Non-parametric Regression and Selection on Observables Designs

Joel Ferguson

10/4/2021

Why go non-parametric? And why now?

You may be wondering why we just spent a week learning about non-parametric regression methods. The answer lies in the way we motivated the regression adjustment design. Recall that if $D_i \perp \{Y_i(d)\}_D | X$ and treatment effects are homogenous, then OLS estimation of

$$Y_i = \alpha + \beta D_i + \gamma \mathbb{E}[D_i|X_i] + \varepsilon_i$$

yields a consistent estimate of the ATE. In practice, we never really know $\mathbb{E}[D_i|X_i]$ unless we assigned treatment via an experiment. However, non-parametric regression methods give us a way to approximate it well. To see this point in action, let's estimate the ATE from a DGP similar to one considered last section, but instead of approximating $\mathbb{E}[D_i|X_i]$ with a linear function, as we implicitly did last time, we can explicitly approximate it non-parametrically and include the prediction as a regressor.

```
epi_k <- function(u,h){ # Epanechnikov Kernel with bandwidth h
  out <- (3/4)*(1-(u/h)^2)*as.numeric(abs(u/h)<1)
  return(out)
}

k_dens <- function(x,X_emp,h){ # Kernel density estimate with data X_emp
  f_hat <- 1/(length(X_emp)*h)*sum(sapply(x-X_emp, epi_k, h=h))
  return(f_hat)
}

k_reg <- function(x,X_emp,y_emp,h){ # Kernel regression
  f_hat <- 1/(length(X_emp)*h)*sum(sapply(x-X_emp, epi_k, h=h)*y_emp)/k_dens(x,X_emp,h)
  return(f_hat)
}

set.seed(92021)
N <- 1000
X_i <- rnorm(N) # One continuous covariate drawn from ~N(0,1)
# Find min and max of X
minX <- min(X_i)
maxX <- max(X_i)

# Draw untreated POs from ~N(X,1), Y(1)=Y(0)+0.3
Y_i0 <- sapply(X_i,function(x) rnorm(1,x,1))
Y_i1 <- Y_i0+.3
  #sapply(Y_i0,function(x) rnorm(1,10+x))
```

```r
print(paste("ATE:",mean(Y_i1-Y_i0)))
```

```
## [1] "ATE: 0.3"
```

```r
# F'n to estimate ATE given a treatment vector
ATE_linreg <- function(D){
  df <- data.frame("X"=X_i,
                   "Y0"=Y_i0,
                   "Y1"=Y_i1,
                   "D"=D,
                   "Y"=Y_i1*D+(1-D)*Y_i0)
  ATE_lin <- lm(Y~D+X,df) # Reg Y on X and D
  return(summary(ATE_lin))
}
# F'n to plot the prob of treatment by X
plotD <- function(Dfn){ # Dfn is a function which assigns treatment prob to X
  D <- sapply(X_i,Dfn) # Find treatment Probs
  df <-  data.frame("X"=X_i,
                   "D"=as.numeric(D)) # Make a DF
  out <- ggplot(data=df)+
    geom_line(aes(x=X,y=D))
  return(out)
}

Dfn_exp <- function(x){
  return(exp(x)/exp(maxX))
}
D_exp <- sapply(X_i, function(x) rbernoulli(1,Dfn_exp(x)))
ED <- sapply(X_i, Dfn_exp)

# Naive ATE estimate
ate_naive <- mean(Y_i1[D_exp==1])-mean(Y_i0[D_exp==0])
print(paste("Naive estimate of ATE:",ate_naive))
```

```
## [1] "Naive estimate of ATE: 1.51604870784317"
```

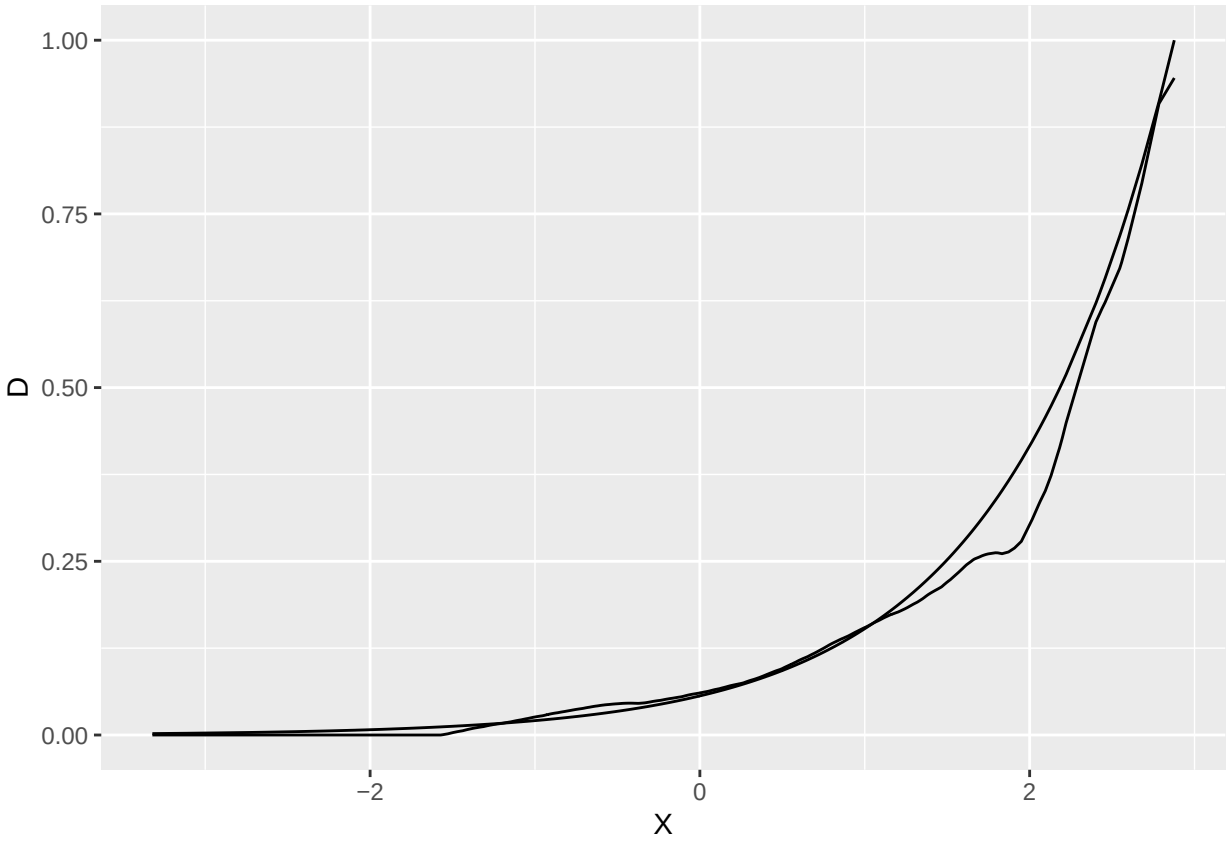
```r
ate_lin_est <- ATE_linreg(D_exp)
print(paste("OLS Regression adjustment estimate of ATE:",ate_lin_est$coefficients[2]))
```

```
## [1] "OLS Regression adjustment estimate of ATE: 0.517715564475442"
```

```r
# Now estimate E[D|X] non-parametrically
D_hat <- sapply(X_i,k_reg,X_emp=X_i, y_emp = D_exp,h=0.6)

# Plot the estimate against the truth
df <- data.frame("Y"=Y_i1*D_exp+(1-D_exp)*Y_i0,
                 "X"=X_i,
                 "D"=D_exp,
                 "D_hat"=D_hat,
                 "ED"=ED)
D_hat_gr <- plotD(Dfn_exp)+
```

```
  geom_line(data=df,mapping=aes(x=X,y=D_hat))
D_hat_gr
```




```
# Estimate the regression with D_hat as a control

np_reg <- lm(Y~D+D_hat,
             data=df)
print(paste("OLS estimate of ATE with non-parametric estimate of p-score:",np_reg$coefficients[2]))
```

```
## [1] "OLS estimate of ATE with non-parametric estimate of p-score: 0.484246951977213"
```

```
# Finally, include the true p-score as a control
pscore_reg <- lm(Y~D+ED,
                 data=df)
print(paste("OLS P-score adjustment estimate of ATE:",np_reg$coefficients[2]))
```

```
## [1] "OLS P-score adjustment estimate of ATE: 0.484246951977213"
```

In this case, controlling for our non-parametric estimate of $\mathbb{E}[D|X]$ gives a slightly better estimate than OLS, but not by much. Even though we know $\hat{\beta}$ is estimated consistently, we run into the issue of covariate overlap in practice. For very low values of X, we don't observe the proper proportion of treated observations (the proportion determined by $\mathbb{E}[D|X]$), biasing our estimate.

There is a simple non-parametric estimator we can use, however, that may perform better. Since treatment effects are homogenous, we can estimate them at any point in the X distribution by comparing outcomes of

units with different treatment status, but close values of X. In this example, this is equivalent to a method called *Propensity Score Binning*, where rather than making bins in the X distribution, we make them in the distribution of $\widehat{\mathbb{E}[D|X]}$.

```
# Make 10 bins in the X distribution
Y <- Y_i1*D_exp+(1-D_exp)*Y_i0
bin_ate_est <- function(x){
  t1 <- mean(Y[(D_exp==1)&(X_i>=bin_ends[x-1])&(X_i<bin_ends[x])])
  t0 <- mean(Y[(D_exp==0)&(X_i>=bin_ends[x-1])&(X_i<bin_ends[x])])
  return(t1-t0)
}
bin_ns <- function(x){
  n <- length(Y[(X_i>=bin_ends[x-1])&(X_i<bin_ends[x])])
  return(n)
}
bin_ends <- seq(minX,maxX,length.out = 11)
bin_ests <- sapply(c(2:11), bin_ate_est)
ns <- sapply(c(2:11), bin_ns)
ate_est <- sum(bin_ests[!is.na(bin_ests)]*ns[!is.na(bin_ests)])/(sum(ns[!is.na(bin_ests)]))
print(paste("P-score binning ATE estimate:",ate_est))
```

```
## [1] "P-score binning ATE estimate: 0.419285466282934"
```

While we're still overestimating the ATE, there's a little improvement over the OLS-based method above. It's also quite easy to understand why this method works. We know that under our assumptions

$$\mathbb{E}[Y_i(1)] \approx \mathbb{E}[Y_j(0)] \; if \; X_i \approx X_j$$

since uncounfoundedness implies $\mathbb{E}[Y_i(0)|X = x] = \mathbb{E}[Y_i(1)|X = x]$. Thus, as long as untreated potential outcomes don't change too quickly over the width of a bin and we have enough observations in each bin to average out the error terms, we should get good estimates of $\mathbb{E}[Y_i(0)|X \approx x]$ and $\mathbb{E}[Y_i(1)|X \approx x]$ in each bin.

As an aside, in the particular example above, we can exactly derive how well this approximation does (in expectation) in a given bin $b_k = [b_k^-, b_k^+]$:

$$\begin{aligned}\mathbb{E}[Y|D=1, X \in b_k] - \mathbb{E}[Y|D=0, X \in b_k] &= \frac{\int_{b_k^-}^{b_k^+}(x+0.3)\phi(x)(e^x/e^{max\ X})dx}{\int_{b_k^-}^{b_k^+}\phi(x)(e^x/e^{max\ X})dx} - \frac{\int_{b_k^-}^{b_k^+}x\phi(x)(1-e^x/e^{max\ X})dx}{\int_{b_k^-}^{b_k^+}\phi(x)(1-e^x/e^{max\ X})dx} \\ &= 0.3 + \frac{\mathbb{P}(X \in b_k)}{\mathbb{P}(X \in b_k \wedge D=1)\mathbb{P}(X \in b_k \wedge D=0)}\mathbb{E}[X|X \in b_k \wedge D=1] - \frac{\mathbb{E}[X|X}{\mathbb{P}(X \in b_k \wedge}\end{aligned}$$

We know that $\mathbb{E}[X|X \in b_k \wedge D=1] > \mathbb{E}[X|X \in b_k]$ since $\mathbb{P}(D=1)$ is increasing in X and $\frac{\mathbb{P}(X \in b_k)}{\mathbb{P}(X \in b_k \wedge D=1)} > 1$, so the estimate is upward biased. This should accord with our intuition, units in the right end of the bin have greater untreated potential outcomes than units at the left end of the bin, who are more likely to serve as controls for the units on the right end of the bin.

More importantly, these examples shows that with strong unconfoundedness, we aren't limited to estimating the ATE via OLS, and even if we do want to use it in the end, we may want to non-parametrically estimate something before running OLS. Conditional on observables designs are all about approximating $\mathbb{E}[Y|D, X]$ well, and for that reason we might be willing to introduce a little variance in our estimation in exchange for a reduction in bias by estimating some or all of the CEF with a more flexible procedure than OLS.