Obtain variance for difference-in-difference from a Poisson log-linear model using the Delta Method

Jade Benjamin-Chung

12/5/2018

Objective

When using count data to obtain estimates of incidence rates, log-linear models are commonly since a population-offset can be conveniently included in the model, so individual-level data is not required. This document uses the Delta Method to obtain the standard error for a difference-in-difference parameter using the coefficients from a log-linear model, fit as follows:

$\ln\left(\frac{E[Y|X,T]}{PT}\right) = \beta_0 + \beta_1 X + \beta_2 T + \beta_3 X \cdot T$

- Y: count outcome
- PT: person-time used to calculate rate
- X: indicator for treatment
- T: indicator for time

The parameter of interest is:

$$\text{DID} = \left(\left(\frac{E[Y|X=1,T=1]}{PT}\right) - \left(\frac{E[Y|X=1,T=0]}{PT}\right)\right) - \left(\left(\frac{E[Y|X=0,T=1]}{PT}\right) - \left(\frac{E[Y|X=0,T=0]}{PT}\right)\right)$$

Using the coefficients from the model above, the transformation function is:

$F(\beta) = (e^{\beta_0+\beta_1+\beta_2+\beta_3} - e^{\beta_0+\beta_1}) - (e^{\beta_0+\beta_2} - e^{\beta_0})$

Calculate difference-in-difference by hand

```
# Pre-program numerator and denominator
data %>% group_by(dist) %>%
  filter(seas>=1112 & seas<1415) %>%
  group_by(seas, dist) %>%
  summarise(flucases = sum(flucases), pop = sum(N)) %>%
  group_by(dist) %>%
  summarise(flucases = mean(flucases), pop = mean(pop))
```

```
## # A tibble: 2 x 3
##   dist    flucases    pop
##   <fct>      <dbl>  <dbl>
## 1 OUSD        117. 447832
## 2 WCCUSD       56  255318
```

```
# 2014-15 numerator and denominator
data %>% filter(seas==1415) %>%
  group_by(dist) %>%
  summarise(flucases = sum(flucases), pop = sum(N))
```

```
## # A tibble: 2 x 3
##   dist   flucases    pop
##   <fct>     <dbl>  <dbl>
## 1 OUSD        158 447832
## 2 WCCUSD      113 255318
```

```
((158/447832)-(117/447832))-((113/255318)-(56/255318))
```

```
## [1] -0.0001316988
```

Obtain ratio of incidence ratios using Poisson model

```
glm.fit=glm(flucases ~ tr*time,offset=logN,data=d1415,
             family=poisson(link=log))
```

Use Delta Method to get variance for difference-in-difference

Step 1: Get partial derivatives of $F(\beta)$ with respect to each parameter

$\frac{dF}{d\beta_0} = (e^{\beta_0+\beta_1+\beta_2+\beta_3} - e^{\beta_0+\beta_1}) - (e^{\beta_0+\beta_2} - e^{\beta_0})$

$\frac{dF}{d\beta_1} = e^{\beta_0+\beta_1+\beta_2+\beta_3} - e^{\beta_0+\beta_1}$

$\frac{dF}{d\beta_2} = e^{\beta_0+\beta_1+\beta_2+\beta_3} - e^{\beta_0+\beta_2}$

$\frac{dF}{d\beta_3} = e^{\beta_0+\beta_1+\beta_2+\beta_3}$

```
# Gradient
b0 = glm.fit$coefficients[1]
b1 = glm.fit$coefficients[2]
b2 = glm.fit$coefficients[3]
b3 = glm.fit$coefficients[4]

dfdb0 = (exp(b0 + b1 + b2 + b3) - exp(b0 + b1)) -
          (exp(b0 + b2) - (exp(b0)))
dfdb1 = exp(b0 + b1 + b2 + b3) - exp(b0 + b1)
dfdb2 = exp(b0 + b1 + b2 + b3) - exp(b0 + b2)
dfdb3 = exp(b0 + b1 + b2 + b3)

grad <- c(dfdb0, dfdb1, dfdb2, dfdb3)
names(grad) <- c("dfdb0", "dfdb1", "dfdb2", "dfdb3")
grad
```

```
##         dfdb0         dfdb1         dfdb2         dfdb3
## -1.309545e-04  9.229652e-05 -8.977445e-05  3.528109e-04
```

Step 2: Get the covariance variance matrix from regression output

```
vb <- vcov(glm.fit)
vb
```

```
##              (Intercept)           tr         time      tr:time
## (Intercept)  0.005952352 -0.005952352 -0.005952352  0.005952352
## tr          -0.005952352  0.008809490  0.005952352 -0.008809490
## time        -0.005952352  0.005952352  0.014801884 -0.014801884
## tr:time      0.005952352 -0.008809490 -0.014801884  0.023988043
```

Step 3: Calculate the variance

$Var(F(\beta)) = JVJ^T$, where J is the Jacobian or gradient (derivative of $F(\beta)$ with respect to each parameter) and V is the variance covariance matrix.

```r
vF <- t(grad) %*% vb %*% grad
vF
```

```
##              [,1]
## [1,] 3.001531e-09
```

```r
seF <- sqrt(vF)
seF
```

```
##              [,1]
## [1,] 5.478623e-05
```

Obtain difference-in-difference and 95% CI from log-linear model

```r
get_did = function(fit){

# Gradient
b0 = fit$coefficients[1]
b1 = fit$coefficients[2]
b2 = fit$coefficients[3]
b3 = fit$coefficients[4]

dfdb0 = (exp(b0 + b1 + b2 + b3) - exp(b0 + b1)) -
          (exp(b0 + b2) - (exp(b0)))
dfdb1 = exp(b0 + b1 + b2 + b3) - exp(b0 + b1)
dfdb2 = exp(b0 + b1 + b2 + b3) - exp(b0 + b2)
dfdb3 = exp(b0 + b1 + b2 + b3)

grad <- c(dfdb0, dfdb1, dfdb2, dfdb3)
names(grad) <- c("dfdb0", "dfdb1", "dfdb2", "dfdb3")
grad

# Estimate DID
did = (exp(b0+b1+b2+b3) - exp(b0+b1)) - (exp(b0+b2) - exp(b0))

# Variance-covariance matrix
vb <- vcov(fit)
vb

# Variance of DID
vF <- t(grad) %*% vb %*% grad
vF
```

```r
se = sqrt(vF)

# 95% CI
lb = did - (qnorm(0.975)*sqrt(vF))
ub = did + (qnorm(0.975)*sqrt(vF))

return(list = c(did = did, se =se, lb = lb, ub = ub))

}

get_did(glm.fit)
```

```
## did.(Intercept)              se              lb              ub
##   -1.309545e-04    5.478623e-05   -2.383335e-04   -2.357546e-05
```

Use deltamethod package to obtain difference-in-difference and 95% CI from log-linear model

```r
g = as.formula(~ (exp(x1+x2+x3+x4) - exp(x1+x2)) - (exp(x1+x3) -
              exp(x1)))
deltamethod(g = g, coef(glm.fit), vcov(glm.fit))
```

```
## [1] 5.478623e-05
```

Compare to exponentiated beta3 coefficient

```r
summary(glm.fit)
```

```
##
## Call:
## glm(formula = flucases ~ tr * time, family = poisson(link = log),
##     data = d1415, offset = logN)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8259  -0.9487  -0.5815   0.3371   5.7110
##
## Coefficients:
##             Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -8.42491    0.07715 -109.200  < 2e-16 ***
## tr           0.17206    0.09386    1.833   0.0668 .
## time         0.70204    0.12166    5.770 7.91e-09 ***
## tr:time     -0.39876    0.15488   -2.575   0.0100 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2355.3  on 1627  degrees of freedom
## Residual deviance: 2314.3  on 1624  degrees of freedom
## AIC: 3547.8
```

```
##
## Number of Fisher Scoring iterations: 6
```