Derivative Method for Mediation Analysis for Count Data (Poisson Regression)

Disclaimer:

This pdf is part of a YouTube tutorial (https://youtu.be/l3igREAFRXE).

1. Please make sure to read the published papers mentioned in the video description. Since I am not the author of the papers, I cannot guarantee that the R code and information in the tutorial correctly reflect the idea of those papers. Please read the papers by yourself and make your own judgment.
2. This video tutorial and this pdf are not peer-reviewed and the correctness and quality of the information and R code presented in this tutorial are not guaranteed. Thus, please do NOT cite the video tutorial or this pdf as a reference, if you are writing an academic paper.
3. This pdf is for your personal usage only. Please do NOT distribute this pdf.

Generate Sample Data

```
# set the size of the sample
n=500
# set seed
set.seed(123)

# simulate x (normal distribution)
X <- rep(0:1, each=n/2)
#print(X)

# simulate x (normal distribution)
#X <- rnorm(n, 5, 4)
# calculate the mean of X
#mean_x=mean(X)

# simulate a residual for M
residual_1<-rnorm(n,0,1)
M<-0.3+0.5*X+residual_1

# mu for Poisson regression via a log link
mu_1 <- exp(0.2 + 0.2*M+0.08*X)
# use rpois to generate Y
Y <- rpois(n, lambda=mu_1)

# combine into a dataframe and print out the first 6 rows
data <- data.frame(X=X, M=M, Y=Y)
head(data)
```

```
##   X          M Y
## 1 0 -0.26047565 0
## 2 0  0.06982251 1
## 3 0  1.85870831 0
## 4 0  0.37050839 2
## 5 0  0.42928774 2
## 6 0  2.01506499 2
```

Key Function

```
Mediation_function_count_method1<-function(data_used,i,x_predetermined)
{
  # Sample a data
  data_temp=data_used[i,]

  # a path
  result_a<-lm(M~X, data = data_temp)
  a_0<-result_a$coefficients[1]
  a_1<-result_a$coefficients[2]

    # b path
  result_b<-glm(Y~M+X, data = data_temp, family=poisson(link = "log"))
  b_0<-result_b$coefficients[1]
  b_1<-result_b$coefficients[2]
  c_1_apostrophe<-result_b$coefficients[3]

   #calculating the indirect effect
  M_estimated=a_0+a_1*x_predetermined
  indirect_effect<-a_1*b_1*exp(b_0+b_1*M_estimated+c_1_apostrophe*x_predetermined)
  return(indirect_effect)
}
```

Apply Function

Control Condition:

set x_predetermined=0

```
# use boot() to do bootstrapping mediation analysis
library(boot)
```

```
## Warning: package 'boot' was built under R version 4.1.3
```

```
boot_mediation <- boot(data, Mediation_function_count_method1, R=1000, x_predetermined=0)
boot_mediation
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data, statistic = Mediation_function_count_method1,
##     R = 1000, x_predetermined = 0)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.1684014 -0.001835435  0.03461317
```

Treatment Condition:

set x_predetermined=1

```
# use boot() to do bootstrapping mediation analysis
library(boot)
boot_mediation <- boot(data, Mediation_function_count_method1, R=1000, x_predetermined=1)
boot_mediation
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data, statistic = Mediation_function_count_method1,
##     R = 1000, x_predetermined = 1)
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* 0.2166155 0.001274214  0.04507241
```