Sensitivity Analysis

Jeanne Li, Susan Glenn, Stewart Kerr, Jingcheng Xu

December 2019

Sensitivity analysis for paired matching with binary outcome

In our study, we matched treated to control subjects by similar measured covariates. We are now interested in what happens if we have unmeasured confounding covariates. Specifically, what is the magnitude of hidden bias needed to change our conclusions?

Sensitivity parameter Γ with $\alpha = 0.05$ as benchmark

Sensitivity analysis is conducted using Rosenbaum's R packages. Package "sensitivitymw" is used for matched pairs and "sensitivitymv" is used for amplification of the sensitivity parameter. The default of the "senmw" function is an unweighted M-test using Huber's psi-function, which can be used for any general setting, but reports a greater sensitivity to bias than other methods. Rosenbaum recommends the trimmed mean test for matched pairs with one matched control. We use both the unweighted M-test and the unweighted trimmed mean test and report them together for comparison.

The null hypothesis is that there is no treatment effect of wealth shock on mortality. The one-sided alternative hypothesis is that having a wealth shock leads to a negative impact on length of life span.

Assuming that one of the two subjects of our matched pair could have $\Gamma \geq 1$ times more likely than the other to get a wealth shock due to unmeasured, uncontrolled covariates. At $\Gamma > 1$, we get a set of possible p-values. The p-value reported in each column is the approximate upper bound on the one-sided p-value. We use a benchmark of $\alpha = 0.05$ and we increase Γ until our reported p-value is just below 0.05.

We want the results in the sequence following outcome of treated, outcome of control, and paired ID for easy use of "senmw" function, which reads either a matrix of outcome of treatment followed by outcome of control, or a vector of the differences in the outcome of treatment and outcome of control.

```r
result_matrix <- read.csv(file="../data/results-final.csv", header=TRUE, sep=",")
nonneg_outcomes <- subset(result_matrix,result_matrix$outcome!=-1)

Ymatrix <- matrix(0,nrow=length(nonneg_outcomes$pair_ID)/2,ncol=3)

k=0
for (i in unique(nonneg_outcomes$pair_ID)){
k=k+1
  for (j in which(result_matrix$pair_ID %in% i)){
    Ymatrix[k,3] = result_matrix$pair_ID[j]
    if (result_matrix$treated[j]==1){
      Ymatrix[k,1] = result_matrix$outcome[j]
    } else {
      Ymatrix[k,2] = result_matrix$outcome[j]
    }
  }
}

colnames(Ymatrix) = c("outcome_of_treated", "outcome_of_control", "paired_ID")
Ymatrix <- as.data.frame(Ymatrix)
```

We use the vector format to plug into "senmw."

```
Ymatrix$outcome_diff <- Ymatrix$outcome_of_treated - Ymatrix$outcome_of_control
Yvector <- Ymatrix$outcome_diff
Yvector_sgnchange <- -1*Ymatrix$outcome_diff #use this for senmw
```

We retrieve the results of the different gamma parameter values.

```
library(sensitivitymw)
#senmw for paired matching with one paired control
GammaSeq = seq(1,1.15,0.01) #list starting a 1, ending at 1.5, incrementing by 0.01
upperBound = matrix(0,length(GammaSeq),2) #length(GammaSeq)-by-3 matrix of zeroes (21x3 matrix)
for(i in 1:length(GammaSeq)) {
    upperBound[i,1] =senmw(Yvector_sgnchange, gamma = GammaSeq[i], method = "h")$pval
    upperBound[i,2] =senmw(Yvector_sgnchange, gamma = GammaSeq[i], method = "p")$pval
}
out = cbind(GammaSeq,upperBound)
colnames(out) = c("Gamma","unweighted M-test using Huber's psi-function","trimmed mean test")
round(out,3)
```

```
##       Gamma unweighted M-test using Huber's psi-function trimmed mean test
##  [1,]  1.00                                        0.009             0.009
##  [2,]  1.01                                        0.014             0.014
##  [3,]  1.02                                        0.021             0.021
##  [4,]  1.03                                        0.032             0.032
##  [5,]  1.04                                        0.046             0.046
##  [6,]  1.05                                        0.064             0.064
##  [7,]  1.06                                        0.088             0.088
##  [8,]  1.07                                        0.117             0.117
##  [9,]  1.08                                        0.152             0.152
## [10,]  1.09                                        0.194             0.194
## [11,]  1.10                                        0.240             0.240
## [12,]  1.11                                        0.292             0.292
## [13,]  1.12                                        0.348             0.348
## [14,]  1.13                                        0.407             0.407
## [15,]  1.14                                        0.467             0.467
## [16,]  1.15                                        0.528             0.528
```

The two columns report the same p-value for the same gamma value. Using a benchmark of $\alpha = 0.05$, we see that $\Gamma = 1.04$ is where we are just below surpassing the 0.05 threshold. We use $\Gamma = 1.04$ for getting point estimates, confidence intervals, and amplification of the sensitivity parameter.

Range of point estimates and confidence intervals:

```
conf_ints <- senmwCI(Yvector_sgnchange, gamma = 1.04, method = "p", one.sided = TRUE)
conf_ints
```

```
## $PointEstimate
## minimum maximum
##  0.0247  0.0456
##
## $Confidence.Interval
## minimum maximum
##   6e-04     Inf
```

Amplification of Γ

Let $\Gamma = \frac{\beta_{UA}\beta_{UY}+1}{\beta_{UA}+\beta_{UY}}$. The β's measure the unmeasured confounding covariate U's odds of increasing wealth shock (β_{UA}) and mortality (β_{UY}). We solve for E by setting $\beta_{UA} = \beta_{UY} = E$ and using $\Gamma = 1.04$.

```
library(sensitivitymv)
```

```
##
## Attaching package: 'sensitivitymv'

## The following objects are masked from 'package:sensitivitymw':
##
##     mscorev, multrnks, newurks
```

```
amplify(1.04, c(2:5)) #\beta_{UA} amplitude c(2:5)
```

```
##        2        3        4        5
## 1.125000 1.081633 1.067568 1.060606
```

```
uniroot(function(x){amplify(1.04,x) - x},c(1.04+0.01,10))$root #E-value 1.325657
```

```
## [1] 1.325657
```

Therefore, in order to overturn our current conclusion, an unmeasured confounder needs to have at least a 1.33 odds of increasing the outcome and treatment. Since 1.33 is a low threshold to surpass, our study is sensitive to unmeasured covariates.