Pset5

2024-01-30

In this code block, we import the data, as well as installing any necessary packages.

```
source("./Functions.R")

install_packages_if_needed(c("utils"))

#Import the csv files
dt_psid <- data.table::as.data.table(utils::read.delim(file = "nswpsid.csv",
                                     sep = ","))
```

Q1

$p_i = E[D_i|x_i] = Pr[D_i = 1|x_i] = \sum_m \theta_m 1_{\{x_i=m\}}$ So what is θ_m? $Pr[D_i = 1|x_i = m]$ of course!

Now what is a good estimator of θ_m? I propose the simple bin estimator:

$\hat{\theta}_m = \sum_i D_i 1_{\{x_i=m\}} / \sum_i 1_{\{x_i=m\}}$.

This is a consistent and unbiased estimator of θ_m, by law of large numbers

In words: it is the number of treated individuals with m years of schooling, divided by the total number of individuals with m years of schooling. This estimator has a problem though: if the sample does not include certain years of schooling, then the estimator cannot tell us anything about $Pr[D_i = 1|x_i = m]$ for those m's.

Hence we get our estimator for $\hat{p}_i$:

$\hat{p}_i = \sum_m \hat{\theta}_m 1_{\{x_i=m\}}$

Q2 a

```
#mutate the data table to add additional variables:
dt_psid <-dt_psid %>%
  dplyr::mutate(.data=dt_psid,
                agesq = age**2,
                edusq = edu**2,
                re74sq = re74**2,
                re75sq = re75**2,
                u74black = u74*black)


#Write the formula:
pscore_formula <- as.formula("treat ~ age + agesq +edu + edusq + married + nodegree + black +
hisp + re74 + re75 + re74sq + re75sq + u74black")

#OLS estimator for theta
lpm <- stats::lm(pscore_formula, data = dt_psid)

#This will be our theta_hat, for LPM
lpm$coefficients
```

```
##   (Intercept)           age         agesq           edu         edusq
##  4.758788e-02 -3.827796e-03  2.161565e-05  3.273680e-02 -1.198170e-03
##       married      nodegree         black          hisp          re74
## -1.344377e-01  6.545316e-02  2.438672e-02  9.121562e-02 -8.770625e-07
##          re75        re74sq        re75sq      u74black
## -2.523223e-06  2.414093e-11  2.152485e-11  5.702562e-01
```

Q2 b

i

Since in LPM, we assume that $Pr[D_i = 1|x_i = x] = x'\theta = x_i'\theta$ (since given $x_i = x$. This really is for readability), we take the derivative with respect to $x_{i,k}$ to see how much change in $x_{i,k}$ would impact p:

$$\begin{aligned}\frac{\partial x'\theta}{\partial x_{i,k}} &= \frac{\partial}{\partial x_{i,k}} \sum_k x_{i,k}\theta_k \\ &= \theta_k\end{aligned}$$

ii

In this case, change in re75 actually changes 2 variables: re75 and re75sq:

$\frac{\partial x'\theta}{\partial re75} = \theta_{re75} + 2\theta_{re75sq} re75$

Using our estimated values, the equation would be: $\frac{\partial \hat{p}}{\partial re75} = -2.5232e(-6) + 4.30497e(-11)re75$

iii

First, what is the mean for re75?

```
mean(dt_psid$re75)
```

```
## [1] 17850.89
```

Now we plug this number into our estimated equation:

```
lpm$coefficients["re75"]+2*lpm$coefficients["re75sq"]*mean(dt_psid$re75)
```

```
##          re75
## -1.754747e-06
```

But we are not quite there yet. This really represents the change in $\hat{p}$ associated with *1* dollar change in re75 around the mean. To get the change in $\hat{p}$ due to $10000 change around the mean, we need to multiply this value by 10000.

So we may say: "Evaluated at average 1975 earnings, a $10,000 increase in 1975 earnings is associated with 1.75 percentage points lower probability of being included in the treated group."

(we can re-integrate the problem to get a more precise value, but linearizing a rather flat quadratic equation seems alright)

Q2 c

```
p_hat = stats::predict(lpm)
summary(p_hat)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.16451 -0.01970  0.01192  0.06916  0.07728  0.86091
```

Q2 d

```
dt_psid <-dt_psid %>%
  dplyr::mutate(.data=dt_psid,
                p_lpm = p_hat)
#this works?! This is magical!!!
```

Q2 e

We see from summary of p_hat that 0 of the predicted value is above 1 (max = 0.86) BUT: we do have negative values:

```
nrow(dt_psid[dt_psid$p_lpm<0,])
```

```
## [1] 1076
```

1076 of the p_hat is negative, under LPM. This is not looking good.

Q3 a

Now we implement the lasso regression:

```
install_packages_if_needed(c('glmnet'))
# Estimate the pscore using Lasso
# Predictor variables
x <- stats::model.matrix(pscore_formula, data = dt_psid)[,-1]
# Outcome variable
y <- dt_psid$treat
# Find the best lambda using cross-validation
set.seed(123)
cv_lasso <- glmnet::cv.glmnet(x, y, alpha = 1)
# Display the best lambda value
cv_lasso$lambda.min
```

```
## [1] 0.0003247039
```

```
# Apply Lasso
lasso_propensity <- glmnet::glmnet(x = x, y = y, alpha = 1,
                                   standardize = TRUE,
                                   lambda = cv_lasso$lambda.min)
#Display Lasso
summary(lasso_propensity)
```

```
##           Length Class     Mode
## a0         1     -none-    numeric
## beta      13     dgCMatrix S4
## df         1     -none-    numeric
## dim        2     -none-    numeric
## lambda     1     -none-    numeric
## dev.ratio  1     -none-    numeric
## nulldev    1     -none-    numeric
## npasses    1     -none-    numeric
## jerr       1     -none-    numeric
## offset     1     -none-    logical
## call       6     -none-    call
## nobs       1     -none-    numeric
```

Q3 b

Now we compare OLS estimated coefficients and Lasso estimated coefficients:

```
#Coefficients estimated by lasso
coefficients(lasso_propensity)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  6.587496e-02
## age         -2.312782e-03
## agesq        .
## edu          2.593927e-02
## edusq       -9.329911e-04
## married     -1.351921e-01
## nodegree     6.124630e-02
## black        2.341096e-02
## hisp         8.828106e-02
## re74        -8.310602e-07
## re75        -2.454234e-06
## re74sq       2.304273e-11
## re75sq       2.023693e-11
## u74black     5.709564e-01
```

```
#Coefficients estimated by OLS
coefficients(lpm)
```

```
##   (Intercept)           age         agesq           edu         edusq
##  4.758788e-02 -3.827796e-03  2.161565e-05  3.273680e-02 -1.198170e-03
##       married      nodegree         black          hisp          re74
## -1.344377e-01  6.545316e-02  2.438672e-02  9.121562e-02 -8.770625e-07
##          re75        re74sq        re75sq      u74black
## -2.523223e-06  2.414093e-11  2.152485e-11  5.702562e-01
```

Note that by Lasso, the coefficients on agesq is estimated to be 0, while in OLS it is estimated to be 2.161565e-05.

In general Lasso estimated coefficients seems to be smaller than OLS estimated coefficients.

Part 2 Q4

By definition, the likelihood function of this example is:

$$\mathcal{L}(\gamma; \tilde{D}) = \prod_{i=1}^{N} \Pr(D_i = \tilde{D}_i | \gamma)$$

But what is $\Pr(D_i = \tilde{D}_i |; \gamma)$ in terms of γ here? Since we are dealing with Bernoulli distribution, we may set $\gamma = \Pr(D_i = 1)$, and $1 - \gamma = \Pr(D_i = 0)$. In term:

$$\Pr(D_i = \tilde{D}_i | \gamma) = \gamma^{\tilde{D}_i} (1 - \gamma)^{1 - \tilde{D}_i}$$

Hence our likelihood function is:

$$\mathcal{L}(\gamma;\tilde{D}) = \prod_{i=1}^{N} \gamma^{\tilde{D}_i}(1-\gamma)^{1-\tilde{D}_i}$$

Now we get the log likelihood function:

$$\ell(\gamma;\tilde{D}) = \sum_{i=1}^{N} \tilde{D}_i ln(\gamma) + (1-\tilde{D}_i) ln(1-\gamma)$$

FOC:

$$\begin{aligned}\frac{\partial}{\partial\gamma}\ell(\gamma;\tilde{D}) &= \sum_{i=1}^{N} \frac{\tilde{D}_i}{\gamma} - \frac{1-\tilde{D}_i}{1-\gamma} \\ &= \sum_{i=1}^{N} \frac{\tilde{D}_i - \gamma}{\gamma(1-\gamma)} \\ &= \frac{1}{\gamma(1-\gamma)}(N \times \gamma - \sum_{i=1}^{N} D_i)\end{aligned}$$

Which is equal to zero when $\gamma = \frac{\sum_i D_i}{N}$. In our case, this would be 0.1

Q5 a

Here we estimate the gamma’s:

```
mle <- stats::glm(pscore_formula, family = binomial( ), data = dt_psid)
summary(mle) #Interesting. It seems MLE is better at fitting the model to OPV (no surprise ac
tually since MLE is biased to fit best).
```

```
##
## Call:
## stats::glm(formula = pscore_formula, family = binomial(), data = dt_psid)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6950  -0.0957  -0.0248  -0.0049   3.8300
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.552e+00  2.452e+00  -3.080 0.002067 **
## age          3.306e-01  1.203e-01   2.747 0.006012 **
## agesq       -6.343e-03  1.856e-03  -3.417 0.000633 ***
## edu          8.248e-01  3.534e-01   2.334 0.019613 *
## edusq       -4.832e-02  1.861e-02  -2.597 0.009410 **
## married     -1.884e+00  2.995e-01  -6.292 3.14e-10 ***
## nodegree     1.300e-01  4.284e-01   0.303 0.761582
## black        1.133e+00  3.521e-01   3.218 0.001292 **
## hisp         1.963e+00  5.674e-01   3.459 0.000541 ***
## re74        -1.047e-04  3.551e-05  -2.948 0.003194 **
## re75        -2.172e-04  4.154e-05  -5.228 1.71e-07 ***
## re74sq       2.358e-09  6.572e-10   3.587 0.000334 ***
## re75sq       1.580e-10  6.671e-10   0.237 0.812716
## u74black     2.137e+00  4.274e-01   5.000 5.72e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1345.30  on 2674  degrees of freedom
## Residual deviance:  409.86  on 2661  degrees of freedom
## AIC: 437.86
##
## Number of Fisher Scoring iterations: 10
```

Q5 b

i

First, we know that:

$$\begin{aligned}\frac{\partial}{\partial x_{i,k}} e^{\mathbf{x}'\gamma} &= \frac{\partial}{\partial x_{i,k}} \prod_{j=1}^{K} e^{\gamma_j x_{i,j}} \\ &= \gamma_k e^{\mathbf{x}'\gamma}\end{aligned}$$

Hence:

$$\frac{\partial}{\partial x_{i,k}}(\frac{e^{\mathbf{x}'\gamma}}{1+e^{\mathbf{x}'\gamma}}) = \frac{\gamma_k e^{\mathbf{x}'\gamma}(1+e^{\mathbf{x}'\gamma}) - \gamma_k e^{\mathbf{x}'\gamma}e^{\mathbf{x}'\gamma}}{(1+e^{\mathbf{x}'\gamma})^2}$$
$$= \gamma_k \frac{e^{\mathbf{x}'\gamma}}{(1+e^{\mathbf{x}'\gamma})^2}$$

ii

We can generalize this problem to finding $\frac{e^f(x)}{1+e^f(x)}$:

$$\frac{\partial}{\partial x}(\frac{e^f(x)}{1+e^f(x)}) = \frac{f'(x)e^{f(x)}(1+e^{f(x)}) - e^{f(x)}f'(x)e^{f(x)}}{(1+e^{f(x)})^2}$$
$$= f'(x)\frac{e^{f(x)}}{(1+e^{f(x)})^2}$$

It follows then that:

$$\frac{\partial}{\partial re75}(\frac{e^{x'\gamma}}{1+e^{x'\gamma}}) = (\gamma_{re75} + 2\gamma_{re75sq}re75)\frac{e^{x'\gamma}}{(1+e^{x'\gamma})^2}$$

Q6

Now we use MLE to predict the p score:

```
p_logit <- stats::predict(mle, type = "response")
summary(p_logit)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0000341 0.0006388 0.0691589 0.0109155 0.9748754
```

Q7

Mutating …

```
dt_psid <-dt_psid %>%
  dplyr::mutate(.data=dt_psid,
                p_logit = p_logit)

#Store for future use
write.csv(dt_psid,"nswpsid_logit.csv", row.names = FALSE)
```

Q8

Why is Logit score always between 0 and 1? Consider the limits:

$$\lim_{\mathbf{x}'\gamma\to\infty}\frac{e^{\mathbf{x}'\gamma}}{1+e^{\mathbf{x}'\gamma}} = \lim_{\mathbf{x}'\gamma\to\infty}\frac{e^{\mathbf{x}'\gamma}}{e^{\mathbf{x}'\gamma}} = 1 \text{ (by l'H}\hat{o}\text{pital)}$$
$$\lim_{\mathbf{x}'\gamma\to-\infty}\frac{e^{\mathbf{x}'\gamma}}{1+e^{\mathbf{x}'\gamma}} = 0$$

And this function is increasing in $\mathbf{x}'\gamma$, as $\frac{\partial}{\partial x}\frac{e^x}{1+e^x} = \frac{e^x}{(1+e^x)^2} \geq 0$ for all x.