Parametric Regression for Effect Estimation

Ashley I Naimi

November 2023

Contents

1 Introduction to Parametric Regression

Regression is a cornerstone tool of any empirical analysis. It is arguably the most widely used tool in science. Regression models are often deployed in an attempt to understand cause-effect relations between exposures and outcomes of interest, or to obtain predictions for an outcome of interest. Consider a scenario in which we are interested in regressing an outcome Y against a set of covariates X. These covariates can be an exposure combined with a set of confounders needed for identification, or a set of predictors used to create a prediction algorithm via regression. In its most basic (?) formulation, a regression model can be written as[1]:

[1] Note that, in this formulation, the function of interest on the left hand side of the equation is the conditional mean function, $E(Y \mid X)$. However, there are other options, including hazards, failure times, distribution quantiles, and many more.

$$E(Y \mid X) = f(X)$$

In principle, this model is most flexible in that it states that the conditional mean of Y is simply a *arbitrary function* of the covariates X. We are not stating (or assuming) precisely **how** the conditional mean is related to these covariates. Using this model, we might get predictions from to facilitate a decision making process, or obtain a contrast of expected means between two groups.

Because of the flexibility of this model, we may be interested in fitting it to a given dataset. But we can't. There is simply not enough information in this equation for us to quantify $f(X)$, even if we had all the data we could use. In addition to data, we need some "traction" or "leverage" to be able to quantify the function of interest.

The earliest attempt to find some "traction" to quantify $f(X)$ was proposed in the 1800s. The approach starts by accepting a few tenets[2]. First, we want the difference between the observed Y for any given individual and the fitted values $f(X)$ for that person to be "small." If we didn't care about the direction of these errors (we usually don't), we can square the error and take it's average: $E[(Y - f(X))^2]$. Thus, we can define the "optimal" $f(X)$ as the function of X that minimizes the mean squared error.

[2] Much of this section is based on the (excellent) forthcoming (?) book by Cosma Shalizi (2019).

Mean squared error can be be re-written as the sum of the squared bias and the variance:

$$E[(Y - f(X))^2] = [E(Y) - f(X)]^2 + Var(Y)$$

And you may recall that finding the $f(X)$ that minimizes mean squared error can be achieved by taking the derivative of the mean squared error with respect to $f(X)$, setting it to zero, then solving for $f(X)$.

We've made some progress, but without a better sense of what $f(X)$ looks like, we still can't move forward. For example, there are several functions where either the derivative simply does not exist (e.g., if $f(X)$ is discontinuous), or where the derivative is still complex enough that we can't make progress with finding a unique solution for $f(X)$ that minimizes mean squared error (see technical note on nonlinear models).

Early on, it was recognized that if we select $f(X)$ to be *linear* (more technically, *affine*) the problem of finding the optimal $f(X)$ becomes much easier. That is, if we can simplify $f(X) = b_0 + b_1 X$, then we can use calculus and simple algebra to find an optimal *linear* solution set $b_0 = \beta_0, b_1 = \beta_1$ that minimizes MSE.

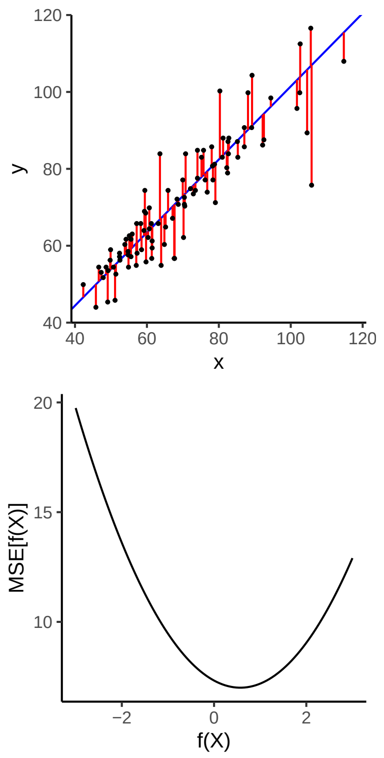

Figure 1: Line of 'best fit' (blue line) defined on the basis of minimizing the sum of squared residuals (red lines) displayed in the top panel; Partial representation of the mean squared error as a function of f(X) in the bottom panel.

Technical Note:
Technically (almost to the point of pedantry), a nonlinear model is a model where the first derivative of the expectation taken with respect to the parameters is itself a function of other parameters (Seber and Wild, 1989). For example,

$$E(Y \mid X) = \beta_0 + \frac{X}{\beta_1}$$

is a nonlinear model, because it's first derivative taken with respect to β_1 is still a function of β_1:

$$\frac{dE(Y \mid X)}{d\beta_1} = \frac{d\left(\beta_0 + \frac{X}{\beta_1}\right)}{d\beta_1} = -\frac{X}{\beta_1^2}$$

Why is this important? Solutions to these regression equations (which serve as our estimates), are obtained by finding where the slope of the tangent line of the parameters is zero. To do this, we need to set the first derivative of these regression equations to zero. But if there are still parameters in these first derivative equations, then there will not be a unique solution to the equation, and finding an estimate will require more complex approaches. This is the complication introduced by nonlinear models.
On the other hand, curvilinear models are easy to find solutions for, since their first derivatives are not functions of parameters. For instance, for a quadratic model such as:

$$E(Y \mid X) = \beta_0 + \beta_1 X + \beta_2 X^2$$

The first derivatives taken with respect to each parameters turn out to be:

$$\frac{dE(Y \mid X, C)}{d\beta_0} = 1$$

$$\frac{dE(Y \mid X, C)}{d\beta_1} = X$$

$$\frac{dE(Y \mid X, C)}{d\beta_2} = X^2$$

Thus, even though the regression function will not be a “straight line” on a plot, this model is still linear.

Specifically, we can re-write the mean squared error as a function of $b_0 + b_1 X$ to be $MSE(b_0, b_1) = E[(Y - (b_0 + b_1 X))^2]$. We need not get into details here, but taking the partial derivatives of $MSE(b_0, b_1)$ with respect to b_0 and b_1 gives us the ordinary least squares estimator (Rencher, 2000, Shalizi (2019)).

There are some important points to note in how we formulated the problem of estimating $E(Y \mid X) = f(X)$ with a linear model:

- What we needed to invoke to make this work is that a linear approximation to $f(X)$ is "good enough" for our interest. We need the linear approximation so we can take derivatives of the MSE function without running into problems. These assumptions are usually referred to as "regularity" conditions (Longford, 2008).
- We didn't explicitly state it, but on the basis of Figure 1, this approach makes most sense if Y is continuous. If Y is binary, categorical, or time-to-event, the rationale motivating this approach starts to break down to a degree. That is, while it is possible to fit an OLS estimator to a binary outcome data, not everyone agrees that this is a good idea, and caution is warranted when doing so.
- We *did not* need to invoke any any assumptions about homoscedasticity, or independent and identically distributed (iid) observations. If we were able to make these assumptions, then we obtain an estimator that is the "best linear unbiased estimator" (Rencher, 2000).
- We *did not* need to invoke any distributional assumptions about Y (or, more specifically, the conditional mean of Y). If we can assume Gaussian with constant variance, then we can equate the OLS estimator with the maximum likelihood estimator with a Gaussian distribution and identify link function.

Unfortunately, in the way we formulated it above, the set of linear models we can use to find a solution $f(X)$ that minimizes $MSE(f(X))$ is limited. For instance, in the case where Y is binary, and $E(Y) = P(Y = 1)$, using a linear model such as $b_0 + b_1X$ can easily lead to problems, most notably that predicted probabilities lie outside of the bounds $[0, 1]$.

We may elect to constrain $f(X)$ so that the predictions are forced to lie within $[0, 1]$. For instance, we could use the inverse of the logistic function and define $f(X) = \frac{1}{1+\exp(-b_0-b_1X)}$. However, in this case the derivatives would no longer work out as simply as we'd need them too (see Technical Note). In effect, we would now have to deal with the fact that the model is nonlinear (i.e., the derivatives of the model cannot simply be set to zero).[3]

As it turns out, in the early 1970's, Nelder and Wedderburn (Nelder and Wedderburn, 1972) made a seminal contribution that enabled fitting nonlinear models when the outcome belongs to the exponential family of distributions,[4] and

[3] I am taking many liberties here. The formulation under a logistic link function is different in that it is usually (required?) expressed in terms of likelihood functions, and not simply MSE. There are relations between these formulations, but we need not get into that here.

[4] The exponential family of distributions is not to be confused with the exponential distribution. It refers to a family of distributions that can be re-written such that they can be represented in a common form. These distributions include (but are not limited to) the Gaussian, exponential, bernoulli (binomial), Poisson, and negative binomial.

the conditional mean of the outcome can be linked to the covariates through some smooth and invertible linearizing function (which includes the $\log$ and logit functions).

2 Conditionally Adjusted Parametric Regression

Suppose we wanted to use the NHEFS data to estimate the confounder adjusted effect of quitting smoking on weight change. Suppose further that we were interested specifically in whether an individual's weight change was greater than the median value in the data.

```
nhefs <- read_csv(here("data", "nhefs.csv")) %>%
    mutate(wt_delta = as.numeric(wt82_71 >
        median(wt82_71)))

#' Quick view of data
dim(nhefs)
```

```
## [1] 1394   12
```

```
names(nhefs)
```

```
##  [1] "seqn"     "qsmk"     "sex"
##  [4] "age"      "income"   "sbp"
##  [7] "dbp"      "price71"  "tax71"
## [10] "race"     "wt82_71"  "wt_delta"
```

A typical approach would be to use a generalized linear model to regress the indicator of weight change against the indicator of whether the individual quit smoking:

```
#' Here, we start fitting relevant regression models to the data.
#' modelForm is a regression argument that one can use to regress the
#' outcome (wt_delta) against the exposure (qsmk) and selected confounders.

formulaVars <- "qsmk + sex + age + income + sbp + dbp + price71 + tax71 + race"
```

```
modelForm <- as.formula(paste0("wt_delta ~",
    formulaVars))
modelForm
```

```
## wt_delta ~ qsmk + sex + age + income + sbp + dbp + price71 +
##     tax71 + race
```

```
#' This model can be used to quantify a conditionally adjusted
#' odds ratio with correct standard error
modelOR <- glm(modelForm, data = nhefs, family = binomial("logit"))

summary(modelOR)
```

```
##
## Call:
## glm(formula = modelForm, family = binomial("logit"), data = nhefs)
##
## Coefficients:
##              Estimate Std. Error z value
## (Intercept) -0.822736   1.178775  -0.698
## qsmk         0.594853   0.133325   4.462
## sex          0.126227   0.113976   1.107
## age         -0.033421   0.005579  -5.991
## income       0.027180   0.022297   1.219
## sbp         -0.004227   0.004232  -0.999
## dbp          0.027001   0.006978   3.870
## price71      0.121377   0.837939   0.145
## tax71       -0.193909   0.886114  -0.219
## race        -0.304103   0.175398  -1.734
##             Pr(>|z|)
## (Intercept) 0.485203
## qsmk        8.13e-06 ***
## sex         0.268084
## age         2.09e-09 ***
## income      0.222846
## sbp         0.317912
## dbp         0.000109 ***
## price71     0.884828
## tax71       0.826782
## race        0.082956 .
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1932.5  on 1393  degrees of freedom
## Residual deviance: 1828.0  on 1384  degrees of freedom
## AIC: 1848
##
## Number of Fisher Scoring iterations: 4
```

```
tidy(modelOR)[2, ]
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic    p.value
##   <chr>    <dbl>     <dbl>     <dbl>      <dbl>
## 1 qsmk     0.595     0.133      4.46 0.00000813
```

If we were interested in estimating conditionally adjusted risk differences or risk ratios for the effect of quitting smoking, we could use a similar approach with the identity link function, ordinary least squares, or Poisson regression (Zou, 2004, Naimi and Whitcomb (2020)).

3 Marginally Adjusted Parametric Regression

Another approach to obtaining risk differences and ratios from GLMs that are not subject to the limitations noted above is to use marginal standardization, which is equivalent to g computation (aka the parametric g formula) when the exposure is measured at a single time point (Naimi et al., 2017). This process can be implemented by fitting a single logistic model, regressing the binary outcome against all confounder variables. But instead of reading the coefficients the model, one can obtain odds ratios, risk ratios, or risk differences by using this model to generate predicted risks for each individual under "exposed" and "unexposed" scenarios in the dataset. To obtain standard errors, the entire procedure must be bootstrapped.

Here is some code to implement this marginal standardization in the NHEFS data:

```r
#' Regress the outcome against the confounders with interaction
ms_model <- glm(modelForm, data = nhefs,
    family = binomial("logit"))
##' Generate predictions for everyone in the sample to obtain
##' unexposed (mu0 predictions) and exposed (mu1 predictions) risks.
mu1 <- predict(ms_model, newdata = transform(nhefs,
    qsmk = 1), type = "response")
mu0 <- predict(ms_model, newdata = transform(nhefs,
    qsmk = 0), type = "response")

#' Marginally adjusted odds ratio
marg_stand_OR <- (mean(mu1)/mean(1 - mu1))/(mean(mu0)/mean(1 -
    mu0))
#' Marginally adjusted risk ratio
marg_stand_RR <- mean(mu1)/mean(mu0)
#' Marginally adjusted risk difference
marg_stand_RD <- mean(mu1) - mean(mu0)

#' Using the bootstrap to obtain confidence intervals for the marginally adjusted
#' risk ratio and risk difference.
bootfunc <- function(data, index) {
    boot_dat <- data[index, ]
    ms_model <- glm(modelForm, data = boot_dat,
        family = binomial("logit"))
    mu1 <- predict(ms_model, newdata = transform(boot_dat,
        qsmk = 1), type = "response")
    mu0 <- predict(ms_model, newdata = transform(boot_dat,
        qsmk = 0), type = "response")

    marg_stand_OR_ <- (mean(mu1)/mean(1 -
        mu1))/(mean(mu0)/mean(1 - mu0))
    marg_stand_RR_ <- mean(mu1)/mean(mu0)
    marg_stand_RD_ <- mean(mu1) - mean(mu0)
    res <- c(marg_stand_RD_, marg_stand_RR_,
        marg_stand_OR_)
    return(res)
}

#' Run the boot function. Set a seed to obtain reproducibility
set.seed(123)
boot_res <- boot(nhefs, bootfunc, R = 2000)

boot_RD <- boot.ci(boot_res, index = 1)
boot_RR <- boot.ci(boot_res, index = 2)
boot_OR <- boot.ci(boot_res, index = 3)

marg_stand_OR
```

```
## [1] 1.746213
```

```
marg_stand_RR
```

```
## [1] 1.295328
```

```
marg_stand_RD
```

```
## [1] 0.1377615
```

```
boot_RD
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
```

```
##
## CALL :
## boot.ci(boot.out = boot_res, index = 1)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 0.0793,  0.1966 )   ( 0.0802,  0.1976 )
##
## Level     Percentile            BCa
## 95%   ( 0.0779,  0.1953 )   ( 0.0779,  0.1952 )
## Calculations and Intervals on Original Scale
```

```
boot_RR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 2)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.158,  1.431 )   ( 1.156,  1.428 )
##
## Level     Percentile            BCa
## 95%   ( 1.163,  1.435 )   ( 1.163,  1.435 )
## Calculations and Intervals on Original Scale
```

```
boot_OR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 3)
```

```
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.302,  2.162 )   ( 1.266,  2.125 )
##
## Level     Percentile            BCa
## 95%   ( 1.367,  2.227 )   ( 1.367,  2.224 )
## Calculations and Intervals on Original Scale
```

This marginal standardization approach yields an estimate of the average treatment effect under the required identifiability assumptions. However, it assumes a constant effect of qsmk on weight change across levels of all of the other variables in the model. This constant effect assumption might be true, but if one wanted to account for potential interactions between the exposure and all of the confounders in the model, there is an easy way. We call this the "stratified modeling approach."

This stratified modeling approach avoids the exposure effect homogeneity assumption across levels of all the confounders. In effect, the approach fits a separate model for each exposure stratum. To obtain predictions under the "exposed" scenario, we use the model fit to the exposed individuals to generate predicted outcomes in the entire sample. To obtain predictions under the "unexposed" scenario, we repeat the same procedure, but with the model fit among the unexposed. One can then average the risks obtained under each exposure scenario, and take their difference and ratio to obtain the risk differences and ratios of interest.

```
#' Marginal Standardization
##' To avoid assuming no interaction between
##' quitting smoking and any of the other variables
##' in the model, we subset modeling among
##' exposed/unexposed. This code removes qsmk from the model,
##' which will allow us to regress the outcome
##' against the confounders among the exposed and
##' the unexposed separately. Doing so will allow us
##' to account for any potential exposure-covariate interactions
##' that may be present.
```

```r
formulaVars <- "sex + age + income + sbp + dbp + price71 + tax71 + race"
modelForm <- as.formula(paste0("wt_delta ~",
    formulaVars))
modelForm
```

```
## wt_delta ~ sex + age + income + sbp + dbp + price71 + tax71 +
##     race
```

```r
#' Regress the outcome against the confounders
#' among the unexposed (model0) and then among the exposed (model1)
model0 <- glm(modelForm, data = subset(nhefs,
    qsmk == 0), family = binomial("logit"))
model1 <- glm(modelForm, data = subset(nhefs,
    qsmk == 1), family = binomial("logit"))
##' Generate predictions for everyone in the sample using the model fit to only the
##' unexposed (mu0 predictions) and only the exposed (mu1 predictions).
mu1 <- predict(model1, newdata = nhefs, type = "response")
mu0 <- predict(model0, newdata = nhefs, type = "response")

#' Marginally adjusted odds ratio
marg_stand_OR <- (mean(mu1)/mean(1 - mu1))/(mean(mu0)/mean(1 -
    mu0))
#' Marginally adjusted risk ratio
marg_stand_RR <- mean(mu1)/mean(mu0)
#' Marginally adjusted risk difference
marg_stand_RD <- mean(mu1) - mean(mu0)

#' Using the bootstrap to obtain confidence intervals for the marginally adjusted
#' risk ratio and risk difference.
bootfunc <- function(data, index) {
    boot_dat <- data[index, ]
    model0 <- glm(modelForm, data = subset(boot_dat,
        qsmk == 0), family = binomial("logit"))
    model1 <- glm(modelForm, data = subset(boot_dat,
        qsmk == 1), family = binomial("logit"))
    mu1 <- predict(model1, newdata = boot_dat,
        type = "response")
    mu0 <- predict(model0, newdata = boot_dat,
        type = "response")

    marg_stand_OR_ <- (mean(mu1)/mean(1 -
        mu1))/(mean(mu0)/mean(1 - mu0))
    marg_stand_RR_ <- mean(mu1)/mean(mu0)
    marg_stand_RD_ <- mean(mu1) - mean(mu0)
    res <- c(marg_stand_RD_, marg_stand_RR_,
        marg_stand_OR_)
    return(res)
}

#' Run the boot function. Set a seed to obtain reproducibility
set.seed(123)
boot_res <- boot(nhefs, bootfunc, R = 2000)

boot_RD <- boot.ci(boot_res, index = 1)
boot_RR <- boot.ci(boot_res, index = 2)
boot_OR <- boot.ci(boot_res, index = 3)

marg_stand_OR
```

```
## [1] 1.757907
```

```
marg_stand_RR
```

```
## [1] 1.298577
```

```
marg_stand_RD
```

```
## [1] 0.139347
```

```
boot_RD
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 1)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 0.0794,  0.1995 )   ( 0.0791,  0.1996 )
##
## Level     Percentile            BCa
## 95%   ( 0.0791,  0.1996 )   ( 0.0790,  0.1995 )
## Calculations and Intervals on Original Scale
```

```
boot_RR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 2)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.159,  1.436 )   ( 1.156,  1.432 )
##
## Level     Percentile            BCa
## 95%   ( 1.165,  1.442 )   ( 1.164,  1.440 )
## Calculations and Intervals on Original Scale
```

```
boot_OR
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_res, index = 3)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 1.297,  2.187 )   ( 1.247,  2.142 )
##
## Level     Percentile            BCa
## 95%   ( 1.374,  2.269 )   ( 1.372,  2.261 )
## Calculations and Intervals on Original Scale
```

When predicted risks are estimated using a logistic model, relying on marginal standardization will not result in probability estimates outside the bounds [0, 1]. And because the robust variance estimator is not required, model-based standardization will not be as affected by small sample sizes. However, the bootstrap is more computationally demanding than alternative variance estimators, which may pose problems in larger datasets.

4 Introduction to Propensity Score Methods

So far, we've focused on obtaining quantitative estimates of the average treatment effect using an outcome modeling approach. In this approach, one regresses the outcome against the exposure and confounders. This works for the conditionally adjusted estimator, where we read the coefficient for the exposure from the regression model, or the marginally adjusted estimator (e.g., using marginal standardization). The basic formulation of this outcome modeling approach can be depicted heuristically using the DAG in Figure 2:

In this Figure, the open back-door path from X to Y is blocked by conditioning on C. This leads to a conditionally adjusted estimate of the exposure effect. One can marginalize over the distribution on C to get a marginally adjusted estimate from a model for the outcome. For a binary outcome, such estimates may be obtained on the risk difference, risk ratio, or odds ratio scales, or one may obtain a marginally adjusted estimate of the cumulative risk function that would be observed if everyone were exposed or unexposed.

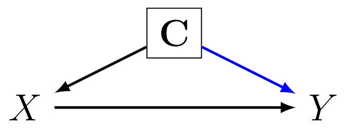

Figure 2: Directed acyclic graph depicting confounder adjustment using an outcome modelling approach. In this approach, information from the confounder to the outcome is 'blocked' (blue arrow). With this adjustment, the exposure X os d-separated from the outcome Y, rendering the estimate of the exposure-outcome association unconfounded.

On the other hand, we may want to obtain an adjusted estimate of the exposure effect by modeling the exposure. This can be accomplished using propensity score methods.

The propensity score was first defined by Rosenbaum and Rubin (Rosenbaum and Rubin, 1983) as the conditional probability of receiving the treatment (or, equivalently, of being exposed). The true propensity score is defined as

$$e(C) = P(X = 1 \mid C)$$

This propensity score is typically known in a randomized trial. It's important to distinguish this true PS from the estimated PS:

$$\hat{e}(C) = \hat{P}(X = 1 \mid C)$$

This estimated PS can be fit using a parametric model, in which case, we might write:

$$\hat{e}(C; \alpha) = \hat{P}(X = 1 \mid C) = \text{expit}(\alpha C)$$

For reasons that are not entirely obvious, it is usually better to use the estimated PS, even when the true PS is known (Robins et al., 1992, Henmi and Eguchi (2004)).

The propensity score is a univariate proxy for the multivariate set of confounders. We can use the propensity score to adjust for confounding using a number of different techniques. These techniques include regression adjustment, propensity score matching, stratification, and weighting. In this lecture, we will focus on the creation and use of IP-weights. The reason for this emphasis is that i in epidemiology, the most commonly used PS technique is inverse probability weighting, and so it is important to know how to implement; and ii, there are theoretical justifications suggesting that weighting is the most optimal use of the PS.

5 Inverse Probability Weighting

The most commonly employed propensity score adjustment technique is inverse probability weighting. The simple heuristic often used to describe the way IP-weighting works is that, when applied to data, they yield a "pseudo population" where there is no longer an effect of the confounder on the exposure. The causal structure of the variables in this new pseudo-population can be depicted in Figure 3:

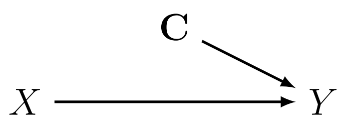

Figure 3: Directed acyclic graph depicting the causal relations between variables in a 'pseudo-population' obtained via inverse probability weighting. Again, with this adjustment, the exposure X os d-separated from the outcome Y, rendering the estimate of the exposure-outcome association unconfounded.

The weights for each individual needed to create this pseudo-population are defined as the inverse of the probability of receiving their observed exposure. Let's consider the following simple example to explain why this works. In this example, there are 20 observations, with one binary confounder (50:50 split) and one binary exposure. Let's suppose the probability that the exposure `x = 1` is 0.2 for those with `c = 0` and 0.9 for those with `c = 1`:

Under these scenarios, we'd have exposure and confounder data that looks like this:

```
c <- c(0, 0, 1, 1)
x <- c(1, 0, 1, 0)
n <- c(2, 8, 9, 1)

tibble(x, c, n)
```

```
## # A tibble: 4 x 3
##       x     c     n
##   <dbl> <dbl> <dbl>
## 1       1       0       2
## 2       0       0       8
## 3       1       1       9
## 4       0       1       1
```

Let's focus on the stratum of individuals with `c = 0`. In this stratum, there are 2 exposed individuals, and 8 unexposed individuals. Now let's ask what these data should look like if we were able to implement an ideal (marginally) randomized trial with the probability of treatment being 50% for all individuals. We would expect that within the stratum of individuals with `c = 0`, there would be 5 exposed and 5 unexposed individuals. Inverse probability weighting seeks to accomplish this balance.

Consider that the inverse probability of the **observed exposure** among those with `c = 0` is 0.2 for those who were actually exposed, and 0.8 for those who were unexposed.

The inverse probability weight is thus $\frac{1}{0.2} = 5$ for the exposed in this confounder stratum, and $\frac{1}{0.8} = 1.25$ for the unexposed in this confounder stratum.

This suggests that, in their contribution to the overall analysis, the two exposed individuals in the `c = 0` status would each receive a weight of 5, while the eight unexposed individuals in the `c = 0` stratum would each receive a weight of 1.25. Under these conditions, we would have a re-balanced set of observations in the `c = 0` stratum of:

$$\begin{aligned}\text{Exposed Observations:} \quad & 5 \times 2 = 10 \\ \text{Unexposed Observations:} \quad & 8 \times 1.25 = 10\end{aligned}$$

In effect, our inverse probability weighting strategy made it such that we now have an equal number of exposed and unexposed observations within the `c = 0` stratum. In these weighted data, we can now compute the difference in the outcome among the exposed and unexposed individuals (if we had it) to obtain an estimate of our average treatment effect.

Simply taking the inverse of the probability of the observed exposure, while valid, is not the usual strategy for implementing inverse probability weights. In practice, one will often use stabilized weights, stabilized normalized weights, potentially with some degree of "truncation" or, more accurately, trimming of the weights.[5]

Furthermore, it's important to note that "data" are often not weighted, but rather the contribution that each individual in the sample makes to the estimating function (e.g., likelihood, estimating equation, or other function used to find parameters). This is important in that one must choose a fitting algorithms that allows for this type of weighting.

[5] In contrast to our emphasis of the usage of the word "truncation" which refers to the removal of observations from the dataset, researchers will often refer to "truncating" the weights, which sets the largest value to be equal to the 99th or 95th percentile values. This is more accurately referred to as "trimming" the weights, since no truncation is occurring.

To start, the simplest type of weight used in practice is the stabilzied inverse probability weight. These are often defined as:

$$sw = \begin{cases} \dfrac{P(X=1)}{P(X=1 \mid C)} & \text{if } X = 1 \\ \dfrac{P(X=0)}{P(X=0 \mid C)} & \text{if } X = 0 \end{cases}$$

but are sometimes written more succinctly as[6]:

$$sw = \frac{f(X)}{f(X \mid C)}$$

[6] This formulation is unusual, since $f(.)$ represents the probability density function, which is usually taken for a specific realization of the random variable. However, in this case, because the weights are defined as a function of the observed exposure status, the argument in the operator is the observed data (denoted with capital letter), as opposed to some specific realization.

Let's use the cohort data again to construct the stabilized weights. We will re-fit the PS model to the data, and construct the weights:

```r
# create the propensity score in the
# dataset
nhefs$propensity_score <- glm(qsmk ~ sex +
    age + income + sbp + dbp + price71 +
    tax71 + race, data = nhefs, family = binomial("logit"))$fitted.values

# stabilized inverse probability
# weights
nhefs$sw <- (mean(nhefs$qsmk)/nhefs$propensity_score) *
    nhefs$qsmk + ((1 - mean(nhefs$qsmk))/(1 -
    nhefs$propensity_score)) * (1 - nhefs$qsmk)

summary(nhefs$sw)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.
##  0.5090  0.9110  0.9814  1.0001  1.0638
##    Max.
##   3.0970
```

```
nhefs %>%
    select(seqn, qsmk, wt_delta, propensity_score,
        sw) %>%
    print(n = 5)
```

```
## # A tibble: 1,394 x 5
##     seqn  qsmk wt_delta propensity_score    sw
##    <dbl> <dbl>    <dbl>            <dbl> <dbl>
## 1   233     0        0            0.205 0.951
## 2   235     0        1            0.254 1.01
## 3   244     0        1            0.154 0.894
## 4   245     0        1            0.241 0.996
## 5   252     0        1            0.260 1.02
## # i 1,389 more rows
```

As we can see from the output above, the stabilized weights are, in fact, well behaved, with a mean of one and a max value that is small.

```
model_RD_weighted <- glm(wt_delta ~ qsmk,
    data = nhefs, weights = sw, family = quasibinomial("identity"))

summary(model_RD_weighted)$coefficients
```

```
##              Estimate Std. Error   t value
## (Intercept) 0.4668671 0.01537912 30.357194
## qsmk        0.1380031 0.03066295  4.500646
##                  Pr(>|t|)
## (Intercept) 9.184292e-156
## qsmk         7.339547e-06
```

A final note that is important with these weighted approaches is to consider how to estimate the standard errors. In fact, the model-based standard errors are no longer valid when weighting is used. One must instead use the robust variance estimators, or the bootstrap. For example:

```r
library(lmtest)
library(sandwich)
coeftest(model_RD_weighted, vcov. = vcovHC)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error z value
## (Intercept) 0.466867   0.015446 30.2249
## qsmk        0.138003   0.032042  4.3069
##              Pr(>|z|)
## (Intercept) < 2.2e-16 ***
## qsmk        1.655e-05 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

One can then construct CIs in the standard way using the estimated standard error in the output above.

References

Masayuki Henmi and Shinto Eguchi. A paradox concerning nuisance parameters and projected estimating functions. *Biometrika*, 91(4):929–941, 12 2004.

NT Longford. *Studying Human Populations: An Advanced Course in Statistics*. Springer, New York, 2008.

Ashley I Naimi and Brian W Whitcomb. Estimating risk ratios and risk differences using regression. *American Journal of Epidemiology*, 189(6):508–510, 2020.

Ashley I Naimi, Stephen R Cole, and Edward H Kennedy. An Introduction to G Methods. *Int J Epidemiol*, 46(2):756–62, 2017.

J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *JRSS-A*, 135 (3):370–384, 1972.

Alvin C. Rencher. *Linear Models in Statistics*. Wiley, New York, 2000.

J. M. Robins, S. D. Mark, and W. K. Newey. Estimating exposure effects by modelling the expectation of exposure conditional on confounders. *Biometrics*, 48(2):479–95, 1992.

Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

G. A. F. Seber and C. J. Wild. *Nonlinear regression*. Wiley, New York, 1989.

Cosma Rohilla Shalizi. *The Truth About Linear Regression*. https://www.stat.cmu.edu/ cshalizi/TALR/TALR.pdf, 2019.

Guangyong Zou. A modified poisson regression approach to prospective studies with binary data. *Am J Epidemiol*, 159(7):702–706, Apr 2004.