Monte Carlo Integration: A More General Example

Ashley I. Naimi, PhD

This function can be used to implement Monte Carlo integration for computing the true value of the average treatment effect, the effect of treatment on the treated, and the effect of treatment on the untreated, all on the odds ratio scale.

```r
## define the expit function, whihc is
## equivalient to plogis in base R
expit<-function(a){1/(1+exp(-a))}

# set a single seed for reproducibility
set.seed(123)

# initialize the function. In this case, we're calling it
# the collapsibiliity function, due to copy-paste
# a different name would have been more suitable
collapsibility_function <- function(sample_size, intercept){

  #define the sample size, large enough to minimize
  #monte carlo error
  n = sample_size

  #simulate a single binary confounder
  C <- rbinom(n, size = 1,p = .5)

  # simulate the propensity score
  theta <- c(0,log(2))
  pi_a <- expit(theta[1]+theta[2]*C) ## note the correction!

  # simulate the observed exposure
  A_obs <- rbinom(n, size = 1, p = pi_a)

  # define the outcome model
  beta <- c(intercept, log(1.5), log(.5), log(3))
  mu <- expit(beta[1] + beta[2]*A_obs + beta[3]*C + beta[4]*A_obs*C)

  # compute the expected values from the outcome model defined above,
  # but only among those who were exposed
  # Note that for the mean unexposed outcome among the exposed, we need to
  # change the `A_obs` value in the model to 0, and subset the `C` values to be among
  # those with A_obs = 1
  mu1_A1 <- mean(mu[A_obs==1])
```

```
  mu0_A1 <- mean(expit(beta[1] + beta[2]*0 + beta[3]*C[A_obs==1] + beta[4]*0*C[A_obs==1]))

  # compute the expected values from the outcome model defined above,
  # but only among those who were UNexposed
  # Note that for the mean exposed outcome among the unexposed, we need to
  # change the `A_obs` value in the model to 1, and subset the `C` values to be among
  # those with A_obs = 0
  mu1_A0 <- mean(expit(beta[1] + beta[2]*1 + beta[3]*C[A_obs==0] + beta[4]*1*C[A_obs==0]))
  mu0_A0 <- mean(mu[A_obs==0])

  # construct means needed for ATE
  mu1 <- mean(expit(beta[1] + beta[2]*1 + beta[3]*C + beta[4]*1*C))
  mu0 <- mean(expit(beta[1] + beta[2]*0 + beta[3]*C + beta[4]*0*C))

  ## compute the odds ratios
  OR_A1 <- (mu1_A1/(1-mu1_A1))/(mu0_A1/(1-mu0_A1))
  OR_A0 <- (mu1_A0/(1-mu1_A0))/(mu0_A0/(1-mu0_A0))
  OR <- (mu1/(1-mu1))/(mu0/(1-mu0))

  ## output the marginally adjusted odds
  return(c(mean(mu),OR_A1, OR_A0, OR))
}


collapsibility_function(sample_size = 1e6, intercept = -2.5)

## [1] 0.1048714 2.7001949 2.2509481 2.5005998

collapsibility_function(sample_size = 5e6, intercept = -2.5)

## [1] 0.1049293 2.7048736 2.2500201 2.5026931

collapsibility_function(sample_size = 1e7, intercept = -2.5)

## [1] 0.1048962 2.7037424 2.2514794 2.5027044

collapsibility_function(sample_size = 1e8, intercept = -2.5)

## [1] 0.1049093 2.7042860 2.2510773 2.5028353
```