where Γ_M is any[1] square root of Σ_M (i.e. $\Gamma_M\Gamma_M^T = \Sigma_M$). While Equation (7) looks intimidating, in our numerical example there are only three random covariates, so the operator $\bar{T}_D$ only involves summing over $2^3 = 8$ terms. This does still require us to compute 8 logit-normal integrals, but that shouldn't be too hard.

Numerics

This section will require us to evaluate some logit-normal integrals. We will use the `phi` function I wrote earlier to do this. The function `phi` takes two arguments: `mu` and `sigma`, which correspond respectively to the mean and SD of the normal RV whose expit we are taking the expected value of. Usually, this will be $X^T\alpha$ and $\sqrt{Z^T\Sigma_M Z}$ respectively.

```r
# Theory
get_fixef_M <- function(b_M, x, c1, c2){
  output = b_M[1]
  output = output + x * b_M[2]
  output = output + c1 * b_M[3]
  output = output + c2 * b_M[4]

  return(as.numeric(output))
}

get_ranef_var_M <- function(Sigma_M, x){
  z = c(1, as.numeric(x))

  output = t(z) %*% Sigma_M %*% z
  return(as.numeric(output))
}

## The mean of M is an average of a bunch of logit-normal integrals, or terms. Compute the individual t
all_terms_M = c()
for(i in 1:nrow(data_all_combs)){
  this_data = data_all_combs[i,]
```

[1] We may prefer to require that Γ_M be symmetric and positive definite. There is only one such square root matrix for a given Σ_M.

```
  this_mean = get_fixef_M(b_M, this_data[2], this_data[3], this_data[4])
  this_var = get_ranef_var_M(Sigma_M, this_data[2])
  this_SD = sqrt(this_var)

  this_term = phi(this_mean, this_SD)
  all_terms_M = c(all_terms_M, this_term)
}

E_M = mean(all_terms_M)


# Estimate
M_bar = mean(all_M_bars)
```

```
## [1] "Theory: 0.494039764913277"

## [1] "Numerics:0.48903"

## [1] "Rel-Err: 0.0102442895390409"
```

Outcome-Y

We follow the same steps here as with M, starting with group-wise means and then proceeding to the global mean. The main difference now is that we must correctly account for the dependence of Y on M.

Adding a bit of additional notation, we now write β for the vector of fixed effects, and $V \sim N(0, \Sigma_Y) \in \mathbb{R}^{q_Y}$ for the random effects. We still use D to refer to the fixed and random effect covariates of M, but note that Y also depends on M.

$\mathbb{E}(Y|V, D, M)$

This quantity is the mean of a particular observed Y. As for M, its derivation is simply a matter of applying the inverse logit transformation to the linear predictor.

$$\begin{aligned} \mathbb{E}(Y|V, D, M) &= g^{-1}[\zeta(V, D, M)] \\ &=: s(V, D, M), \end{aligned}$$

where ζ is the linear predictor of Y, $\zeta(V, D, M) = D^T\beta_{-M} + M\beta_M + Z^T V_{-M} + MV_M$.

There isn't really anything to compare here, since we only observe a 1 or a 0 for each Y. I guess we could try grouping by covariates, but that doesn't sound worth the trouble.

$\mathbb{E}(Y|V)$

This quantity is notably more complicated to compute theoretically than the corresponding value for M. This is because the non-linear dependence of Y on M prevents us from hiding the dependence of M on its own random effects vector, U. We will need to do some logit-normal integrals.

Theory

We first observe that

$$\begin{aligned} \mathbb{E}(Y|V, D) &= \mathbb{E}_M[\mathbb{E}_Y(Y|V, D, M)|V, D] && (8) \\ &= s(V, D, 1)\mathbb{P}(M = 1|D) + s(V, D, 0)\mathbb{P}(M = 0|D) && (9) \\ &=: s(V, D, 1)p_{M|D} + s(V, D, 0)(1 - p_{M|D}). && (10) \end{aligned}$$

We already have an expression for $s(V, D, M)$. We now derive one for $p_{M|D}$. To this end, observe that it is easy to show that each term in the average given by Equation (7) is of the form $\mathbb{E}(M|D)$. We therefore directly use the argument to $\bar{T}_D$ in (7) as our expression for $\mathbb{P}(M = 1|D)$:

$$\mathbb{P}(M = 1|D) = \varphi(X^T\alpha, \sqrt{Z^T\Sigma_M Z}), \tag{11}$$

where $\varphi(\mu, \sigma)$ is the logit-normal integral with mean μ and SD σ.

We are now equipped to evaluate $\mathbb{E}(Y|V, D)$ using a single logit-normal integral. Passing to $\mathbb{E}(Y|V)$ is then simply a matter of applying the operator $\bar{T}_D$. That is, averaging over all combinations of covariate values in D. This requires a total of 8 logit-normal integrals for our example problem.

Putting everything together, we have

$$\mathbb{E}(Y|V) = \bar{T}_D[s(V, D, 1)\varphi(X^T\alpha, \sqrt{Z^T\Sigma_M Z}) + s(V, D, 0)(1 - \varphi(X^T\alpha, \sqrt{Z^T\Sigma_M Z}))]$$

Numerics

We recycle some computation from previous sections

```r
all_E_Ys = c()
all_Y_bars = c()

for(i in seq_along(all_REs)){
    # if (i %% 50 == 0) print(paste0("i = ", i, " of ", length(all_REs)))

    # i=1
    this_group = all_REs[[i]][["data"]]
    # this_group = filter(data, group == paste0("G", i))
    this_REs = all_REs[[i]][["Y"]]




    #* Compute conditional expectation of Y given V



    ## Expit of lin pred for Y at each combination of X, C1, C2 and M. Grouped by M


    ### X, C1 and C2
    covariate_grid = expand.grid(c(0,1), c(0,1), c(0,1))
    X_vals = covariate_grid[,1]
    C1_vals = covariate_grid[,2]
    C2_vals = covariate_grid[,3]


    ### M=1 group

    #### Add M=1
    data_combs_M1 = cbind(1, X_vals, 1, C1_vals, C2_vals)
    #? Order on REs is intercept, X, M
    lin_preds_M1 = lin_preds_from_vecs(data_combs_M1, b_Y, this_REs, 1:3)
    probs_M1 = expit(lin_preds_M1)
    ### M=0 group
    data_combs_M0 = cbind(1, X_vals, 0, C1_vals, C2_vals)
    lin_preds_M0 = lin_preds_from_vecs(data_combs_M0, b_Y, this_REs, 1:3)
    probs_M0 = expit(lin_preds_M0)






    ## Probabilities for M
    all_M_probs = all_terms_M
    all_M_inv_probs = 1 - all_M_probs



    ## Compute each E(Y | V, X, C1, C2)
    terms_M1 = probs_M1 * all_M_probs
    terms_M0 = probs_M0 * all_M_inv_probs
    E_Y_given_V_and_covs = terms_M1 + terms_M0

    ## Average over covariates
    E_Y_given_V = mean(E_Y_given_V_and_covs)

    all_E_Ys[i] = E_Y_given_V

    #* Estimate conditional expectation of Y given V using MC sample
    this_Y_bar = mean(this_group$Y)
    all_Y_bars[i] = this_Y_bar
    # this_M_SD = sd(this_group$M)
}

# Compare analytical and numerical values of E(M | U)
data_Y_given_V = data.frame(theory = all_E_Ys, numerical = all_Y_bars)
print(head(data_M_given_U))
```

```
##      theory numerical
## 1 0.6121352     0.587
## 2 0.6215573     0.643
## 3 0.5784946     0.574
## 4 0.3966781     0.401
## 5 0.4524328     0.446
## 6 0.4174304     0.452
```

```
# Mean and SD of relative error (normalized by theoretical value)
rel_errs = abs(all_Y_bars - all_E_Ys)/ all_E_Ys
print(mean(rel_errs))
```

```
## [1] 0.06978162
```

```
print(sd(rel_errs))
```

```
## [1] 0.05472759
```

While these relative errors are a bit high, we will see in the next section that the global mean of Y (i.e. the mean of the group means) is estimated with much lower error.

$\mathbb{E}Y$

This quantity is a straightforward extension of the previous section. We simply take the expectation over V, the random effect for Y, of each term in the sum used to define $\mathbb{E}(Y|V)$. Note that each such term requires two logit-normal integrals, one for each value of M, for a total of $2 \cdot 8 = 16$ logit-normal integrals.

Theory

Elaborating on the above paragraph, recall that the function $s(V, D, M)$ in Equation (10) is the inverse logit of a linear predictor in Y. The V-expectation of such an s can therefore be computed using our φ function, with mean equal to the fixed effects component of the linear predictor and variance equal to the random effects variance.

More formally, we recall that

$$\mathbb{E}(Y) = \mathbb{E}_V \mathbb{E}_Y(Y|V) \tag{12}$$

$$= \int \mathbb{E}_Y(Y|V = v)\phi(v; 0, \sqrt{\Sigma_Y}) \tag{13}$$

$$= \int \bar{T}_D[s(V, D, 1)p_{M|D} + s(V, D, 0)(1 - p_{M|D}]\phi(v; 0, \sqrt{\Sigma_Y}) \tag{14}$$

$$= \bar{T}_D \left[\varphi(X_M(1)^T\beta, \sqrt{Z_M(1)^T \Sigma_Y Z_M(1)})p_{M|D} + \varphi(X^T\beta, \sqrt{Z_M(0)^T \Sigma_Y Z_M(0)})(1 - p_{M|D})\right], \tag{15}$$

where X_M and Z_M are the vectors of fixed and random effects covariates respectively, augmented to include M (the order is intercept, X, M, $C1$, $C2$, with the latter two dropped for Z_M). The value of M is indicated by the argument to X_M and Z_M. See Equation (10) for the definition of $p_{M|D}$.

```
get_fixef_Y <- function(b_Y, x, m, c1, c2){
  output = b_Y[1]
  output = output + x * b_Y[2]
  output = output + m * b_Y[3]
  output = output + c1 * b_Y[4]
  output = output + c2 * b_Y[5]

  return(as.numeric(output))
}

get_ranef_var_Y <- function(Sigma_Y, x, m){
  z = c(1, as.numeric(x), as.numeric(m))

  output = t(z) %*% Sigma_Y %*% z
  return(as.numeric(output))
}

this_phi_Y <- function(b_Y, Sigma_Y, x, m, c1, c2){
  this_mu = get_fixef_Y(b_Y, x, m, c1, c2)

  this_var = get_ranef_var_Y(Sigma_Y, x, m)
  this_SD = sqrt(this_var)

  this_term = phi(this_mu, this_SD)
}
```

Numerics

Conveniently, we can repeat the computation of the $\mathbb{E}(Y|V)$ terms from the previous section, with each s replaced by its V-expectation.

The global mean of Y is estimated by averaging the group means.

```r
# Compute global mean of Y

## X, C1 and C2
covariate_grid = expand.grid(c(0,1), c(0,1), c(0,1))


## Compute logit-normal integrals
all_M1_phis = c()
all_M0_phis = c()
for(i in 1:nrow(data_all_combs)){
  this_data = covariate_grid[i,]
  this_X = this_data[1]
  this_C1 = this_data[2]
  this_C2 = this_data[3]

  this_M1_phi = this_phi_Y(b_Y, Sigma_Y, this_X, 1, this_C1, this_C2)
  this_M0_phi = this_phi_Y(b_Y, Sigma_Y, this_X, 0, this_C1, this_C2)

  all_M1_phis = c(all_M1_phis, this_M1_phi)
  all_M0_phis = c(all_M0_phis, this_M0_phi)
}

## Probabilities for M
all_M_probs = all_terms_M
all_M_inv_probs = 1 - all_M_probs


## Compute terms in sum
### Note: Each one is E(Y | D) = E_V E(Y|V, D)
all_terms_Y = all_M1_phis * all_M_probs + all_M0_phis * all_M_inv_probs


E_Y = mean(all_terms_Y)




# Estimate mean of Y using MC sample
Y_bar = mean(all_Y_bars)
SE_Y_bar_ish = sd(all_Y_bars) / sqrt(length(all_Y_bars))




# Compare analytical and numerical values of E(Y)
print(paste0("Theory: ", E_Y))
```

```
## [1] "Theory: 0.48951267623642"
```

```r
print(paste0("Numerics:", Y_bar))
```

```
## [1] "Numerics:0.486735"
```

```
print(paste0("Rel-Err: ", abs(E_Y - Y_bar)/Y_bar))
```

```
## [1] "Rel-Err: 0.00570675261984505"
```