Average Effects Based on Regressions with a Logarithmic Link Function: A New Approach with Stochastic Covariates

Supplementary Material A

.

Average Effects Based on Regressions with a Logarithmic Link Function: A New Approach with Stochastic Covariates

Simulation details

In this section we provide additional information on the simulation study. First, a detailed description of the data generation process is given. Second, we state the explicit moment-based estimators for each unconditional distribution of Z used in the simulation. Third, we provide the listings for the data generation procedure as well as for the estimators.

Data simulation

As the population model, we choose a regression with log link according to Equation (9) and conditional NB distributions for Y. The model includes a dichotomous treatment variable X, a covariate Z and their interaction XZ.

In order to generate scenarios that are comparable across different distributions of Z, we computed the regression coefficients $\beta_{00}, \beta_{01}, \beta_{11}$ from their standardized versions $\beta^*_{00}, \beta^*_{01}, \beta^*_{11}$ (i.e., with regard to a z-transformed covariate $Z^* = (Z - E(Z))/\sqrt{\text{Var}(Z)}$). **These are derived as follows:**

$$
\begin{aligned}
\log[E(Y|X, Z^*)] = \beta^*_{00} + \beta^*_{10}X + \beta_{01}Z^* + \beta^*_{11}XZ^* \\
&= \beta^*_{00} + \beta^*_{10}X + \beta_{01}\left(\frac{Z - E(Z)}{\sqrt{Var(Z)}}\right) + \beta^*_{11}X\left(\frac{Z - E(Z)}{\sqrt{Var(Z)}}\right) \\
&= \left(\beta^*_{00} - \frac{\beta^*_{01}}{\sqrt{Var(Z)}}E(Z)\right) + \left(\beta^*_{10} - \frac{\beta^*_{11}}{\sqrt{Var(Z)}}E(Z)\right)X \\
&\quad + \left(\frac{\beta^*_{01}}{\sqrt{Var(Z)}}\right)Z + \left(\frac{\beta^*_{11}}{\sqrt{Var(Z)}}\right)XZ
\end{aligned}
$$

Starting from the regression on X and the standardized Z^*, we derived the

corresponding regression coefficients for the unstandardized Z, which are

$$\beta_{00} = \beta^*_{00} - \frac{\beta^*_{01}}{\sqrt{\mathrm{Var}(Z)}} E(Z), \tag{1}$$

$$\beta_{01} = \frac{\beta^*_{01}}{\sqrt{\mathrm{Var}(Z)}}, \tag{2}$$

$$\beta_{11} = \frac{\beta^*_{11}}{\sqrt{\mathrm{Var}(Z)}} \tag{3}$$

The standardized regression coefficients are divided by the standard deviation of Z, because the covariates are on a different scale depending on their distribution in the simulation. We manipulated the effect size of the ATE, and thus the remaining regression coefficient β_{10} was not a free parameter but determined by

$$\beta_{10} = \log\left(\frac{\Delta \cdot \sqrt{\mathrm{Var}(Y|X=0)} + M_Z(\beta_{01})\exp(\beta_{00})}{\exp(\beta_{00}) M_Z(\beta_{01} + \beta_{11})}\right)$$

The standardized regression coefficients were chosen $\beta^*_{00} = 0$ and $\beta^*_{01} = 0.5$ to simulate a moderate covariate effect in the control group $X = 0$. The standardized interaction parameter was varied from no interaction ($\beta^*_{11} = 0$) to strong interaction ($\beta^*_{11} = 1$) with a step width of 0.25. As the derivations above suggest, the difference between the fixed-covariate and stochastic-covariate-based standard errors should increase with the strength of the interaction parameter. The population distributions of Z were chosen to resemble distinct distributional shapes: a standard normal distribution $\mathcal{N}(0, 1)$ for a symmetric shape with zero skewness and kurtosis, a Poisson distribution $\mathcal{P}(1)$ for a positively skewed shape, and a uniform distribution $U(-\sqrt{3}, \sqrt{3})$ for a flat shape with negative kurtosis. The parameters of $\mathcal{N}(0, 1)$ and $U(-\sqrt{3}, \sqrt{3})$ represent standardized variables, that is, $E(Z) = 0,\ \mathrm{Var}(Z) = 1$. For the Poisson distribution, however, the parameter $\lambda = 1$ was chosen to create a moderate skew. We further varied the effect size Δ (Glass, 1976) of the ATE and the sample size N. In total, we evaluated 225 experimental scenarios (see Table 3 for more details on the design parameters).

The simulation consisted of three steps, which were carried out using the statistical software R (R Core Team, 2018). In the first step, N covariate values z_i were generated by drawing from the populational distribution of Z. In the second step, the values of the dichotomous treatment x_i were drawn from a Binomial distribution $\mathrm{B}(1, p)$ where

$$p = P(X|Z^*) = 1 - \frac{1}{1 + \exp(Z^*)} \tag{4}$$

which was generated as a logistic function with standardized Z^* (i.e., the higher the value of Z, the higher the probability of getting into the treatment group). This means that we simulated a quasi-experimental setting, where treatment assignment is not random but depending on Z. As mentioned above, ATE and UTE are not equivalent in this case, and consideration of the covariate Z is necessary to estimate the average (causal) effect. In the third step, the conditional expectations $E(Y|X = x_i, Z = z_i)$ were computed and the observed y_i were drawn from a negative binomial distribution. The overdispersion parameter was $\alpha = 1$. Each experimental condition was replicated $R = 5,000$ times. For more details and the corresponding R code, see the supplementary material.

Moment-based estimators by distribution

For normal distributed $Z \sim \mathcal{N}(\hat{\mu}, \hat{\sigma})$ with $\hat{\mu} = \bar{z}$ and $\hat{\sigma} = \hat{\sigma}_z$

$$\widehat{\mathrm{ATE}}_{\mathrm{MOM}}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}) = \exp(\hat{\beta}_{00} + \hat{\beta}_{10}) \exp\left(\hat{\mu}(\hat{\beta}_{01} + \hat{\beta}_{11}) + \frac{\hat{\sigma}^2(\hat{\beta}_{01} + \hat{\beta}_{11})^2}{2}\right) - \exp(\hat{\beta}_{00}) \exp\left(\hat{\mu}\hat{\beta}_{01} + \frac{\hat{\sigma}^2\hat{\beta}_{01}^2}{2}\right) \tag{5}$$

For Poisson distributed $Z \sim \mathcal{P}(\hat{\lambda})$ where $\hat{\lambda} = \bar{z}$

$$\widehat{\mathrm{ATE}}_{\mathrm{MOM}}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}) = \exp(\hat{\beta}_{00} + \hat{\beta}_{10}) \exp(\hat{\lambda}(\exp(\hat{\beta}_{01} + \hat{\beta}_{11}) - 1)) - \exp(\hat{\beta_{00}}) \exp(\hat{\lambda}(\exp(\hat{\beta_{01}}) - 1)) \tag{6}$$

For Uniform distributed $Z \sim \mathrm{U}(\hat{a}, \hat{b})$ where $\hat{a} = \bar{z} - \sqrt{3}\hat{\sigma}_z$ and $\hat{b} = \bar{z} + \sqrt{3}\hat{\sigma}_z$

$$\widehat{\mathrm{ATE}}_{\mathrm{MOM}}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}) = \exp(\hat{\beta}_{00} + \hat{\beta}_{10}) \frac{e^{(\hat{\beta}_{01}+\hat{\beta}_{11})\hat{b}} - e^{(\hat{\beta}_{01}+\hat{\beta}_{11})\hat{a}}}{(\hat{\beta}_{01} + \hat{\beta}_{11})(\hat{b} - \hat{a})} - \exp(\hat{\beta}_{00}) \frac{e^{\hat{\beta}_{01}\hat{b}} - e^{\hat{\beta}_{01}\hat{a}}}{\hat{\beta}_{01}(\hat{b} - \hat{a})} \tag{7}$$

Estimation of sample moments for moment-based approach

In our simulation, we used a two-step procedure for obtaining sample moment estimates and their covariance. The first step was based on a decomposition of the unconditional expectation and variance of Z with respect to the treatment groups:

$$E(Z) = E(Z|X = 1) \cdot \mathrm{P}(X = 1) + E(Z|X = 0) \cdot \mathrm{P}(X = 0) \tag{8}$$

$$\mathrm{Var}(Z) = \mathrm{Var}(Z|X = 1) \cdot \mathrm{P}(X = 1) + \mathrm{Var}(Z|X = 0) \cdot \mathrm{P}(X = 0) \tag{9}$$

$$+ (E(Z|X = 1) - E(Z))^2 \cdot \mathrm{P}(X = 1) + (E(Z|X = 0) - E(Z))^2 \cdot \mathrm{P}(X = 0) \tag{10}$$

We used a multigroup structural equation model with stochastic group sizes to estimate the $X = x$-conditional expectations and variances, and the probabilities $\mathrm{P}(X = 0)$ and $\mathrm{P}(X = 1)$.

$$\bar{z}_0 = \hat{E}(Z|X = 0) \tag{11}$$

$$\bar{z}_1 = \hat{E}(Z|X = 1) \tag{12}$$

$$s_0^2 = \widehat{\mathrm{Var}}(Z|X = 0) \tag{13}$$

$$s_1^2 = \widehat{\mathrm{Var}}(Z|X = 1) \tag{14}$$

$$\mathrm{p}_0 = \hat{\mathrm{P}}(X = 0) \tag{15}$$

$$\mathrm{p}_1 = \hat{\mathrm{P}}(X = 1) \tag{16}$$

The sample mean $\bar{z}$ and variance s^2 were then obtained by

$$\bar{z} = \bar{z}_0 \cdot \mathrm{p}_0 + \bar{z}_1 \cdot \mathrm{p}_1 \tag{17}$$

$$s^2 = s_0^2 \cdot \mathrm{p}_0 + s_1^2 \cdot \mathrm{p}_1 + (\bar{z}_0 - \bar{z})^2 \cdot \mathrm{p}_0 + (\bar{z}_1 - \bar{z})^2 \cdot \mathrm{p}_1 \tag{18}$$

This approach allows us to model the group sizes as random variables. Thus, variability in sample moment estimates due to different group weights are accounted for. For more information on the underlying model and assumptions, see Mayer, Dietzfelbinger, Rosseel, and Steyer (2016, p. 4).

In the second step, we used the estimated sample moments $\bar{z}$ and s^2 to compute the estimated parameters of the distribution. Depending on the distribution this was

Normal $\mathcal{N}(\mu, \sigma)$:	$\hat{\mu} = \bar{z}$,	$\hat{\sigma}^2 = s^2$
Poisson $\mathcal{P}(\lambda)$:	$\hat{\lambda} = \bar{z}$	
Uniform $\mathrm{U}(a, b)$	$\hat{a} = \bar{z} - \sqrt{3s^2}$,	$\hat{b} = \bar{z} + \sqrt{3s^2}$

For the three distributions in our simulation study, this procedure provided unbiased and efficient estimates for the parameters of the distribution. We especially compared the two-step procedure to conventional maximum likelihood estimates. For the uniformly distributed Z, the MLE provided biased estimates in small samples, while the two step-procedure was unbiased. For normally or Poisson distributed Z, point estimates from MLE and the two-step procedure were identical, however, the MLE was a little less efficient.

Data generation function

```
data.sim <- function(N, cohensd, b101, distribution){
  # Standardized Regression Coefficients in Control Group
  b000 <- 0      # Intercept
  b001 <- 0.5    # Slope

  # Remaining parameters are set depending on the covariate's distribution
  if (distribution == "normal"){
    muZ <- 0    # Mean of Z
    sigmaZ <- 1    # Variance of Z
    a000 <- b000-b001*muZ/sqrt(sigmaZ)   # Unstandardized Intercept in Control Group
    a001 <- b001/sqrt(sigmaZ)   # Unstandardized Slope in Control Group
    a101 <- b101/sqrt(sigmaZ)   # Unstandardized Interaction Coefficient
    varY0 <- 1.089     # Var(Y|X=0)
    a100 <- log(sqrt(varY0)*cohensd + exp(.5*b001^2)) - .5*(b001 + b101)^2   # Unstandardized Coefficient of Treatment Variable
    z <- rnorm(N, muZ, sigmaZ)   # Z from a Standard Normal Distribution
  } else if (distribution == "poisson"){
    lambdaZ = 1    # Mean and Variance of Z
    varY0 <- 1.06    # Var(Y|X=0)
    a000 <- b000-b001*lambdaZ/sqrt(lambdaZ)   # Unstandardized Intercept in Control Group
    a001 <- b001/sqrt(lambdaZ)   # Unstandardized Slope in Control Group
    a101 <- b101/sqrt(lambdaZ)   # Unstandardized Interaction Coefficient
    a100 <- log(sqrt(varY0)*cohensd + exp(a000)*exp(lambdaZ*(exp(a001) - 1))) - log(exp(a000)*exp(lambdaZ*(exp(a001+a101)-1)))   # Unstandardized Coefficient of Treatment Variable
    z <- rpois(N, lambdaZ)   # Z from a Poisson distribution
  } else if (distribution == "uniform"){
    aZ <- -sqrt(3)    # lower limit of Z
    bZ <- sqrt(3)    # upper limit of Z
    muZ <- (sqrt(3)-sqrt(3))/2   # Mean of Z
    sigmaZ <- sqrt((sqrt(3)+sqrt(3))^2/12)   # Variance of Z
    a000 <- b000-b001*muZ/sqrt(sigmaZ)   # Unstandardized Intercept in Control Group
    a001 <- b001/sqrt(sigmaZ)   # Unstandardized Slope in Control Group
    a101 <- b101/sqrt(sigmaZ)   # Unstandardized Interaction Coefficient
    varY0 <- 1.09      # Var(Y|X=0)
    a100 <- log(sqrt(varY0)*cohensd + exp(b000)*(exp(b001*bZ) - exp(b001*aZ))/(b001*(bZ-aZ))) -
    log(exp(b000)*(exp((b001+b101)*bZ)-exp((b001+b101)*aZ))/((b001+b101)*(bZ-aZ)))   # Unstandardized Coefficient of Treatment Variable
    z <- runif(N, aZ, bZ)   # Z from a Uniform distribution
}
  zS <- scale(z)      # z-transformation of Z for generation of X
  x <- rbinom(N, 1, plogis(zS))   # Treatment with treatment probability as logistic function of Z
  muY <- exp(a000 + a001*z + a100*x + a101*x*z)   # Conditional Expectation of Y depending on Z and X
  y <- rnbinom(N, mu = muY, size =1)   # Y from negative binomial distribution

  d <- data.frame(y,x,z)
  return(d)
}
```

Estimator functions

```
library(lavaan)
library(car)

######## function estimating TEMZ #########
computeTemz <- function(m1, d){
  # Get regression  coefficients
  coefs <- coef(m1)
  b00 <- coefs[1]
  b10 <- coefs[2]
  b01 <- coefs[3]
  b11 <- coefs[4]

  # Get mean of covariate z
  zM <- mean(d$z)

  # Compute and return TEMZ
  temz <- exp(b00+b10+b01*zM+b11*zM) - exp(b00 + b01*zM)
  return(temz)
}


####### function estimating ATE (empirical approach, fixed covariate) ############
computeAveEmp <- function(m1, d){

  # Compute all conditional effects
  newdata0 <- newdata1 <- d
  newdata0$x <- 0
  newdata1$x <- 1

  tau0 <- predict(m1, newdata=newdata0, type="response") ## tau0 is almost constant
  tau1 <- predict(m1, newdata=newdata1, type="response") ## more variation in tau1
  delta10 <- tau1 - tau0

  # Compute VCOV of all conditional effects
  vcovs <- vcov(m1)
  mat <- cbind(delta10, tau1, d$z*delta10, d$z*tau1)
  vcov_delta10 <- mat %*% vcovs %*% t(mat)

  # Compute standard error for ATE
  mat <- rbind(1/nrow(d),
  as.numeric(d$x==0)/sum(d$x==0),
  as.numeric(d$x==1)/sum(d$x))
  vcov_eff <- mat %*% vcov_delta10 %*% t(mat)
  se_Ave <- sqrt(vcov_eff[1,1])

  # Return results
  res <- list(Ave=mean(delta10),
  se_Ave=se_Ave)

  return(res)
}

```

```
####### function estimating ATE (moment-based approach) #############
computeAveMom <- function(m1, d, distribution, se = "stochastic"){
  # Get regression coefficients and there covariance
  coefs <- coef(m1)
  vcovs <- vcov(m1)

  pnames <- c("g000", "g100", "g001", "g101")
  names(coefs) <- row.names(vcovs) <- colnames(vcovs) <- pnames

  # lavaan model for joint estimation of sample moments
  modelz <- '
    z ~ c(meanz001,meanz101)*1
    z ~~ c(varz001,varz101)*z
    group % c(groupw0,groupw1)*w
    gw0 := groupw0
    gw1 := groupw1
    mz001 := meanz001
    mz101 := meanz101
    vz001 := varz001
    vz101 := varz101
    N := exp(gw0) + exp(gw1)
     relfreq0 := exp(gw0)/N
     relfreq1 := exp(gw1)/N
    Ez1 := mz001*relfreq0 + mz101*relfreq1
    Vz1 := vz001*relfreq0 + vz101*relfreq1 + relfreq0*(mz001-Ez1)^2 + relfreq1*(mz101-Ez1)^2
    Px0 := relfreq0
    Px1 := relfreq1
    Pk0gx0 := relfreq0/Px0
    Pk0gx1 := relfreq1/Px1
    Ez1gx0 := mz001*Pk0gx0
    Ez1gx1 := mz101*Pk0gx1
    Vz1gx0 := vz001*Pk0gx0
    Vz1gx1 := vz101*Pk0gx1
  '
  mz <- sem(modelz, data=d, group="x", group.label=c(0,1), group.w.free=TRUE)

  ## augment coefs and vcovs
  acoefs <- c(coefs, coef(mz, type="user")[-c(1:6)])

  if (se == "stochastic"){
    # for stochastic covariate
    avcovs <- lav_matrix_bdiag(vcovs, lavInspect(mz, "vcov.def", add.class = FALSE))
    } else if (se == "fixed"){
    # for fixed covariate: covariance of distribution parameters fixed to zero
    avcovs <- lav_matrix_bdiag(vcovs, lavInspect(mz, "vcov.def", add.class = FALSE)*0)
  }

  row.names(avcovs) <- colnames(avcovs) <- names(acoefs)

  if (distribution == "normal"){
    Eg1 <- deltaMethod(acoefs,
        "exp(g000+g100)*exp((g001+g101)*Ez1+((g001+g101)^2*Vz1/2))-exp(g000)*exp(g001*Ez1+(g001^2*Vz1/2))",
        avcovs, func="Eg1")
  } else if (distribution == "uniform"){
```

```
    Eg1 <- deltaMethod(acoefs,
    "exp(g000+g100)*(exp((g001+g101)*(Ez1+sqrt(3*Vz1)))-exp((g001+g101)*(Ez1-sqrt(3*Vz1))))/((g001+g101)*(2*sc
    avcovs, func="Eg1")
  } else if ( distribution == "poisson"){
    Eg1 <- deltaMethod(acoefs,
         "exp(g000+g100)*exp(Ez1*(exp(g001+g101)-1))-exp(g000)*exp(Ez1*(exp(g001)-1))", avcovs,
         func="Eg1")
  } else if ( distribution == "chisquare"){
    Eg1 <- deltaMethod(acoefs,
         "(exp(g000+g100)/(1-2*(g001+g101))^(Ez1/2))-(exp(g000)/(1-2*g001)^(Ez1/2))", avcovs,
         func="Eg1")
  }

  res <- list(Ave=Eg1$Estimate,
  se_Ave=Eg1$SE)

  return(res)

  }
```

Simulation procedure

```
library(doParallel)
library(foreach)
library(MASS)
source("dataSimulation.R")
source("estimators.R")

# Set design parameters
sampleN <- c(50, 100, 250, 500, 1000)
interact <- c(0, 0.25, 0.5, 0.75, 1)
cohensd <- c(-.5, 0, .5)
distribution <- c("normal", "poisson", "uniform")
reps <- 5000
conditions <- expand.grid(sampleN, interact, cohensd, distribution)
colnames(conditions) <- c("sampleN", "interact", "cohensd", "distribution")

### Simulation procedure
#setup parallel backend to use 4 processors
cluster<-makeCluster(4, outfile = "debug.txt")
registerDoParallel(cluster)
results <- list()
for (i in 1:nrow(conditions)){
  # Parallel replications within each scenario
  res <- foreach::foreach(icount(reps), .combine = rbind, .packages = c("car", "lavaan","MASS"),
      .errorhandling = "remove") %dopar% {
    # Generate data set
    d <- data.sim(conditions$sampleN[i],
          conditions$cohensd[i],
          conditions$interact[i],
          conditions$distribution[i])
    # Estimate NB-GLM
    m1 <- glm.nb(y ~ x*z, data=d)
    # Get results from ATE estimators
    tmp <- c(unlist(computeAveEmp(m1, d, se = "fixed")),
        unlist(computeAveAnaStoch(m1, d, distribution = conditions$distribution[i])),
        unlist(computeAveAnaFixed(m1, d, distribution = conditions$distribution[i])),
        unlist(computeAveAnaStoch(m1, d, distribution = "normal")),
        computeTemz(m1, d),
        m1$converged)
    tmp
  }
  # Combine and save results
  results <- data.frame(res,
            conditions$sampleN[i],
            conditions$cohensd[i],
            conditions$interact[i],
            conditions$distribution[i])
            names(results) <- c("aveEmpFixed", "seEmpFixed",
            "aveAnaStochastic", "seAnaStochastic",
            "aveAnaFixed", "seAnaFixed",
            "aveStochNorm", "seStochNorm",
            "TEMZ",
            "conv", "N", "cohensd", "interact", "distribution")
```

```
  saveRDS(results, file = paste0("simulation/res_",
                         conditions$sampleN[i],"_",
                         conditions$cohensd[i],"_",
                         conditions$interact[i],"_",
                         conditions$distribution[i],
                         ".rds"))
}
stopCluster(cluster)
```

CountEffects package

As a supplement to this article, we provide an R package (R Core Team, 2018) called CountEffects, which comprises the functions related to the moment-based approach presented in this article.

It is available from GitHub and can be installed as follows:

```
library(devtools)
install_github('chkiefer/CountEffects')
```

References

Glass, G. V. (1976). Primary, secondary, and meta-analysis of research. *Educational Researcher*, *5*(10), 3-8. doi: 10.2307/1174772

Mayer, A., Dietzfelbinger, L., Rosseel, Y., & Steyer, R. (2016). The EffectLiteR approach for analyzing average and conditional effects. *Multivariate Behavioral Research*, *51*, 374–391. doi: http://dx.doi.org/10.1080/00273171.2016.1151334

R Core Team. (2018). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `http://www.R-project.org/`