ARE213 Problem Set #1B

Peter Alstone & Frank Proulx

October 16, 2013

1 Problem #1

1.1 Part A

Under the assumption of random assignment conditional on the observables, what are the sources of misspecification bias in the estimates generated by the linear model estimated in Problem Set 1a?

Wrong functional form. In Problem Set 1A we used linear (i.e., $y = \beta x + \epsilon$) estimators to make "corrections" while the true functional form of the relationships between the covariates we included in the modern were certainly not linear. By imposing a linear function on a non-linear data generating process (described by the CEF), we introduce misspecification bias in the model.

Omitted Variables Bias. We were able to use variables included in the dataset in our linear model, but not the unobserved variables that may be important for control. If omitted variables exist that both determine outcomes related to birth weight and are correlated with smoking status we will over- or under-estimate the effect (depending on the characteristics of the omission).

1.2 Part B

Now, consider a series estimator. Estimate the smoking effects using a flexible functional form for the control variables (e.g., higher order terms and interactions). What are the benefits and drawbacks to this approach?

We can attempt to reduce the magnitude of the first source of bias mentioned above (functional form) by introducing non-parametric series estimators as a replacement for linear regression. To implement this we used a natural cubic spline with two knots on the “dmar” variable (maternal age) in the regression from PS1A. The tobacco use (treatment) and marital status remain as factors. We also implemented a version of the model with interactions between the splined maternal age term and the two discrete terms. The summary of the results are in Table 1. The ATE for the model we used in PS1A for tobacco use was 200 grams (rounded from an exact estimate of 195 grams). This is essentially unchanged with the addition splines to the maternal age relationship (an exact estimate of 199 grams). Adding interaction terms results in an ATE for tobacco use of 220 grams.

The benefits to applying splines in this case is that the regression model more closely matches the reality of the data, which show that birth weight’s relationship to maternal age has a peak and is not monotonically increasing. The drawback is that the true functional form is only obscured in this approach. While the interaction terms result in an ATE that is different from the one in a non-interacting model, the interpretation becomes much more difficult. In a policymaking environment the addition of splines and interactions would represent a potential roadblock to the essential message, which remains unchanged, which is that birth weight is reduced in mothers who use tobacco (by about 200 grams).

2 Problem #2

The Propensity Score Method (PSM) uses a “surrogate” normalized metric (p-score) as a replacement for the observable controls that would normally be used to condition the estimates of a treatment response to the variable in question. The p-score is defined as a normalized score that represents the likelihood a sample selects into treatment conditioned on observables. Because it collapses all the dimensions into a 0:1 continuum PSM avoids the curse of dimensionality encountered with large nonparametric regression models, where it can be difficult to find neighbors or “bandwidth-mates” in n-dimensional space.

Table 1: Comparison of three linear models for birth weight

	Birth Weight		
	1) PS1A LM	2) 1+spline	3) 2+interaction
tobacco2	195.101***	199.060***	252.395***
	(4.764)	(4.774)	(84.171)
dmage	2.716***		
	(0.338)		
ns(dmage, df = 3)1		147.297***	85.607**
		(10.062)	(37.532)
ns(dmage, df = 3)2		342.494***	386.881**
		(36.880)	(177.619)
ns(dmage, df = 3)3		−10.419	−55.186
		(25.826)	(104.185)
dmar2	−146.507***	−123.962***	57.854
	(4.538)	(4.797)	(85.517)
tobacco2:ns(dmage, df = 3)1			101.145**
			(41.432)
tobacco2:ns(dmage, df = 3)2			−68.050
			(194.021)
tobacco2:ns(dmage, df = 3)3			33.670
			(109.995)
tobacco2:dmar2			−267.716***
			(94.824)
ns(dmage, df = 3)1:dmar2			−135.516***
			(50.824)
ns(dmage, df = 3)2:dmar2			−398.948*
			(214.796)
ns(dmage, df = 3)3:dmar2			−275.447*
			(165.568)
tobacco2:ns(dmage, df = 3)1:dmar2			42.062
			(59.539)
tobacco2:ns(dmage, df = 3)2:dmar2			619.892***
			(239.750)
tobacco2:ns(dmage, df = 3)3:dmar2			391.482**
			(192.669)
Constant	3,170.698***	3,040.520***	2,989.989***
	(10.921)	(16.392)	(76.157)
N	114,610	114,610	114,610
R^2	0.037	0.039	0.040
Adjusted R^2	0.037	0.039	0.040

Notes:

***Significant at the 1 percent level.
**Significant at the 5 percent level.
*Significant at the 10 percent level.

2.1 Part A

To calculate the propensity score, we estimated a logit model of mother's tobacco use (0=non-smoker, 1=smoker) as determined by the predetermined covariates shown in Table 2. Model #1 shows the full model using all of the covariates suspected to be predetermined. Model #2 is a reduced form of the same model, preserving just those covariates that were significant at the 1% level in Model #1.

To test whether the propensity scores predicted by these two models are significantly different we perform a likelihood ratio test between the full and reduced model. This test yields the following output:

Likelihood ratio test for MLE method: Chi-squared 3 d.f. = 10.21274, P value = 0.01684173

NOTE from class: including insignificant terms can be beneficial in terms of getting better fit (by reducing

2.2 Part B

This estimation assumes unconfoundedness. and conditional independence

This suggests that (pursuant to these assumptions) the average effect of smoking on birthweight is a reduction of 223 grams.

2.3 Part C

We estimate the average treatment effect of smoking on birthweight to be 222 grams when using the reweighting approach. This approach involved taking a weighted average of all observations using the (normalized inverse) of the propensity score as a weighting factor. This is consistent with the estimate produced using the regression approach.

We estimate the average treatment on the treated to be

2.4 Part D

Here we estimate the density function with a kernel density estimator, using the density() function in R. We estimate the density function separately for the smoking and non-smoking members of the sample, and weight

Table 2: Logistic function coefficients for propensity score models

	Mother Tobacco-Use Status	
	(1)	(2)
Mother's Race not White or Black	−1.956***	−1.954***
	(0.134)	(0.133)
Mother's Years of Education	−0.817***	−0.818***
	(0.028)	(0.028)
Marital status	−0.205***	−0.204***
	(0.005)	(0.005)
Father's age	−1.256***	−1.251***
	(0.022)	(0.021)
Father's Years of Education	0.029***	0.030***
	(0.002)	(0.001)
Father Mexican	−0.131***	−0.131***
	(0.005)	(0.005)
Father Puerto Rican	−1.961***	−1.957***
	(0.173)	(0.173)
Father Cuban	−1.267***	−1.268***
	(0.058)	(0.058)
Father Central or South American	−0.567	−0.567
	(0.364)	(0.364)
Father Race Other or Unknown Hispanic	−1.933***	−1.932***
	(0.205)	(0.205)
Plurality of Infant	−0.890***	−0.889***
	(0.120)	(0.120)
Sex of Infant	−0.148***	
	(0.054)	
Mother's age	−0.019	
	(0.017)	
dmage	0.003	
	(0.002)	

their responses with the propensity scores normalized to the subsample (e.g. $p(X_i)/\sum j = 1^{N_{smokers}} p(X_j))$

We estimate the density function using the Epanechnikov kernel and bandwidths ranging from 15 grams to 50 grams in increments of 5 grams. Figure 1 shows the density function estimated with a bandwidth of 40 grams. This bandwidth appears to be a good compromise between washing out some of the noise at lower bandwidths while preserving the underlying CEF.

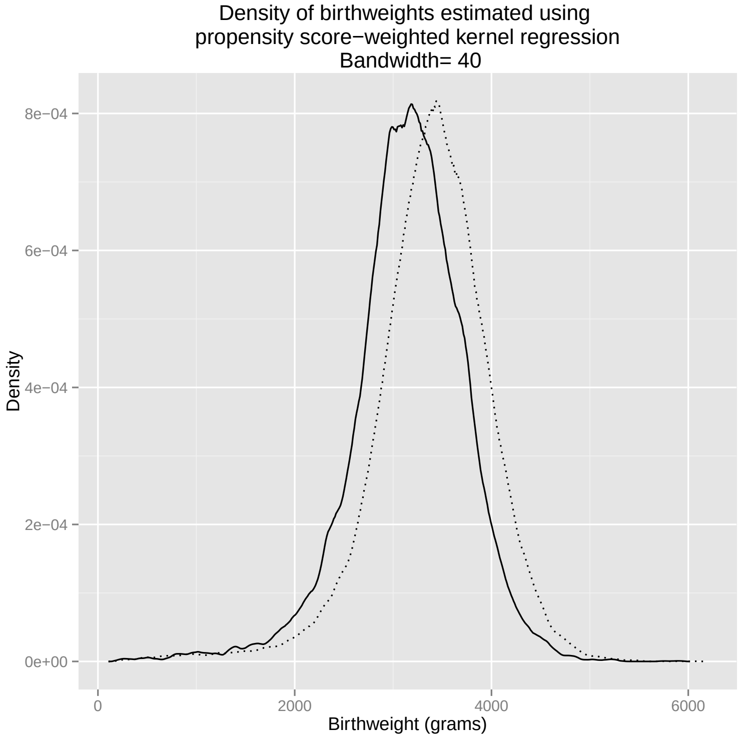

Figure 1: Birthweight density function estimates produced using Epanechnikov kernel estimator for smokers and non-smokers.

We also estimate the density at x=3000 grams by hand. We first calculate the average propensity score for both smokers and non-smokers with infants at 3000 grams. The dataset has 12 smokers and 47 non-smokers with infants born at this weight. The average propensity score for these smokers is 0.27 and for non-smokers it is 0.20. These values stand to reason- the people who have opted in to smoking in this class have been predicted as more likely to be smokers.

3 Code

We used R to complete this assignment. The code is below:

```
# PROBLEM SET 1B
# ARE 213 Fall 2013
## TO DO:
# Figure out ATT in 2c
# Get a legend on the kerndensity plots to make the "beautiful and publication ready"
# Figure the kernel regression by hand problem
# problem 5
# writing up a number of problems (I will get more done on this in the morning - need to go take a midterm).

# Frank’s Directory
#setwd("/media/frank/Data/documents/school/berkeley/fall13/are213/are213/ps1")

# Peter’s Directory
#setwd("~/Google Drive/ERG/Classes/ARE213/are213/ps1")


# Packages --------
library(foreign) #this is to read in Stata data
library(Hmisc)
library(psych)
library(stargazer)
library(ggplot2) # for neato plotting tools
library(plyr) # for nice data tools like ddply
library(epicalc) # For likelihood ratio test
library(car) # "companion for applied regression" - recode fxn, etc.
library(gmodels) #for Crosstabs
library(splines) # for series regression
library(np) #nonparametric regression
library(rms) #regression modeling tools
library(effects)

# Homebrewed functions
source("../util/are213-func.R")
source("../util/watercolor.R") # for watercolor plots

# Data -------
ps1.data <- read.dta(file="ps1.dta")

var.labels <- attr(ps1.data, "var.labels")

# Data Cleaning Step
full.record.flag <- which(ps1.data$cardiac != 9 &
                            ps1.data$cardiac != 8 &
                            ps1.data$lung != 9 &
                            ps1.data$lung != 8 &
                            ps1.data$diabetes !=9 &
                            ps1.data$diabetes !=8 &
                            ps1.data$herpes != 9 &
                            ps1.data$herpes != 8 &
                            ps1.data$chyper != 9 &
                            ps1.data$chyper != 8 &
                            ps1.data$phyper != 9 &
                            ps1.data$phyper != 8 &
                            ps1.data$pre4000 !=9 &
                            ps1.data$pre4000 !=8 &
                            ps1.data$preterm != 9 &
                            ps1.data$preterm != 8 &
                            ps1.data$tobacco != 9 &
                            ps1.data$cigar != 99 &
                            ps1.data$cigar6 !=6 &
                            ps1.data$alcohol != 9 &
                            ps1.data$drink != 99 &
                            ps1.data$drink5 !=5 &
                            ps1.data$wgain !=99
)

ps1.data$full.record <- FALSE # initialize column as F
ps1.data$full.record[full.record.flag] <- TRUE #reassign level to T for full records

ps1.data.clean <- subset (ps1.data, full.record == TRUE)
ps1.data.missingvalues <- subset(ps1.data, full.record == FALSE)

# Problem 1a : Describes PS1a results. -------
# Problem 1b -------------
# This is using a series estimator. I think smooth.spline() is the right function to use, but let me know if you think we should be doing kernel regression instead. I'm also not sure how to go about adding interaction terms.  I think a kernel regression is more appropriate here...mostly because I don't know the spline function and there seems to be a good package ("np") for running kernel regression.

# SPLINE FIT FOR # CIGS

# sm.flex <- with(ps1.data.clean, smooth.spline(cigar, y=dbrwt, nknots=10, spar = 0.7, tol = 0.0001)) # Fits a smooth line to the data
# sm.flex.df <- data.frame(sm.flex$x, sm.flex$y) #converts the fitted values into a data frame for ggplot
#
# splineplot <- ggplot(sm.flex.df, aes(x = sm.flex.x, y=sm.flex.y))
# splineplot <- splineplot +
#   geom_point(data=ps1.data.clean, aes(x = cigar, y = dbrwt), pch = 1) +
#   geom_line(color='red') +
#   labs(x = 'Cigarettes smoked per day by mother', y= 'Birthweight')
# splineplot
#
# ggsave(filename = 'img/splineplot.pdf')

# Using Series estimator with splines on maternal age.

ps1.data.clean$tobacco <- as.factor(ps1.data.clean$tobacco)
ps1.data.clean$dmar <- as.factor(ps1.data.clean$dmar)
```

```r

wsp.ps1a <- lm(dbrwt ~ tobacco + dmage + dmar, data=ps1.data.clean)
wsp <- lm(dbrwt ~ tobacco + ns(dmage, df=3) + dmar, data=ps1.data.clean)
wsp.int <- lm(dbrwt ~ tobacco * ns(dmage, df=3) * dmar, data=ps1.data.clean)

stargazer(wsp.ps1a, wsp, wsp.int, style="qje", no.space = TRUE, dep.var.labels = "Birth Weight")



# This is the ATE with a splines regression on Age
summary(effect("tobacco", wsp))

# This is the ATE with a complex, interacting splines regression on AGe
summary(effect("tobacco", wsp.int))

# Problem 2a --------
ps1.data.clean$tobacco.rescale <- with(ps1.data.clean, recode(tobacco, "2='0'", as.numeric.result=TRUE)) #rescales the tobacco use variable to be 0/1, where 0=no and 1 = yes
ps1.data.clean$dmar.rescale <- with(ps1.data.clean, recode(dmar, "2='0'"))

smoke.propensity.all <- glm(tobacco.rescale ~ as.factor(mrace3) + dmeduc + dmar.rescale + dfage + dfeduc + as.factor(orfath) + dplural + csex + dmage, data=ps1.data.clean, family = binomial()) ## Did I miss any predetermined covariates here? No.

smoke.propensity.reduced <- glm(tobacco.rescale ~ as.factor(mrace3) + dmeduc + dmar.rescale + dfage + dfeduc + as.factor(orfath), data=ps1.data.clean, family = binomial())


stargazer(smoke.propensity.all, smoke.propensity.reduced,
           type = "latex",
           covariate.labels = c("Mother's Race not White or Black", "Mother's Years of Education", "Marital status", "Father's age", "Father's Years of Education", "Father Mexican", "Father Puerto Rican", "Father Cuban", "Father Central or South American", "Father Race Other or Unknown Hispanic", "Plurality of Infant", "Sex of Infant", "Mother's age"),
          style ="qje",
          align = TRUE,
          label = "tab:propensities",
          title = "Logistic function coefficients for propensity score models",
          dep.var.labels = "Mother Tobacco-Use Status",
          out = "propensityscores.tex"
           )

ps1.data.clean$propensityfull <- predict(smoke.propensity.all, type = "response")
ps1.data.clean$propensityreduced <- predict(smoke.propensity.reduced, type = "response")

detach("package:rms")
sink(file = "lrtest.tex", append = FALSE)
lrtest(smoke.propensity.all, smoke.propensity.reduced) #Test whether the two
```

```
scores are statistically different
sink()

require(rms)

#Problem 2b - Estimating a regression model using propensity scores --------

sm.propensityregression <- lm(dbrwt ~ propensityreduced * tobacco.rescale, ps1.data.clean)

#calculation of average treatment effect:
coefficients(sm.propensityregression)[2] + coefficients(sm.propensityregression)[4]*mean(ps1.data.clean$propensityreduced)

tobacco.effects <- (effect("tobacco.rescale", sm.propensityregression))

stargazer(sm.propensityregression,
          type = "latex",
          covariate.labels = c("Delta1", "Beta", "Delta2", "Constant"),
          style ="qje",
          align = TRUE,
          font.size="footnotesize",
          label = "tab:propensitymodel",
          title = "Model of effects of tobacco use on birthweight using propensity score as a control",
          dep.var.labels = "Mother Tobacco-Use Status",
          out = "propensityscoremodel.tex"
)

#Problem 2c - Using reweighting with propensity scores --------

ps1.data.clean$tobacco.rescale.n <- as.numeric(levels(ps1.data.clean$tobacco.rescale))[ps1.data.clean$tobacco.rescale]

term1 <- with(ps1.data.clean, sum((tobacco.rescale.n*dbrwt)/propensityreduced)/sum(tobacco.rescale.n/propensityreduced))
term2 <- with(ps1.data.clean, sum(((1-tobacco.rescale.n)*dbrwt)/(1-propensityreduced))/sum((1-tobacco.rescale.n)/(1-propensityreduced)))

weightingestimator <- term1-term2 #This should be the average treatment effect

term1.T <- with(subset(ps1.data.clean, tobacco.rescale.n=1), sum((tobacco.rescale.n*dbrwt)/propensityreduced)/sum(tobacco.rescale.n/propensityreduced))
#term2.T <- with(subset(ps1.data.clean, tobacco.rescale=1), sum(((1-tobacco.rescale)*dbrwt)/(1-propensityreduced))/sum((1-tobacco.rescale)/(1-propensityreduced)))

weightingestimator.T <- term1.T#-term2.T #This should be the average treatment on treated


# Problem 2d - Kernel Density Estimator
tot.propensity.nosm <- with(subset(ps1.data.clean, tobacco.rescale == 0), sum(propensityreduced))
tot.propensity.sm <- with(subset(ps1.data.clean, tobacco.rescale == 1), sum(
     propensityreduced))

kerndensity.plot.fn <- function(h){
kerndensity.nosm <- with(subset(ps1.data.clean, tobacco.rescale == 0),
                         density(dbrwt, #if nobody smoked
                                  kernel = "epanechnikov",
                                  bw = h,
                                  weights = propensityreduced/tot.propensity.nosm))
kerndensity.nosm.df <- data.frame(kerndensity.nosm[1], kerndensity.nosm[2])

kerndensity.sm <- with(subset(ps1.data.clean, tobacco.rescale == 1),
                       density(dbrwt, #if everybody smoked
                                kernel = "epanechnikov",
                                bw = h,
                               weights = propensityreduced/tot.propensity.sm))
kerndensity.sm.df <- data.frame(kerndensity.sm[1], kerndensity.sm[2])

kerndensity.plot <- ggplot(kerndensity.nosm.df, aes(x, y))
kerndensity.plot <- kerndensity.plot +
  geom_line(linetype = 'dotted') +
  geom_line(data = kerndensity.sm.df, aes(x, y)) +
  labs(title = paste("Density of birthweights estimated using \n propensity score-weighted kernel regression \n Bandwidth=", as.factor(h)), x = "Birthweight (grams)", y = "Density") +
  guides(linetype = "Legend") # Having trouble getting a legend.

kerndensity.plot

ggsave(file = paste0('img/kerndensity', h,'.pdf'), plot = kerndensity.plot)}

##Problem 2d - calculating kernel value by 'hand' at dbrwt = 3000 --------
##I can't figure out what to do here. Most of this is probably wrong but maybe something is right. Want to take a whack?
##h <- 30
##kernel.epa <- function(u){
##return(0.75*(1-u*u))}

##propensity3000.sm <- with(ps1.data.clean, mean(propensityreduced[which(dbrwt == 3000 & tobacco.rescale == 1)]))
##propensity3000.nosm <- with(ps1.data.clean, mean(propensityreduced[which(dbrwt == 3000 & tobacco.rescale == 0)]))
##for(i in 1:nrow(subset(ps1.data.clean, tobacco.rescale == 1))){
##with(subset(ps1.data.clean, tobacco.rescale == 1),
##     kern3000.sm.num <- kern3000.sm.num +
##      kernel.epa(((propensity3000.sm-propensityreduced[i])/h)*dbrwt))
## with(subset(ps1.data.clean, tobacco.rescale == 1),
##      kern3000.sm.den <- kern3000.sm.den +
##      kernel.epa((propensity3000.sm-propensityreduced[i])/h))
## }
## kern3000.sm <- kern3000.sm.num / kern3000.sm.den


## kernel3000.sm <- with(subset(ps1.data.clean,tobacco.rescale == 1), data.frame(window = (3000 - dbrwt/h)))
## kernel3000.sm$numerator <- with(subset(ps1.data.clean, tobacco.rescale ==
1), kernel.epa(((3000/propensity3000.sm) - (dbrwt/propensityreduced))/h))
## kernel3000.sm$denominator <- with(subset(ps1.data.clean, tobacco.rescale == 1), kernel.epa(((3000/propensity3000.sm) - (dbrwt/propensityreduced))/h))

## with(kernel3000.sm[window < 1 & window > -1], sum(numerator))/(nrow(kernel3000.sm[abs(window < 1)])*h)

#Problem 2e --------
for(h in seq(from = 15, to = 50, by = 5)){
kerndensity.plot.fn(h)} # This should make plots of the kernel density function for bandwidths ranging from 15 to 40 by 5. Feel free to adjust these values



### Problem 3
## Using blocking estimator
# Divide smokers into ~100 equally spaced blocks
prop.max <- with(ps1.data.clean, max(propensityreduced))
prop.min <- with(ps1.data.clean, min(propensityreduced))
prop.binsize <- (prop.max - prop.min)/99

ps1.data.clean$blocknumber <- with(ps1.data.clean,
                                   round(propensityreduced/prop.binsize, digits = 0) + 1)

blocktreatmenteffects <- ddply(ps1.data.clean, .(blocknumber), summarize, smokers = sum(tobacco.rescale == 1), nonsmokers = sum(tobacco.rescale == 0), smokerdbrwt = mean(dbrwt[tobacco.rescale == 1]), nonsmokerdbrwt = mean(dbrwt[tobacco.rescale == 0]))

blocktreatmenteffects$badbin <- with(blocktreatmenteffects, as.numeric(smokers == 0 | nonsmokers == 0))

cleaned.blocks <- subset(blocktreatmenteffects, badbin == 0)
cleaned.blocks$avgtreatmenteffect <- with(cleaned.blocks, smokerdbrwt - nonsmokerdbrwt)
cleaned.blocks$weight <- with(cleaned.blocks, (smokers + nonsmokers)/sum(smokers + nonsmokers))
cleaned.blocks$weightedTE <- with(cleaned.blocks, weight * avgtreatmenteffect)

blocksATE <- sum(cleaned.blocks$weightedTE)

### Problem 4
ps1.data.clean$lowbrwt <- as.numeric(ps1.data.clean$dbrwt < 2500)

blocklowbrwt <- ddply(ps1.data.clean, .(blocknumber), summarize, smokers = sum(tobacco.rescale == 1), nonsmokers = sum(tobacco.rescale == 0), lowbrwtprob.sm = mean(lowbrwt[tobacco.rescale == 1]), lowbrwtprob.nosm = mean(lowbrwt[tobacco.rescale == 0]))

blocklowbrwt$badbin <- with(blocklowbrwt, as.numeric(smokers == 0 | nonsmokers == 0))

```