Directly define α

JCY

2023 年 7 月 30 日

The doubly robust score (Robins and Rotnitzky, 1995).

$$\Gamma_{d,s,i} = \frac{1\{S_i = s\}}{p_s}\left[\frac{1\{D_i = d\}}{e(X_i, S_i)}(Y_i - m_{d,s}(X_i)) + m_{d,s}(X_i)\right] \tag{1}$$

can be estimated by $\hat{\Gamma}_{d,s,i}$ using $\hat{p}_1$ and $\hat{e}(x,s)$, $\hat{m}_{d,s}(.)$.

```
source('./library/library.R')
## This file esitmates the model
### Real Data
library(readstata13)
data <- read.dta13('./data/113492-V1/P2017_1008_data/Lyons_
    Zhang2016_AERPP_analysisdata.dta')
## Estimate propensity score
data <- na.omit(data)
D <- data$accepted
S <- data$female
propensity2 <- mean(S)
Y <- data$short_run_activity
library(glmnet)
propensity1 <- cross_fitting_propensity(D,as.matrix(data[, c(5,
    6,7,8, 9, 10, 11, 12, 13, 14)]), seeds = 123, K = 5)
#propensity1 is e(x,s)
#propensity2 is p(s=1)
X <- data[,c(5,8,9,10,11, 12, 13, 14)]
X_int <- X[, c(3,4,5,6, 7, 8)]
X_reg <- cbind(X, D, D * X_int * S, D * X_int * (1 - S), S * X_
    int, D * X_int) #选择协变量
conditional_means <- cross_fitting_mean(Y, X_reg, X, X_int,
    seeds = 123, K = 5) #治疗d下组s的条件平均值
m_hat11 <- conditional_means[,1]#就是定义的m(d,s)
m_hat10 <- conditional_means[,2]
m_hat01 <- conditional_means[,3]
m_hat00 <- conditional_means[,4]
tau_hat1 <- m_hat11 - m_hat10  #d=1 ,s=1,0
```

```
tau_hat0 <- m_hat01 - m_hat00 #d=0, s=1,0
difference <- abs(m_hat11 - m_hat01)  - abs(m_hat10 - m_hat00)
    #s相同 d不同的绝对差异
difference <-difference * S/propensity2 + difference * (1 - S)
    /(1 - propensity2) #
library(foreach)
library(doParallel)
numcores <- 11
registerDoParallel(numcores)
cost_treatment=0;
m1 <- S*m_hat11 + (1 - S)*m_hat01
m0 <- S*m_hat10 + (1 - S)*m_hat00

G_i1 <- (Y - m0) * (1-D) * S / ((1-propensity1)*propensity2) +
    m0 * S/propensity2
G_i2 <- (Y - cost_treatment - m1) * D * S/ (propensity1*
    propensity2) + m1 * S/propensity2
g_i_S = G_i2 - G_i1

G_i12 <- (Y - m0) * (1-D) * (1 - S) / ((1-propensity1)*(1 -
    propensity2)) + m0 * (1 - S)/(1 - propensity2)
G_i22 <- (Y - cost_treatment - m1) * D * (1 - S)/ (propensity1
    *(1 - propensity2)) + m1 * (1 - S)/(1 - propensity2)
g_i_S2 = G_i22 - G_i12
```

1 Using Γ to choose α

First, we rank individuals based on their Γ_i values, and then select the top 150 individuals with the highest Γ_i values (since the maximum number of treatments is 150). Next, we find the proportion of males and females among the selected 150 individuals.

```
total=vector()
for(i in 1:length(D)){
    if (g_i_S[i]!=0){
            total[i]=g_i_S[i]
    }
    else{
            total[i]=g_i_S2[i]
    }
}

malenum=0;
pre_100=tail(sort(total),150);
```

```
for(i in 1:150){
    if(pre_100[i] %in% g_i_S2){
            malenum=malenum+1;
    }
}
malenum=malenum/150;
```

We find that the proportion of female in this 150-treament group is 28% in constrast to 72% which is the proportion of male.

Then we estimate the maximum score in this deterministic assignment(male 72%, female 28%)

```
X = cbind(S, X[, c(2,8)])
scale_Y = F
cost_treatment = 0
params=NA
maxtime1 = 5000
maxtime2 = 5000
max_treated_units = 150
quick_run = F
no_parity_constraint = T
additional_fairness_constraint = F
distance = 'envy'
library(gurobi)
tolerance = 10**(-3)
numcores=4;
source('./library/helpers.R')
library(gurobi)
maxtime = 300
m1 = 0
m0 = 0
additional_fairness_constraint = F
parity_constraint = '>='
probabilistic = F
threshold_probabilistic = F
parallel = F
tolerance = 10**(-3)
result <- Est_max_score(Y, X, D, S, propensity1, propensity2,B=1, params =
    params, tolerance_constraint = tolerance,
scale_Y = scale_Y, additional_fairness_constraint = additional_fairness_
    constraint,
parity_constraint = parity_constraint, cost_treatment  = cost_treatment,
    alpha = 0.28, #alpha可改即为帕累托的alpha
max_treated_units = max_treated_units, maxtime = maxtime, m1 = m1, m0 = m0,
     cores = numcores)
```

Afterwards, we will calculate how many of the top 150 individuals with the highest Γ values will be selected for treatment under the optimized allocation.

```
est=result$pi
comp=cbind(S,total,est)
pl=comp[order(comp[,2],decreasing = TRUE),]
sum(pl[1:150,3])
```

The result is 65. The proportion of individuals correctly assigned to treatment based solely on the proportion calculation of gamma values is 0.4333333

The beta value of the results obtained by directly calculating the proportion based on Γ values is 1, which is very different from the results obtained by calculating through the model directly.

```
 result$beta
[1] -1.0000000  0.2491919  0.2892211 -0.6804476

res_ms1_envy$result$beta
[1]  1.0000000  0.9750415 -0.1684758 -0.3662181
```

```
envy_gender
[1] 0.2059701
> parrity_gender
[1] 0.2179104
> parity_abs_gender
[1] 0.1223881
```

2 Compare to other fair policy learning method

We consider three notions of UnFairness:

2.1 Counterfactual Envy

Let the conditional welfare, for the policy function being assigned to the opposite attribute, i.e., the effect of $\pi(x, s_1)$, on the group s_2 , conditional on covariates, be

$$V_{\pi(x,s_1)}(x, s_2) = \mathbb{E}\left[\pi(x, s_1) Y_i(1, s_2) + (1 - \pi(x, s_1)) Y_i(0, s_2) \mid X_i(s_2) = x\right] \tag{2}$$

We say that the agent with attribute s_2 envies the agent with attribute s_1, if her welfare (on the right-hand side of Equation (3)) exceeds the welfare she would have received had her covariate and policy been assigned the opposite attribute (left-hand side of Equation (3)), namely

$$\mathbb{E}_{X(s_1)}\left[V_{\pi(X(s_1),s_1)}(X(s_1), s_2)\right] > \mathbb{E}_{X(s_2)}\left[V_{\pi(X(s_2),s_2)}(X(s_2), s_2)\right] \tag{3}$$

We then measure the unfairness towards an individual with attribute s_2 as

$$\mathcal{A}(s_1, s_2; \pi) = \mathbb{E}_{X(s_1)}\left[V_{\pi(X(s_1),s_1)}(X(s_1), s_2)\right] - \mathbb{E}_{X(s_2)}\left[V_{\pi(X(s_2),s_2)}(X(s_2), s_2)\right]. \tag{4}$$

Whenever we aim not to discriminate in either direction, we take the sum of the effects $\mathcal{A}(s_1, s_2; \pi)$ and $\mathcal{A}(s_2, s_1; \pi)$ it connects to previous notions of counterfactual fairness (Kilbertus et al., 2017).

2.2 Predictive Disparity

Prediction disparity and its empirical counterpart take the following form

$$C(\pi) = \mathbb{E}[\pi(X, S) \mid S = 0] - \mathbb{E}[\pi(X, S) \mid S = 1], \quad \hat{C}(\pi) = \frac{\sum_{i=1}^{n} \pi(X_i)(1 - S_i)}{n(1 - \hat{p}_1)} - \frac{\sum_{i=1}^{n} \pi(X_i) S_i}{n \hat{p}_1}, \tag{5}$$

Prediction disparity captures disparity in the treatment probability between groups.

(Welfare disparity). Define the welfare disparity and its empirical counterpart as

$$D(\pi) = W_0(\pi) - W_1(\pi), \quad \widehat{D}(\pi) = \widehat{W}_0(\pi) - \widehat{W}_1(\pi). \tag{6}$$

2.3 Predictive disparity with absolute value

Predictive disparity with absolute value.

The policymaker may also consider $|D(\pi)|$ or $|C(\pi)|$ as measures of UnFairness, in which case the policymaker treats the two groups symmetrically.

Compare the proportions calculated by the other models in fair policy learning with the proportions obtained by sorting based on Γ values.

```
#进行对比
load('./results/alg_parity.RData')
load('./results/elements.RData')
load('./results/alg_envy.RData')
load('./results/alg_parity_abs.RData')
res_ms1_parity_abs <- res_ms1_parity
res_ms3_parity_abs <- res_ms3_parity
load('./results/alg_parity.RData')
envy=res_ms1_envy$result$policies
parity_abs=res_ms1_parity_abs$result$policies
parity=res_ms1_parity$result$policies
comp=cbind(S,total,envy,parity,parity_abs)
envy_gender<-0
parrity_gender<-0
parity_abs_gender<-0
for(i in 1:335){
     if(comp[i,3]==1){
             envy_gender=envy_gender+comp[i,1]
     }
     if(comp[i,4]==1){
             parrity_gender=parrity_gender+comp[i,1]
     }
```

```
    if(comp[i,5]==1){
            parity_abs_gender=parity_abs_gender+comp[i,1]
    }
}

envy_gender<-envy_gender/335
parrity_gender<-parrity_gender/335
parity_abs_gender<-parity_abs_gender/335

envy_gender
parrity_gender
parity_abs_gender
```

The result are:

```
    envy_gender
    [1] 0.2059701
    > parrity_gender
    [1] 0.2179104
    > parity_abs_gender
    [1] 0.1223881
```

The proportions obtained by sorting based on Γ values is 28%, so it is different. We show whether each individual has been treated or not in different methods.

```
    S total       envy parity parity_abs est
86  1   7.50463110   1      1          1   1
30  1   6.72587692   1      1          0   0
13  1   6.31825313   1      1          1   1
4   1   5.66679479   1      1          1   1
262 1   5.66305117   1      1          1   1
75  1   5.48835486   1      1          0   0
9   1   5.13332426   1      1          0   0
124 1   4.93094181   1      1          0   0
79  1   4.71074369   1      1          1   1
82  1   4.58036789   0      1          0   1
31  1   4.38277687   0      0          0   1
101 1   4.26440478   1      1          0   0
191 0   4.22121476   1      0          1   0
51  1   4.00446017   1      1          1   1
235 1   3.90775514   1      1          1   0
119 1   3.71645202   0      0          0   0
237 0   3.69495936   1      1          1   0
221 0   3.60500684   1      1          1   0
251 0   3.59042885   1      1          1   0
69  1   3.40300450   1      1          0   0
```

```
147 1    3.30163529    0       0          0   0
134 1    3.21477597    1       1          1   1
356 1    2.79109025    1       1          0   0
10  1    2.78079896    1       1          0   0
216 1    2.53026005    1       1          1   1
103 0    2.43480164    0       0          1   0
74  1    2.42321199    1       1          0   0
335 0    2.24761100    0       0          1   1
58  0    2.22028988    0       0          0   0
199 0    2.19789487    0       0          0   0
24  0    2.17050890    0       0          1   1
29  0    2.16223668    0       0          1   1
14  0    2.13905361    0       0          0   0
201 0    2.13144217    0       0          1   1
208 0    2.12520605    0       0          1   1
222 0    2.02347926    0       0          1   1
194 0    1.99332122    0       0          0   1
111 0    1.97375185    0       0          0   1
7   0    1.93372477    0       0          0   1
95  0    1.92281949    0       0          0   1
198 0    1.92070857    0       0          0   0
38  0    1.91595826    0       0          1   1
133 0    1.90783226    0       0          0   0
112 1    1.89943444    1       1          0   0
171 0    1.88874043    0       0          1   1
161 0    1.87854104    0       0          0   1
92  1    1.86010076    1       1          1   0
144 0    1.83788583    0       0          0   0
298 1    1.83172444    1       1          0   0
267 0    1.80707470    0       0          0   1
241 0    1.78317083    0       0          0   1
276 0    1.77777455    0       0          1   1
342 0    1.70879571    0       0          0   0
158 0    1.67656692    0       0          0   1
93  1    1.65465169    1       1          1   0
2   1    1.64957463    1       1          1   1
54  0    1.64028037    0       0          0   1
205 0    1.63945446    0       0          0   1
167 0    1.61605686    0       0          0   1
71  0    1.56688910    0       0          0   0
139 0    1.55858097    0       0          0   0
127 0    1.49111444    0       0          0   1
150 1    1.46701494    1       1          1   1
```

```
132 0    1.43671748    0      0          0   1
90  0    1.43042445    0      0          1   1
22  0    1.41974267    0      0          0   0
192 0    1.33537820    0      0          0   0
141 0    1.32919473    0      0          1   1
108 0    1.32240908    0      0          0   1
155 1    1.32067788    1      1          0   0
96  0    1.31789234    0      0          0   1
99  0    1.27805897    0      0          0   1
...
```

We found that only a few of the top 150 individuals with the largest Γ values in each method were treated, as follows:

```
     sum(pl[1:150,3])
     [1] 47
     > sum(pl[1:150,4])
     [1] 45
     > sum(pl[1:150,5])
     [1] 76
     > sum(pl[1:150,6])
     [1] 65
```

We compare the number of individuals treated under the optimal Γ selection with the number of individuals selected for treatment under other fair policy learning methods.

```
re<-c(0,0,0)
for(i in 1:335){
     if(pl[i,6]==1){
             for(j in 1:3){
                     if(pl[i,j+2]==1){
                             re[j]=re[j]+1
                     }
             }
     }
}
```

The number of individuals overlapping between envy method selected and those selected using the optimal Γ selection is 32.The number of individuals overlapping between Predictive disparity method selected and those selected using the optimal Γ selection is 36. The number of individuals overlapping between Predictive disparity with absolute value method selected and those selected using the optimal Γ selection is 71.

3 Conclusion

(1)β

The β value of the results obtained by directly calculating the proportion based on Γ values is 1, which is very different from the results obtained by calculating through other model directly.

(2)**The proportion of males and females selected** The proportion of female in this 150-treament group is 28% in constrast to 72% which is the proportion of male.

In envy unfairness,the proportion of female in this 150-treament group is 20.59%,In Predictive disparity unfairness,the proportion of female in this 150-treament group is 21.79%,In Predictive disparity with absolute value unfairness,the proportion of female in this 150-treament group is 12.23%. So they are very different.

(3)**Top 150 with the largest gamma values who were selected for treatment** The number of individuals among the top 150 with the largest gamma values who were selected for treatment is 65.

In envy unfairness, the number of individuals among the top 150 with the largest gamma values who were selected for treatment is 47. In Predictive disparity unfairness, the number of individuals among the top 150 with the largest gamma values who were selected for treatment is 45. In Predictive disparity with absolute value unfairness, the number of individuals among the top 150 with the largest gamma values who were selected for treatment is 76.

(4)Compare the number of individuals treated under the optimal Γ selection with the number of individuals selected for treatment under other fair policy learning methods. The number of individuals overlapping between envy method selected and those selected using the optimal Γ selection is 32.The number of individuals overlapping between Predictive disparity method selected and those selected using the optimal Γ selection is 36. The number of individuals overlapping between Predictive disparity with absolute value method selected and those selected using the optimal Γ selection is 71.

In summary, we can not directly use Γ to decide α.

4 Code

```
#S中1是女，0是男
source('./library/library.R')
## This file esitmates the model
### Real Data
library(readstata13)
data <- read.dta13('./data/113492-V1/P2017_1008_data/Lyons_Zhang2016_AERPP_
    analysisdata.dta')
## Estimate propensity score
data <- na.omit(data)
D <- data$accepted
```

```r
S <- data$female
propensity2 <- mean(S)
Y <- data$short_run_activity
library(glmnet)
propensity1 <- cross_fitting_propensity(D,as.matrix(data[, c(5,6,7,8, 9, 10,
    11, 12, 13, 14)]), seeds = 123, K = 5)
#propensity1 is e(x,s)
#propensity2 is p(s=1)
X <- data[,c(5,8,9,10,11, 12, 13, 14)]
X_int <- X[, c(3,4,5,6, 7, 8)]
X_reg <- cbind(X, D, D * X_int * S, D * X_int * (1 - S), S * X_int, D * X_
   int) #选择协变量
conditional_means <- cross_fitting_mean(Y, X_reg, X, X_int, seeds = 123, K =
    5) #治疗d下组s的条件平均值
m_hat11 <- conditional_means[,1]#就是定义的m(d,s)
m_hat10 <- conditional_means[,2]
m_hat01 <- conditional_means[,3]
m_hat00 <- conditional_means[,4]
tau_hat1 <- m_hat11 - m_hat10  #d=1 ,s=1,0
tau_hat0 <- m_hat01 - m_hat00 #d=0, s=1,0
difference <- abs(m_hat11 - m_hat01)  - abs(m_hat10 - m_hat00) #s相同 d不同
   的绝对差异
difference <-difference * S/propensity2 + difference * (1 - S)/(1 -
   propensity2) #

cost_treatment=0;
m1 <- S*m_hat11 + (1 - S)*m_hat01
m0 <- S*m_hat10 + (1 - S)*m_hat00

G_i1 <- (Y - m0) * (1-D) * S / ((1-propensity1)*propensity2) + m0 * S/
   propensity2
G_i2 <- (Y - cost_treatment - m1) * D * S/ (propensity1*propensity2) + m1 *
   S/propensity2
g_i_S = G_i2 - G_i1

G_i12 <- (Y - m0) * (1-D) * (1 - S) / ((1-propensity1)*(1 - propensity2)) +
   m0 * (1 - S)/(1 - propensity2)
G_i22 <- (Y - cost_treatment - m1) * D * (1 - S)/ (propensity1*(1 -
   propensity2)) + m1 * (1 - S)/(1 - propensity2)
g_i_S2 = G_i22 - G_i12

total=vector()
for(i in 1:length(D)){
```

```
		if (g_i_S[i]!=0){
			total[i]=g_i_S[i]
		}
		else{
			total[i]=g_i_S2[i]
		}
	}

	malenum=0;
	pre_100=tail(sort(total),150);
	for(i in 1:150){
		if(pre_100[i] %in% g_i_S2){
			malenum=malenum+1;
		}
	}
	malenum=malenum/150;
	#找到最佳男生比例

	X = cbind(S, X[, c(2,8)])
	scale_Y = F
	cost_treatment = 0
	params=NA
	maxtime1 = 5000
	maxtime2 = 5000
	max_treated_units = 150
	quick_run = F
	no_parity_constraint = T
	additional_fairness_constraint = F
	distance = 'envy'
	library(gurobi)
	tolerance = 10**(-3)
	numcores=4;
	source('./library/helpers.R')
	library(gurobi)
	maxtime = 300
	m1 = 0
	m0 = 0
	additional_fairness_constraint = F
	parity_constraint = '>='
	probabilistic = F
	threshold_probabilistic = F
	parallel = F
	tolerance = 10**(-3)
```

```
result <- Est_max_score(Y, X, D, S, propensity1, propensity2,B=1, params =
    params, tolerance_constraint = tolerance,
scale_Y = scale_Y, additional_fairness_constraint = additional_fairness_
    constraint,
parity_constraint = parity_constraint, cost_treatment  = cost_treatment,
    alpha = 0.28, #alpha可改即为帕累托的alpha
max_treated_units = max_treated_units, maxtime = maxtime, m1 = m1, m0 = m0,
    cores = numcores)
est=result$pi
comp=cbind(S,total,est)
pl=comp[order(comp[,2],decreasing = TRUE),]
sum(pl[1:150,3])

result$beta

#进行对比
load('./results/alg_parity.RData')
load('./results/elements.RData')
load('./results/alg_envy.RData')
load('./results/alg_parity_abs.RData')
res_ms1_parity_abs <- res_ms1_parity
res_ms3_parity_abs <- res_ms3_parity
load('./results/alg_parity.RData')
envy=res_ms1_envy$result$policies
parity_abs=res_ms1_parity_abs$result$policies
parity=res_ms1_parity$result$policies
comp=cbind(S,total,envy,parity,parity_abs,est)
envy_gender<-0
parrity_gender<-0
parity_abs_gender<-0
for(i in 1:335){
     if(comp[i,3]==1){
             envy_gender=envy_gender+comp[i,1]
     }
     if(comp[i,4]==1){
             parrity_gender=parrity_gender+comp[i,1]
     }
     if(comp[i,5]==1){
             parity_abs_gender=parity_abs_gender+comp[i,1]
     }
}

envy_gender<-envy_gender/335
```

```r
parrity_gender<-parrity_gender/335
parity_abs_gender<-parity_abs_gender/335

envy_gender
parrity_gender
parity_abs_gender
pl=comp[order(comp[,2],decreasing = TRUE),]
re<-c(0,0,0)
for(i in 1:335){
     if(pl[i,6]==1){
             for(j in 1:3){
                     if(pl[i,j+2]==1){
                             re[j]=re[j]+1
                     }
             }
     }
}
re
sum(pl[1:150,3])
sum(pl[1:150,4])
sum(pl[1:150,5])
sum(pl[1:150,6])
```