analysis-Matching

ss5929

8/19/2020

Matching

Assumptions

1. No spillover effect but allow for some carryover effect. The potential outcome for unit i at time t+F depends neither on the treatment status of other units, e.g., $
2. Sequential ignorability states that conditional on the treatment, outcome, and covariate history up to time t-L, the treatment assignment is unconfounded.

$$\left\{Y_{i,t+F}\left(X_{it}=1, X_{i,t-1}=0, \{X_{i,t-\ell}\}_{\ell=2}^{L}\right), Y_{i,t+F}\left(X_{it}=0, X_{i,t-1}=0, \{X_{i,t-\ell}\}_{\ell=2}^{L}\right)\right\} \perp X_{it} \mid X_{i,t-1}=0, \{X_{i,t-\ell}\}_{\ell=2}^{L}, \{Y_{i,t-\ell}\}_{\ell=1}^{L}, \{\mathbf{Z}_{i,t-\ell}\}_{\ell=0}^{L}$$

3. parallel trend defines that we also adjust for

construct matching sets

Matching sets for subject i is:

$$M_{it} = \{i' : i' \neq i, X_{i't}=0, X_{i't'}=X_{it'} \ for\ all\ t'=t-1,...,t-L\}$$

here i means differet subjects and t means time point.

The matched sets only adjust for treatment h

refinement methods

Here, the author introduces three matching and weighting method to refine the matching set.

1. Mahalanobis distance: Suppose that we wish to match each treated observation with at most J control units from the matched set with replacement, i.e.,$|M_{it}| \leq J$. The average Mahalanobis distance between the treated observation and each control observation over time is:

$$S_{it}(i') = \frac{1}{L}\sum_{\ell=1}^{L}\sqrt{\left(\mathbf{V}_{i,t-\ell}-\mathbf{V}_{i',t-\ell}\right)^{\top}\boldsymbol{\Sigma}_{i,t-\ell}^{-1}\left(\mathbf{V}_{i,t-\ell}-\mathbf{V}_{i',t-\ell}\right)}$$

   for a matched control unit $i' \in M_{it}$, where $V_{it'}$ represents the time-varying covariates one wishes to adjust for and and $\boldsymbol{\Sigma}_{it'}$ is the sample covariance matrix of $V_{it'}$. For a given control unit in the matched set, we compute the standardized distance using the time varing covariates and average it across time periods.

2. propensity score

We can also use distance measure based on the estimated propensity score which is defined as the conditional probability of treatment assignment given pre-treatment covariates. To estimate propensity score, we first create a subset of data consisting of all treated observations and their matched control observations from the same year and then fit a treatment assignment model to this data set. For example, a logistic regression model:

$$e_{it}\left(\{\mathbf{U}_{i,t-\ell}\}_{\ell=1}^{L}\right) = \Pr\left(X_{it}=1 \mid \mathbf{U}_{i,t-1},\ldots,\mathbf{U}_{i,t-L}\right) = \frac{1}{1+\exp\left(-\sum_{\ell=1}^{L}\boldsymbol{\beta}_{\ell}^{\top}\mathbf{U}_{i,t-\ell}\right)}$$

where $\mathbf{U}_{it'} = \left(X_{it'}, \mathbf{V}_{it'}^{\top}\right)^{\top}$.

Given the fitted model, we compute the propensity score for all treated observations and their matched control observations. Then, we adjust for the lagged outcomes and covariates

3. weighting (e.g. inverse propensity score weighting)

mobile health project

analysis process

All time points with $x_j = 1$ will be considered as the treated while $x_j = 0$ will be considered as control. For each treated observation j, there will be a matching set with:

$$M(j) = \{j' : j' \neq j,\ X_{j'} = 0,\ X_{j-L} = X_{j'-L}\}$$

here I choose L=2, which means the matching sets are constructed according to previous two x(i.e., $x_{j-1} = x_{j'-1}$ and $x_{j-2} = x_{j'-2}$).

Since the outcome model is:

$$Y_{t(j)} = \zeta_t + \alpha_t X_{t(j)} + \phi_t Y_{t(j-1)} + \eta_{t(j)} U_{t(j)} + \varepsilon_{t(j)}$$

The propensity score is defined as the probability of x=1 given previous y and other covariates, which is:

$$p(x_j) = prob(x_j = 1 | y_{j-1}, u_j) = E(x_j | y_{j-1}, u_j)$$

Here I use the simplest matching method: nearest neighbour to find the most matching time point j', which means $min|p_j - p_{j'}|$ for x_j, and the causal effect will be $y_j - y_{j_{most\ match}}$ for every time point j.

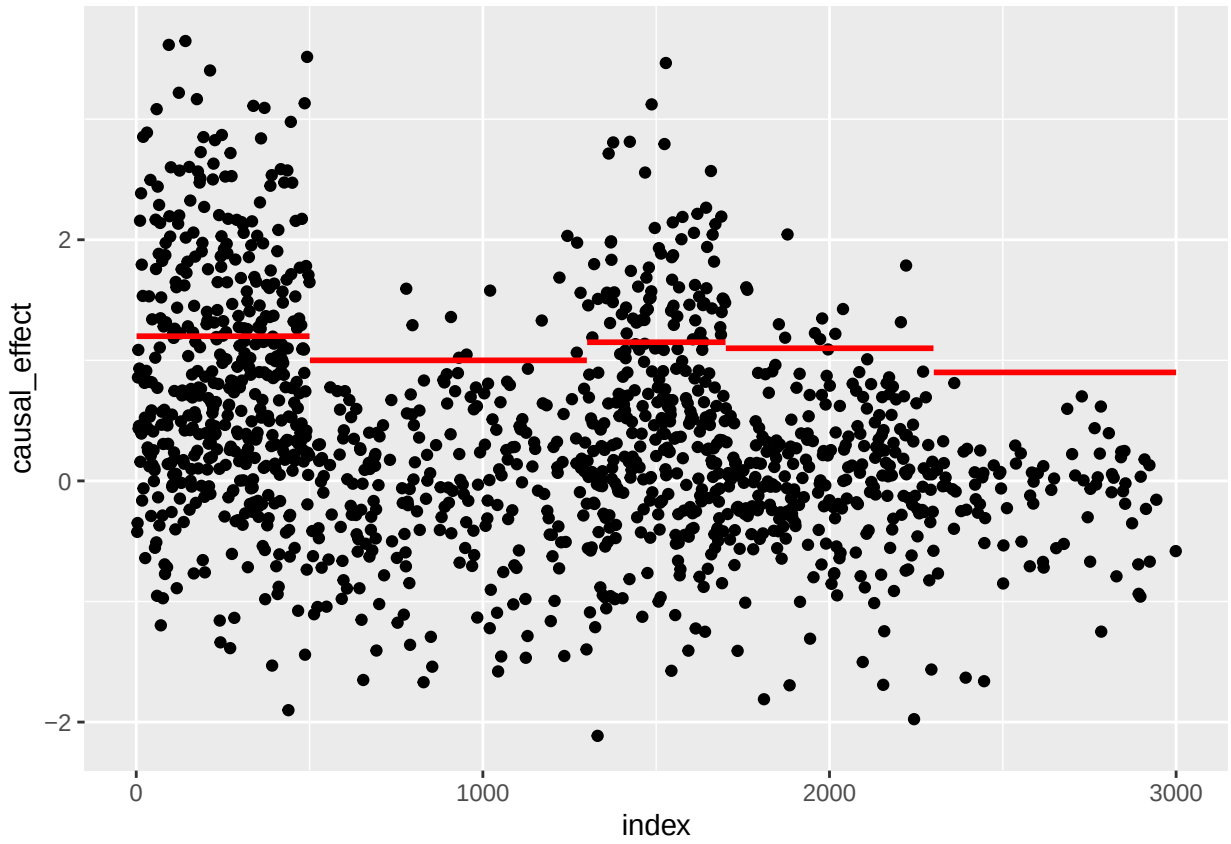
```
final = read.csv("final.csv")
final_match = final %>% mutate(lag_y = lag(y))

#calculate the propensity score
glm1 = glm(x~u+lag_y,family = binomial,data=final_match)
summary(glm1)
```

```
##
## Call:
## glm(formula = x ~ u + lag_y, family = binomial, data = final_match)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3028  -0.8901  -0.5910   0.9334   2.1265
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.24363    0.11123 -20.171  < 2e-16 ***
## u            0.86412    0.12389   6.975 3.06e-12 ***
## lag_y        0.82487    0.04861  16.970  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4116.8  on 2998  degrees of freedom
## Residual deviance: 3433.2  on 2996  degrees of freedom
##   (1 observation deleted due to missingness)
```

```
## AIC: 3439.2
##
## Number of Fisher Scoring iterations: 3
```

```
propensity_score = c(NA,glm1$fitted)

mydata = cbind(final_match,propensity_score)

# finding the most matching y, using the simplest propensity score matching method: nearest neighbour

m = NULL
index = NULL
for (i in 3:3000)
{
  a=1
  if(mydata$x[i] == 1)
  {
  for(j in 4:3000)
  {
    if(mydata$x[j]==0 &&
       mydata$x[j-1]==mydata$x[i-1] &&
       mydata$x[j-2]==mydata$x[i-2] &&
       abs(mydata$propensity_score[j]-mydata$propensity_score[i])<a)
    {
      a=abs(mydata$propensity_score[j]-mydata$propensity_score[i])
      m[i] = mydata$y[j]
      index[i] = j
    }
  }
  }
  else
  {
    m[i] = NA
    index[i]=NA
  }
}

matching_data = cbind(mydata,index,m) %>% mutate(causal_effect = .$y-.$m) %>% mutate(true_effect = c(re

ggplot(data = matching_data)+
  geom_point(aes(x=index,y=causal_effect))+
  geom_step(aes(x=index,y=true_effect, group=true_effect),col="red",size=1)
```

```
## Warning: Removed 1676 rows containing missing values (geom_point).
```




```
# difference in difference estimator
lag_m = NULL
m = NULL
index = NULL
for (i in 3:3000)
{
  a=1
  if(mydata$x[i] == 1 && mydata$x[i-1]==0) # a change from j-1 to j
  {
  for(j in 4:3000)
  {
    if(mydata$x[j]==0 &&
       mydata$x[j-1]==mydata$x[i-1] &&
       mydata$x[j-2]==mydata$x[i-2] &&
       abs(mydata$propensity_score[j]-mydata$propensity_score[i])<a)
    {
      a=abs(mydata$propensity_score[j]-mydata$propensity_score[i])
      m[i] = mydata$y[j]
      lag_m[i] = mydata$y[j-1]
      index[i] = j
    }
  }
  }
  else
  {
    m[i] = NA
    index[i]=NA
```

```
    lag_m[i]=NA
  }
}

matching_data = cbind(mydata,index,m,lag_m) %>% mutate(causal_effect = (.$y-lag(.$y))-(.$m-.$lag_m))
plot(matching_data$causal_effect)
abline(h=0)
```