CEM Results

Skip Moses

4/3/2022

Summary

This is a document for Coarsened Exact Mathcing results.

BIO104

```
# 2017 Student grades/SI/demographic data
data <- read.csv("data/CEM_dataset.csv")
data.fall <- data %>% filter(Term.Type == "Fall")

# Remove Withdraws from class
data.fall <- data.fall %>%
  filter(Student.Class.Grade.Point.per.Unit > 0)

# COnvert to factors
data.fall$SI.Attended <- as.factor(data.fall$SI.Attended)
data.fall$Random.Student.ID <- as.factor(data.fall$Random.Student.ID)
data.fall$Random.Course.ID <- as.factor(data.fall$Random.Course.ID)

# Center the grade per unit
data.fall$Grade.Point.per.Unit.center <- (data.fall$Student.Class.Grade.Point.per.Unit - mean(data.fall$
```

```
# Estimate Propensity Scores
cov <- c("HS.GPA",
  "Student.Orientation.Flag",
  "Major.1.STEM.Flag",
  "Full.Time.Part.Time.Code",
  "Academic.Program",
  "Random.Course.ID")

data.fall %>% group_by(SI.Attended) %>%
  select(one_of(cov)) %>% na.omit() %>%
  summarise_all(funs(mean(., na.rm = T)))
```

```
## Adding missing grouping variables: `SI.Attended`
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
```

```
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
## Warning in mean.default(Student.Orientation.Flag, na.rm = T): argument is not
## numeric or logical: returning NA
```

```
## Warning in mean.default(Student.Orientation.Flag, na.rm = T): argument is not
## numeric or logical: returning NA
```

```
## Warning in mean.default(Major.1.STEM.Flag, na.rm = T): argument is not numeric
## or logical: returning NA
```

```
## Warning in mean.default(Major.1.STEM.Flag, na.rm = T): argument is not numeric
## or logical: returning NA
```

```
## Warning in mean.default(Full.Time.Part.Time.Code, na.rm = T): argument is not
## numeric or logical: returning NA
```

```
## Warning in mean.default(Full.Time.Part.Time.Code, na.rm = T): argument is not
## numeric or logical: returning NA
```

```
## Warning in mean.default(Academic.Program, na.rm = T): argument is not numeric or
## logical: returning NA
```

```
## Warning in mean.default(Academic.Program, na.rm = T): argument is not numeric or
## logical: returning NA
```

```
## Warning in mean.default(Random.Course.ID, na.rm = T): argument is not numeric or
## logical: returning NA
```

```
## Warning in mean.default(Random.Course.ID, na.rm = T): argument is not numeric or
## logical: returning NA
```

```
## # A tibble: 2 x 7
##   SI.Attended HS.GPA Student.Orientation.Flag Major.1.STEM.Flag Full.Time.Part.~
##   <fct>        <dbl>                   <dbl>             <dbl>            <dbl>
## 1 0             3.42                      NA                NA               NA
## 2 1             3.39                      NA                NA               NA
## # ... with 2 more variables: Academic.Program <dbl>, Random.Course.ID <dbl>
```

```
summary(factor(data.fall$SI.Attended))
```

```
##   0   1
## 402 867
```

```
prop.score <- glm(SI.Attended ~ HS.GPA +
                    Student.Orientation.Flag +
                    Major.1.STEM.Flag +
                    Random.Course.ID, family = binomial, data = data.fall)

summary(prop.score)
```

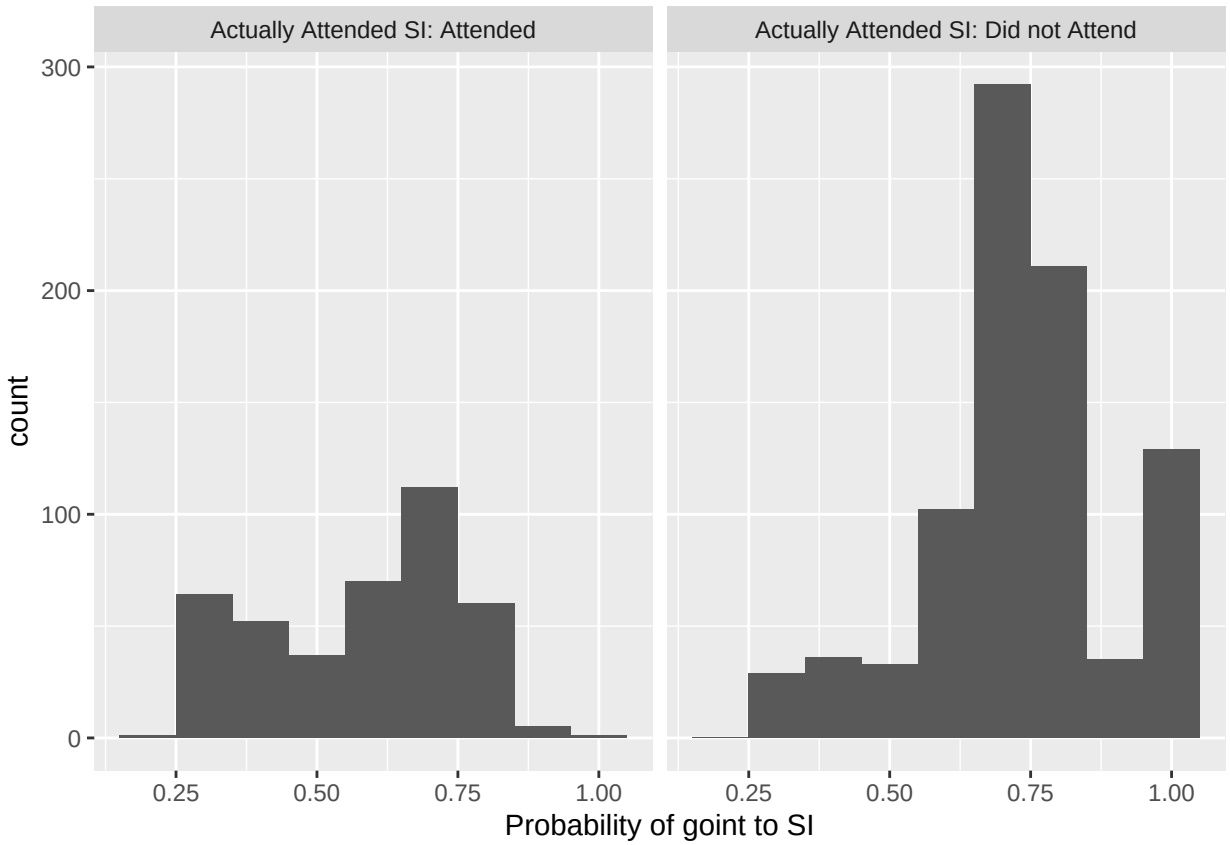
```
##
## Call:
## glm(formula = SI.Attended ~ HS.GPA + Student.Orientation.Flag +
##     Major.1.STEM.Flag + Random.Course.ID, family = binomial,
```

```
##     data = data.fall)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5593  -0.9820   0.6746   0.8392   1.6309
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 4.366e-01  6.397e-01   0.682   0.4949
## HS.GPA                      2.812e-01  1.705e-01   1.649   0.0991 .
## Student.Orientation.FlagY   8.898e-03  1.439e-01   0.062   0.9507
## Major.1.STEM.FlagY         -2.793e-01  1.799e-01  -1.552   0.1207
## Random.Course.ID7490       -7.980e-01  3.296e-01  -2.421   0.0155 *
## Random.Course.ID10788      -1.304e+00  3.341e-01  -3.903 9.49e-05 ***
## Random.Course.ID10792      -1.750e-01  3.610e-01  -0.485   0.6278
## Random.Course.ID10796      -7.345e-01  4.418e-01  -1.662   0.0964 .
## Random.Course.ID11676       7.624e-02  6.145e-01   0.124   0.9013
## Random.Course.ID11783      -6.809e-01  4.040e-01  -1.686   0.0919 .
## Random.Course.ID14054      -2.880e-01  2.955e-01  -0.975   0.3297
## Random.Course.ID15755      -1.819e+00  4.099e-01  -4.437 9.11e-06 ***
## Random.Course.ID17755      -1.990e+00  3.765e-01  -5.286 1.25e-07 ***
## Random.Course.ID18895      -4.936e-01  4.217e-01  -1.170   0.2418
## Random.Course.ID19927       1.631e+01  8.369e+02   0.019   0.9844
## Random.Course.ID19928       1.628e+01  8.224e+02   0.020   0.9842
## Random.Course.ID19929       1.867e+00  1.059e+00   1.763   0.0780 .
## Random.Course.ID21179      -5.288e-01  3.239e-01  -1.633   0.1025
## Random.Course.ID22431       1.630e+01  8.050e+02   0.020   0.9838
## Random.Course.ID22432       1.905e+00  1.061e+00   1.795   0.0726 .
## Random.Course.ID22433       1.905e+00  1.060e+00   1.797   0.0723 .
## Random.Course.ID22599      -1.866e+00  4.651e-01  -4.012 6.03e-05 ***
## Random.Course.ID22824       1.639e+01  2.765e+03   0.006   0.9953
## Random.Course.ID23124      -6.663e-01  5.507e-01  -1.210   0.2264
## Random.Course.ID25606      -8.762e-01  6.536e-01  -1.341   0.1800
## Random.Course.ID265067      2.880e-01  6.186e-01   0.466   0.6415
## Random.Course.ID265068      1.173e+00  7.912e-01   1.483   0.1382
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1584.8  on 1268  degrees of freedom
## Residual deviance: 1370.1  on 1242  degrees of freedom
## AIC: 1424.1
##
## Number of Fisher Scoring iterations: 16
```

```r
data.fall$Prop.Score <- predict(prop.score, type = "response")

labs <- paste("Actually Attended SI:", c("Did not Attend", "Attended"))

# Histogram of Propensity Scores
data.fall %>% mutate(SI.Attended = ifelse(SI.Attended == 1, labs[1], labs[2])) %>%
  ggplot(aes(x = Prop.Score)) +
    geom_histogram(binwidth = .1) + facet_wrap(~SI.Attended) + xlab("Probability of goint to SI")
```




```
# Here we make our matching. The k2K = TRUE will coarsen the covariates first
# and then do an exact matching on the coarsend data. This allows us to compute
# the difference in grade between our matching.
matching <- matchit(SI.Attended ~
                      HS.GPA +
                      Student.Orientation.Flag +
                      Major.1.STEM.Flag +
                      Random.Course.ID,
                    data = data.fall, method = 'cem', estimand = 'ATE',
                    k2k = TRUE,
                    k2k.method = "euclidean")

# Extract Matchings

matched_df1 <- match.data(matching) %>% arrange(subclass, SI.Attended)


# Split data into treatment and control
treatment <- matched_df1 %>% filter(SI.Attended == 1)
control <- matched_df1 %>% filter(SI.Attended == 0)

# Rejoin them so I can determine the grade difference
matched_df2 <- full_join(treatment, control, by = "subclass")

# Treatment - Control ie Positive means SI student did better, Negative means Non SI student did.
matched_df2$Grade.diff <- matched_df2$Grade.Point.per.Unit.center.x - matched_df2$Grade.Point.per.Unit.
```

```
temp <- matched_df2 %>% select(subclass, Grade.diff)

matched_df1 <- matched_df1 %>% left_join(temp)
```

```
## Joining, by = "subclass"
```

```
matched_df1$Grade.diff[matched_df1$SI.Attended == 0] <- -matched_df1$Grade.diff[matched_df1$SI.Attended

model1 <- glm(Grade.diff ~ Prop.Score  +
                IPEDS.Ethnicity.URM.Non.URM, data = matched_df1,
              weights = weights)

coeftest(model1, vcov. = vcovCL,
         cluster = ~subclass)
```

```
##
## z test of coefficients:
##
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     0.123204   0.072783  1.6928  0.09050 .
## Prop.Score                     -0.056891   0.045601 -1.2476  0.21218
## IPEDS.Ethnicity.URM.Non.URMURM -0.212130   0.119627 -1.7733  0.07619 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation

After controlling for other demographic, profile and HS GPA we see the propensity to go to SI is not significantly correlated with the difference in grade between matched pairs, but the difference is .2121 times less for under represented minorities (for a p value of 0.1)