Inverse Probability Weighting

Ashley I. Naimi, PhD

Outline

Inverse Probability Weighting

- Some Preliminaries
- Standardization via IPW
- Time-Varying Confounding

Some Preliminaries

In this section, we will illustrate implementation of inverse probability weighting to estimate the average treatment (causal) effect. The previous example used the g computation algorithm to accomplish the same thing. The average causal effect is a *marginal* effect because it averages (or marginalizes) over all individual-level effects in the population. In contrast, a conditional causal effect focuses on the effect among those with particular characteristics.

When the exposure is measured at a single time point, the distinction between marginal and conditional effecst is relatively uncomplicated. In fact, inverse probability weighting can be used to estimate both marginal and conditional effects in this setting. Complications arise when the exposure and confounders are time-varying.

Consider again the basic time-varying confounding structure we introduced previously:

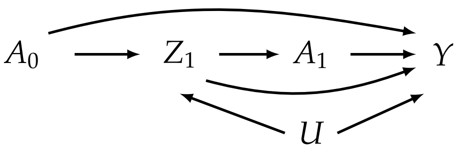

Figure 1: Causal diagram representing the relation between anti-retroviral treatment at time 0 (A_0), HIV viral load just prior to the second round of treatment (Z_1), anti-retroviral treatment status at time 1 (A_1), the CD4 count measured at the end of follow-up (Y), and an unmeasured common cause (U) of HIV viral load and CD4.

The average causal effect for this structure can be defines as $\psi = E(Y^{a_0,a_1} - Y^{0,0})$. We can write this effect as $E(Y^{a_0,a_1} - Y^{0,0}) = \psi_0 a_0 + \psi_1 a_1 + \psi_2 a_0 a_1$, which states that our average causal effect ψ may be composed of two exposure main effects (e.g., ψ_0 and ψ_1) and their two-way interaction (ψ_2). This marginal effect ψ is indifferent to whether the A_1 component ($\psi_1 + \psi_2$) is modified by Z_1: whether such effect modification is present or absent, the marginal effect represents a meaningful answer to the question: what is the effect of A_0 and A_1 in the entire population?

Alternatively, we may wish to estimate this effect *conditional* on Z; for example, if one was specifically interested in effect measure modification by Z_1. When properly modeled, this conditional effect represents a meaningful answer to the question: what is the effect of A_0 and A_1 in those who receive $Z_1 = 1$ versus those who receive $Z_1 = 0$? Unfortunately, IP-weighting cannot be used to quantify such a conditional effect. Indeed, modeling such effect measure modification by time-varying covariates is the fundamental issue that distinguishes marginal structural from structural nested models. We will thus save such EMM for the next section on structural nested models.

In the remaining IP-weighting sections, we will re-visit the g-computation examples and estimate the contrasts of interest using IP-weighting instead.

Example 1: Standardization via IPW in the NHEFS Data

Let's revisit the question about whether quitting smoking will change the risk of gaining weight in the NEHFS data. We will analyze these data using **IP-weighting, which is another way to obtain a standardized effect measure, and thus equivalent to the parametric g formula.**

We again start our estimand of interest, the same marginal average causal effect on the difference scale:

$$E(Y^{a=1} - Y^{a=0})$$

This time, instead of quantifying two averages, we can specify a model that will give us both:

$$E(Y^a) = \psi_0 + \psi_1 a$$

This is a marginal structural model. It is marginal because it models the marginal value of the potential outcome. It is structural because it models the potential outcome. Note that this is the same expectation that we modeled using the g computation algorithm. Thus, both g computation and IP-weighting can be used to quantify the parameters of a marginal structural model.[1]

```
aa <- read_csv("./nhefs.csv")
# original sample size
nrow(aa)
```

```
## [1] 1746
```

[1] Recall, it is important to distinguish between the estimand and the estimator. In our case, the estimand is a parameter from a marginal structural model. In the section on g computation, the estimator was the g computation estimator. Here, we are using an IP-weighted estimator.

```
a <- aa %>% select(seqn, qsmk, smkintensity82_71, smokeintensity, active, exercise, wt82_71,
    sbp, dbp, hbp, hf, ht, hbpmed, sex, age, hf, race, income, marital, school, asthma, bronch,
    diabetes)
a$hbp_71 <- a$hbp

a <- a %>% na.omit()

a$delta <- as.numeric(a$wt82_71 > 0)
```

Let's again assume that the relevant confounders are: `sex`, `age`, `race`, `income`, `marital`, `school`, `active`, `hf`, `hbpmed`, `asthma`, `bronch`, `smokeintensity`, `exercise`, `diabetes`, and hbp_71.

```
a <- a %>% select(delta, qsmk, sex, age, race, income, marital, school, active, hf, hbpmed,
    asthma, bronch, smokeintensity, exercise, diabetes, hbp_71)

a$smokeintensity <- as.numeric(a$smokeintensity > median(a$smokeintensity))
a$age <- as.numeric(a$age > median(a$age))
a$exercise <- as.numeric(a$exercise > 0)
a$income <- as.numeric(a$income > median(a$income))
a$marital <- as.numeric(a$marital > median(a$marital))
a$school <- as.numeric(a$school > median(a$school))
a$active <- as.numeric(a$active > 0)
a$hbpmed <- as.numeric(a$hbpmed > 0)
a$smokeintensity <- as.numeric(a$smokeintensity > median(a$smokeintensity))
a$exercise <- as.numeric(a$exercise > 0)
a$diabetes <- as.numeric(a$diabetes > 0)
a$hbp_71 <- as.numeric(a$hbp_71 > 0)
```

Once we've defined our contrast and the model we will use to quantify it, the next step is to obtain predicted probabilities for the observed exposure. We can do this with a logistic model for our binary exposure in the NHEFS data. Note that this is the same model we used to quantify the natural course in the g formula example:

```
# obtain propensity score
propensity <- glm(qsmk ~ sex + age + race + income + marital + school + active + hf + hbpmed +
    asthma + bronch + smokeintensity + exercise + diabetes + hbp_71, data = a, family = binomial("logit"))
```

The stabilized inverse probability weights are defined as $P(A = 1 \mid C)$ if $A = 1$ and $P(A = 0 \mid C)$ if $A = 0$. We can accomplish this using a coding flag trick:

```
# create stabilized IPW
sw <- (mean(a$qsmk)/propensity) * a$qsmk + (mean(1 - a$qsmk)/(1 - propensity)) * (1 - a$qsmk)
```

To verify our weights are well behaved, we can look at their summary:

```
summary(sw)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5110  0.9037  0.9766  0.9994  1.0621  3.0169
```

The important thing to note here is that the mean of the stabilized weights is close to 1, and the max weight is not very large. This is in line with what we saw in the propensity score overlap plot. Poor overlap tends to yield large stabilized weights with a mean that is not close to one.

Now we can actually implement IP-weighting to quantify the marginal treatment effect:

```
ipw_msmRD <- glm(delta ~ qsmk, data = a, weights = sw, family = gaussian("identity"))
##### gaussian distribution with identity link yields risk difference
RD <- coef(ipw_msmRD)[2]

ipw_msmRR <- glm(delta ~ qsmk, data = a, weights = sw, family = poisson("log"))
##### poisson distribution with log link yields log risk ratio
logRR <- coef(ipw_msmRR)[2]
```

When fit using IPW, the risk difference for quitting smoking on weight gain is 10.68 per 100 participants, and the risk ratio is 1.17.

Confidence Intervals

In contrast to the g computation estimator, where confidence intervals can only be obtained using the bootstrap, there are two ways to get confidence intervals for an IP-weighted estimator. The first is analytic, and based on the robust (or sandwich) variance estimator.

In R, the robust variance estimator can be implemented using the `lmtest` and `sandwich` packages:

```
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
RD_CI <- coefci(ipw_msmRD, vcov = vcovHC(ipw_msmRD, type = "HC1"))
RR_CI <- coefci(ipw_msmRR, vcov = vcovHC(ipw_msmRR, type = "HC1"))
```

Based on the sandwich variance estimator, the 95% CIs for the risk difference and risk ratio are 5.14, 16.21 and 1.08, 1.26, respectively.

We can (should?) also use the bootstrap (Austin 2016). To use the bootstrap correctly, the propensity score must be re-fit at each re-sample:

```
res <- NULL
for (i in 1:100) {
    index <- sample(1:nrow(a), nrow(a), replace = T)
    boot_dat <- a[index, ]
    propensity <- glm(qsmk ~ sex + age + race + income + marital + school + active + hf + hbpmed +
        asthma + bronch + smokeintensity + exercise + diabetes + hbp_71, data = boot_dat, family = binomia

    sw <- (mean(boot_dat$qsmk)/propensity) * boot_dat$qsmk + (mean(1 - boot_dat$qsmk)/(1 -
        propensity)) * (1 - boot_dat$qsmk)

    ipw_msmRD <- glm(delta ~ qsmk, data = boot_dat, weights = sw, family = gaussian("identity"))
    ##### gaussian distribution with identity link yields risk difference
    RD <- coef(ipw_msmRD)[2]

    ipw_msmRR <- glm(delta ~ qsmk, data = boot_dat, weights = sw, family = poisson("log"))
    ##### poisson distribution with log link yields log risk ratio
    logRR <- coef(ipw_msmRR)[2]
    res <- rbind(res, cbind(RD, logRR))
}

head(res)
```

```
##              RD     logRR
## qsmk 0.14739925 0.2025900
## qsmk 0.11190572 0.1574205
## qsmk 0.12003629 0.1751206
## qsmk 0.10446873 0.1495776
## qsmk 0.06955607 0.1005446
## qsmk 0.15622068 0.2213984
```

```
res_sd <- apply(res, 2, sd)

lclRD <- RD - 1.96 * res_sd[1]
uclRD <- RD + 1.96 * res_sd[1]

lclRR <- exp(logRR - 1.96 * res_sd[2])
uclRR <- exp(logRR + 1.96 * res_sd[2])
```

This bootstrap estimator yields 95% CIs of 3.08, 14.29 for the risk difference, and 1.05, 1.22 for the risk ratio. To summarize the results from the IPW estimator:

The IPW risk difference was 8.68, with 95% sandwich CIs of 5.14, 16.21, and 95% bootstrap CIs of 3.08, 14.29.

The IPW risk ratio was 1.13, with 95% sandwich CIs of 1.08, 1.26, and 95% bootstrap CIs of 1.05, 1.22.

Interpretation

Interpretation wise, we have to consider all the same issues as with the g computation algorithm. There are just two considerations that differ.

1) correct model specification

With the g computation example, we have to correctly specify the outcome model. This includes the need to include all exposure-confounder and confounder-confounder interactions. In contrast,

with IP weighting we need not model exposure-confounder interactions. For this reason, IP-weighting is a semiparametric estimator, in contrast to the g computation algorithm, which is fully parametric.
2) positivity

IP-weighted estimators are particularly sensitive to violations of the positivity assumption. This is because the weight is defined as the inverse of the probability of being exposed/unexposed. If that probability is small, then a few people in the analysis will carry much more weight.

The same is not true for the g computation estimator, which is much more robust to positivity violations. However, this does not mean that g computation does not require positivity is met. In the absence of positivity, will extrapolate across regions of the confounding space where there are no data. In such situations, caution is warranted because there is usually no indication that such violations are occurring.

Example 3: ART effect on CD4 Count (Simulated)

We will again re-visit the time-varying data. The causal diagram representing this scenario is depicted in Figure 4.

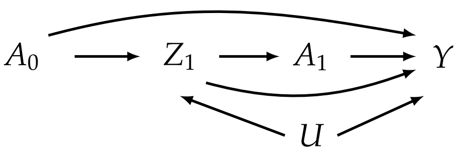

Figure 2: Causal diagram representing the relation between anti-retroviral treatment at time 0 (A_0), HIV viral load just prior to the second round of treatment (Z_1), anti-retroviral treatment status at time 1 (A_1), the CD4 count measured at the end of follow-up (Y), and an unmeasured common cause (U) of HIV viral load and CD4.

Table 1 presents data from a hypothetical observational cohort study ($A = 1$ for treated, $A = 0$ otherwise). Treatment is measured at baseline (A_0) and once during follow up (A_1). The sole covariate is elevated HIV viral load ($Z = 1$ for those with > 200 copies/ml, $Z = 0$ otherwise), which is constant by design at baseline ($Z_0 = 1$) and measured once during follow up just prior to the second treatment (Z_1). The outcome is CD4 count measured at the end of follow up in units of cells/mm^3. Again, the CD4 outcome in Table 1 is summarized (averaged) over the participants at each level of the treatments and covariate.

A_0	Z_1	A_1	Y	N
0	0	0	87.29	209,271
0	0	1	112.11	93,779
0	1	0	119.65	60,654
0	1	1	144.84	136,293
1	0	0	105.28	134,781
1	0	1	130.18	60,789
1	1	0	137.72	93,903
1	1	1	162.83	210,527

Table 1: Prospective study data illustrating the number of subjects (N) within each possible combination of treatment at time 0 (A_0), HIV viral load just prior to the second round of treatment (Z_1), and treatment status for the 2nd round of treatment (A_1). The outcome column (Y) corresponds to the mean of Y within levels of A_0, Z_1, A_1. Note that HIV viral load at baseline is high ($Z_0 = 1$) for everyone by design.

In this particular case, the marginal structural model can be defined as:

$$E(Y^{\bar{a}}) = \psi_t + \psi_1 \text{ cumave}(a)$$

where $\bar{a} = \{a_0, a_1\}$ and where \$ cumave(a) = { a_0+a_1 }/2\$. The reason we use this cumulative average function is because a single unit increase in this exposure (from 0 to 1) yields the following contrast:

$$E(Y^{a_0=1,a_1=1}) - E(Y^{a_0=0,a_1=0}),$$

which is our comparison of interest. Thus the ψ_1 parameter in the above MSM corresponds to our effect of interest.

This parameter can be estimated via inverse probability weighting. To estimate ψ_1 using inverse probability weighted regression, we first obtain the predicted probabilities of the observed treatments. In our example data, there are two possible A_1 values (exposed, unexposed) for each of the four levels in Z_1 and A_0. Additionally, there are two possible A_0 values (exposed, unexposed) overall. This leads to four possible exposure regimes: never treat, treat early only, treat late only, and always treat. For each Z_1 value, we require the predicted probability of the exposure that was actually received. These probabilities are computed by calculating the appropriate proportions of subjects in Table 1. Because there are no variables that affect A_0, this probability is 0.5 for all individuals in the sample. Furthermore, in our example A_1 is not affected by A_0 (Figure 1). Thus, the Z_1 specific probabilities of A_1 are constant across levels of A_0. In settings where A_0 affects A_1, the Z_1 specific probabilities of A_1

would vary across levels of A_0.

In the stratum defined by $Z_1 = 1$, the predicted probabilities of $A_1 = 0$ and $A_1 = 1$ are 0.308 and 0.692, respectively. For example, $(210,527+136,293) / (210,527+136,293+93,903+60,654) = 0.692 $. Thus, the probabilities for each treatment combination are: $0.5 \times 0.308 = 0.155$ (never treated), $0.5 \times 0.308 = 0.155$ (treated early only), $0.5 \times 0.692 = 0.346$ (treated late only), and $0.5 \times 0.692 = 0.346$ (always treated). Dividing the marginal probability of each exposure category (not stratified by Z_1) by these stratum specific probabilities gives stabilized weights of 1.617, 1.617, 0.725, and 0.725, respectively. For example, the never treated weight is $(0.5 \times 0.501)/(0.5 \times 0.308) = 1.617$. The same approach is taken to obtain predicted probabilities and stabilized weights in the stratum defined by $Z_1 = 0$. The weights and weighted data are provided in the following table:

A_0	Z_1	A_1	Y	sw	Pseudo N
0	0	0	87.23	0.72	151222.84
0	0	1	112.23	1.62	151680.46
0	1	0	119.79	1.62	98110.06
0	1	1	144.78	0.72	98789.4
1	0	0	105.25	0.72	97395.08
1	0	1	130.25	1.62	98321.62
1	1	0	137.8	1.62	151884.02
1	1	1	162.8	0.72	152596.51

Table 2: Stabilized inverse probability weights and Pseudo-population obtained by using inverse probability weights.

Fitting a model to the weighted data given in Table 4 can be done in a number of ways. To do this by hand or with the data in wide form, please refer Naimi, Cole, and Kennedy (2017). Here is how to do it in long form:

```r
# first arrange data in long form
a0 <- c(0, 0, 0, 0, 1, 1, 1, 1)
z1 <- c(0, 0, 1, 1, 0, 0, 1, 1)
a1 <- c(0, 1, 0, 1, 0, 1, 0, 1)
y <- c(87.29, 112.11, 119.65, 144.84, 105.28, 130.18, 137.72, 162.83)
N <- c(209271, 93779, 60654, 136293, 134781, 60789, 93903, 210530)
sum(N)
```

```
## [1] 1e+06
```

```
D0 <- NULL
for (i in 1:8) {
    d0 <- data.frame(cbind(rep(0, N[i]), rep(a0[i], N[i]), rep(0, N[i]), rep(y[i], N[i])))
    D0 <- rbind(D0, d0)
}
nrow(D0)
```

```
## [1] 1000000
```

```
D0$id <- 1:nrow(D0)
names(D0) <- c("time", "a", "z", "y", "id")
D1 <- NULL
for (i in 1:8) {
    d1 <- data.frame(cbind(rep(1, N[i]), rep(a1[i], N[i]), rep(z1[i], N[i]), rep(y[i], N[i])))
    D1 <- rbind(D1, d1)
}
nrow(D1)
```

```
## [1] 1000000
```

```
D1$id <- 1:nrow(D1)
names(D1) <- c("time", "a", "z", "y", "id")
D <- rbind(D0, D1)
row.names(D) <- NULL
D <- D[with(D, order(id)), ]

# lag the expsoure by 1 time point
D <- D %>% group_by(id) %>% mutate(a_lag = lag(a, default = 0))

head(D)
```

```
## # A tibble: 6 x 6
## # Groups:   id [3]
##    time     a     z     y    id a_lag
##   <dbl> <dbl> <dbl> <dbl> <int> <dbl>
## 1     0     0     0  87.3     1     0
## 2     1     0     0  87.3     1     0
## 3     0     0     0  87.3     2     0
## 4     1     0     0  87.3     2     0
## 5     0     0     0  87.3     3     0
## 6     1     0     0  87.3     3     0
```

First, we have to estimate the propensity score and create the weights. The numerator for the stabilized weights is just the overall probability of being exposed at each time point.

```
# obtain numerator and denominator for time 1 for time 2, both numerator and denominator
# are .5, so sw=1
den <- glm(a ~ time + z + a_lag, data = D, family = binomial("logit"))$fitted.values
num <- glm(a ~ time, data = D, family = binomial("logit"))$fitted.values

# create stabilized weight
D$sw0 <- D$a * (num/den) + (1 - D$a) * ((1 - num)/(1 - den))
# multiply the weights across time
D <- D %>% group_by(id) %>% mutate(sw = cumprod(sw0))

# check weight distribution
summary(D$sw)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7224  0.7246  1.0000  1.0000  1.0000  1.6184
```

```
# create the cumulative average expsoure
D <- D %>% group_by(id) %>% mutate(cum_a = cumsum(a)/2)
```

Now all we have to do is fit the model with these weights:

```
ipw_msm <- lm(y ~ time + cum_a, data = D, weights = sw)

coef(ipw_msm)[3]
```

```
##    cum_a
## 51.39741
```

which is the same effect we estimated using the g formula in the previous section.

Weighting the observed data by the inverse of the probability of the observed exposure yields a “pseudo-population” (as in Table 4) in which treatment at the second time point (A_1) is no longer related to (and is thus no longer confounded by) viral load just prior to the second time point (Z_1). Thus, weighting a conditional regression model for the outcome by the inverse probability of treatment enables us to account for the fact that Z_1 both confounds A_1 and is affected by A_0.

References

Austin, Peter C. 2016. "Variance Estimation When Using Inverse Probability of Treatment Weighting (Iptw) with Survival Analysis." *Statistics in Medicine* 35 (30). John Wiley; Sons Inc.: 5642–55. doi:10.1002/sim.7084.

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