A guided tour in targeted learning territory

David Benkeser, Antoine Chambaz, Nima Hejazi

08/20/2018

Contents

1 Introduction

This is a very first draft of our article. The current *tentative* title is "A guided tour in targeted learning territory".

Explain our objectives and how we will meet them. Explain that the symbol indicates more delicate material.

Use sectioning a lot to ease cross-referencing.

Do we include exercises? I propose we do, and to flag the corresponding subsections with symbol .

```r
set.seed(54321) ## because reproducibility matters...
suppressMessages(library(R.utils)) ## make sure it is installed
suppressMessages(library(tidyverse)) ## make sure it is installed
suppressMessages(library(caret)) ## make sure it is installed
expit <- plogis
logit <- qlogis
```

Function `expit` implements the link function $\mathrm{expit} : \mathbb{R} \to]0,1[$ given by $\mathrm{expit}(x) \equiv (1+e^{-x})^{-1}$. Function `logit` implements its inverse function $\mathrm{logit} :]0,1[\to \mathbb{R}$ given by $\mathrm{logit}(p) \equiv \log[p/(1-p)]$.

2 A simulation study

blabla

2.1 Reproducible experiment as a law. We are interested in a reproducible experiment. The generic summary of how one realization of the experiment unfolds, our observation, is called O. We view O as a random variable drawn from what we call the law P_0 of the experiment. The law P_0 is viewed as an element of what we call the model. Denoted by $\mathcal{M}$, the model is the collection of *all* laws from which O can be drawn and that meet some constraints. The constraints translate the knowledge we have about the experiment. The more we know about the experiment, the smaller is $\mathcal{M}$. In all our examples, model $\mathcal{M}$ will put very few restrictions on the candidate laws.

Consider the following chunk of code:

```
draw_from_experiment <- function(n, ideal = FALSE) {
  ## preliminary
  n <- Arguments$getInteger(n, c(1, Inf))
  ideal <- Arguments$getLogical(ideal)
  ## ## 'Gbar' and 'Qbar' factors
  Gbar <- function(W) {
    expit(1 + 2 * W - 4 * sqrt(abs((W - 5/12))))
  }
  Qbar <- function(AW) {
    A <- AW[, 1]
    W <- AW[, 2]
    ## A * (cos((1 + W) * pi / 4) + (1/3 <= W & W <= 1/2) / 5) +
    ##  (1 - A) * (sin(4 * W^2 * pi) / 4 + 1/2)
    A * (cos((-1/2 + W) * pi) * 2/5 + 1/5 + (1/3 <= W & W <= 1/2) / 5 +
         (W >= 3/4) * (W - 3/4) * 2) +
      (1 - A) * (sin(4 * W^2 * pi) / 4 + 1/2)
  }
  ## sampling
  ## ## context
  mixture_weights <- c(1/10, 9/10, 0)
  mins <- c(0, 11/30, 0)
  maxs <- c(1, 14/30, 1)
  latent <- findInterval(runif(n), cumsum(mixture_weights)) + 1
  W <- runif(n, min = mins[latent], max = maxs[latent])
  ## ## counterfactual rewards
  zeroW <- cbind(A = 0, W)
  oneW <- cbind(A = 1, W)
  Qbar.zeroW <- Qbar(zeroW)
  Qbar.oneW <- Qbar(oneW)
  Yzero <- rbeta(n, shape1 = 2, shape2 = 2 * (1 - Qbar.zeroW) / Qbar.zeroW)
  Yone <- rbeta(n, shape1 = 3, shape2 = 3 * (1 - Qbar.oneW) / Qbar.oneW)
  ## ## action undertaken
  A <- rbinom(n, size = 1, prob = Gbar(W))
  ## ## actual reward
  Y <- A * Yone + (1 - A) * Yzero
  ## ## observation
  if (ideal) {
    obs <- cbind(W = W, Yzero = Yzero, Yone = Yone, A = A, Y = Y)
```

```r
  } else {
    obs <- cbind(W = W, A = A, Y = Y)
  }
  attr(obs, "Gbar") <- Gbar
  attr(obs, "Qbar") <- Qbar
  attr(obs, "QW") <- function(W) {
    out <- sapply(1:length(mixture_weights),
                  function(ii){
                    mixture_weights[ii] *
                      dunif(W, min = mins[ii], max = maxs[ii])
                  })
    return(rowSums(out))
  }
  attr(obs, "qY") <- function(AW, Y, Qbar){
    A <- AW[, 1]
    W <- AW[, 2]
    Qbar.AW <- do.call(Qbar, list(AW)) # is call to 'do.call' necessary?
    shape1 <- ifelse(A == 0, 2, 3)
    dbeta(Y, shape1 = shape1, shape2 = shape1 * (1 - Qbar.AW) / Qbar.AW)
  }
  ##
  return(obs)
}
```

We can interpret `draw_from_experiment` as a law P_0 since we can use the function to sample observations from a common law. It is even a little more than that, because we can tweak the experiment, by setting its `ideal` argument to `TRUE`, in order to get what appear as intermediary (counterfactual) variables in the regular experiment. The next chunk of code runs the (regular) experiment five times independently and outputs the resulting observations:

```r
(five_obs <- draw_from_experiment(5))
```

```
##              W A         Y
## [1,] 0.4533028 0 0.8979460
## [2,] 0.3716077 0 0.9905312
## [3,] 0.3875802 0 0.8080567
## [4,] 0.4008279 1 0.9954100
## [5,] 0.4038325 0 0.9772926
## attr(,"Gbar")
## function (W)
## {
##     expit(1 + 2 * W - 4 * sqrt(abs((W - 5/12))))
## }
## <bytecode: 0x7fc90a711b28>
## <environment: 0x7fc90a151620>
## attr(,"Qbar")
## function (AW)
## {
##     A <- AW[, 1]
##     W <- AW[, 2]
##     A * (cos((-1/2 + W) * pi) * 2/5 + 1/5 + (1/3 <= W & W <=
##         1/2)/5 + (W >= 3/4) * (W - 3/4) * 2) + (1 - A) * (sin(4 *
##         W^2 * pi)/4 + 1/2)
```

```
## }
## <bytecode: 0x7fc90bc3f8c0>
## <environment: 0x7fc90a151620>
## attr(,"QW")
## function (W)
## {
##     out <- sapply(1:length(mixture_weights), function(ii) {
##         mixture_weights[ii] * dunif(W, min = mins[ii], max = maxs[ii])
##     })
##     return(rowSums(out))
## }
## <bytecode: 0x7fc90d526860>
## <environment: 0x7fc90a151620>
## attr(,"qY")
## function (AW, Y, Qbar)
## {
##     A <- AW[, 1]
##     W <- AW[, 2]
##     Qbar.AW <- do.call(Qbar, list(AW))
##     shape1 <- ifelse(A == 0, 2, 3)
##     dbeta(Y, shape1 = shape1, shape2 = shape1 * (1 - Qbar.AW)/Qbar.AW)
## }
## <bytecode: 0x7fc90d942190>
## <environment: 0x7fc90a151620>
```

We can view the `attributes` of object `five_obs` because, in this section, we act as oracles, *i.e.*, we know completely the nature of the experiment. In particular, we have included several features of P_0 that play an important role in our developments. The attribute `QW` describes the density of W, of which the law $Q_{0,W}$ is a mixture of the uniform laws over $[0,1]$ (weight $1/10$) and $[11/30, 14/30]$ (weight $9/10$).[*] The attribute `Gbar` describes the conditional probability of action $A = 1$ given W. For each $a \in \{0,1\}$, we denote $\bar{G}_0(W) \equiv \Pr_{P_0}(A = 1|W)$ and $\ell\bar{G}_0(a, W) \equiv \Pr_{P_0}(A = a|W)$. The attribute `qY` describes the conditional density of Y given A and W. For each $y \in]0,1[$, we denote by $q_{0,Y}(y, A, W)$ the conditional density evaluated at y of Y given A and W. Similarly, the attribute `Qbar` describes the conditional mean of Y given A and W, and we denote $\bar{Q}_0(A, W) = \mathrm{E}_{P_0}(Y|A, W)$ the conditional mean of Y given A and W.

2.2 Visualizing infinite-dimensional features of the experiment

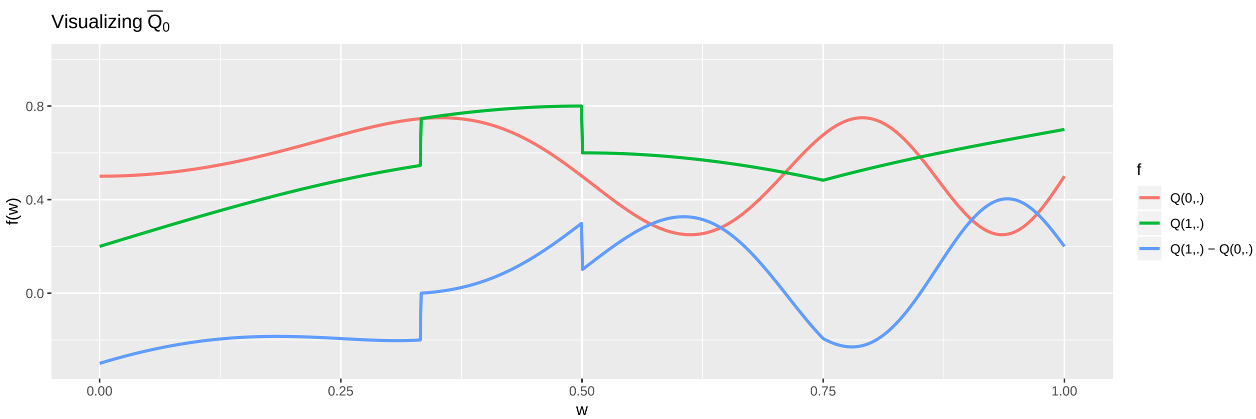
1. Run the following chunk of code. It visualizes the conditional mean $\bar{Q}_0$.

```
Gbar <- attr(five_obs, "Gbar")
Qbar <- attr(five_obs, "Qbar")
QW <- attr(five_obs, "QW")

features <- tibble(w = seq(0, 1, length.out = 1e3)) %>%
  mutate(Qw = QW(w),
         Gw = Gbar(w),
         Q1w = Qbar(cbind(A = 1, W = w)),
         Q0w = Qbar(cbind(A = 0, W = w)),
         blip_Qw = Q1w - Q0w)
```

[*] We fine-tuned (or tweaked, or something else?) the marginal law of W to make it easier later on to drive home important messages. Specifically, . . . (do we explain what happens?)

```
features %>% select(-Qw, -Gw) %>%
  rename("Q(1,.)" = Q1w,
         "Q(0,.)" = Q0w,
         "Q(1,.) - Q(0,.)" = blip_Qw) %>%
  gather("f", "value", -w) %>%
  ggplot() +
  geom_line(aes(x = w, y = value, color = f), size = 1) +
  labs(y = "f(w)", title = bquote("Visualizing" ~ bar(Q)[0])) +
  ylim(NA, 1)
```



2. Adapt the above chunk of code to visualize the marginal density $Q_{0,W}$ and conditional probability $\bar{G}_0$.

2.3 The parameter of interest, first pass. It happens that we especially care for a finite-dimensional feature of P_0 that we denote by ψ_0. Its definition involves two of the aforementioned infinite-dimensional features:

$$\begin{aligned}\psi_0 &\equiv \int \left(\bar{Q}_0(1,w) - \bar{Q}_0(0,w)\right) dQ_{0,W}(w) \qquad (1)\\ &= \mathrm{E}_{P_0}\left(\mathrm{E}_{P_0}(Y \mid A=1,W) - \mathrm{E}_{P_0}(Y \mid A=0,W)\right).\end{aligned}$$

Acting as oracles, we can compute explicitly the numerical value of ψ_0.

```
integrand <- function(w) {
  Qbar <- attr(five_obs, "Qbar")
  QW <- attr(five_obs, "QW")
  ( Qbar(cbind(1, w)) - Qbar(cbind(0, w)) ) * QW(w)
}
(psi_zero <- integrate(integrand, lower = 0, upper = 1)$val)
```

```
## [1] 0.08317711
```

Our interest in ψ_0 is of causal nature. Taking a closer look at `drawFromExperiment` reveals indeed that the random making of an observation O drawn from P_0 can be summarized by the following causal graph and nonparametric system of structural equations:

```
## plot the causal diagram
```

and, for some deterministic functions f_w, f_a, f_y and independent sources of randomness U_w, U_a, U_y,

1. sample the context where the rest of the experiment will take place Not sure I understand this sentence, $W = f_w(U_w)$;

2. sample the two counterfactual rewards of the two actions that can be undertaken, $Y_0 = f_y(0, W, U_y)$ and $Y_1 = f_y(1, W, U_y)$;

3. sample which action is carried out in the given context, $A = f_a(W, U_a)$;

4. define the corresponding reward, $Y = AY_1 + (1 - A)Y_0$;

5. summarize the course of the experiment with the observation $O = (W, A, Y)$, thus concealing Y_0 and Y_1.

The above description of the experiment `draw_from_experiment` is useful to reinforce what it means to run the "ideal" experiment by setting argument `ideal` to `TRUE` in a call to `draw_from_experiment`. Doing so triggers a modification of the nature of the experiment, enforcing that the counterfactual rewards Y_0 and Y_1 be part of the summary of the experiment eventually. In light of the above enumeration, $\mathbb{O} \equiv (W, Y_0, Y_1, A, Y)$ is output, as opposed to its summary measure O. This defines another experiment and its law, that we denote $\mathbb{P}_0$.

It is straightforward to show[†] that under assumptions[‡]

$$\psi_0 = \mathrm{E}_{\mathbb{P}_0}(Y_1 - Y_0) = \mathrm{E}_{\mathbb{P}_0}(Y_1) - \mathrm{E}_{\mathbb{P}_0}(Y_0). \tag{2}$$

Thus, under these assumptions, ψ_0 describes the average difference in of the two counterfactual rewards. In other words, ψ_0 quantifies the difference in average of the reward one would get in a world where one would always enforce action $a = 1$ with the reward one would get in a world where one would always enforce action $a = 0$. This said, it is worth emphasizing that ψ_0 is a well-defined parameter beyond its causal interpretation and describes a standardized association between the action A and the reward Y.

To conclude this subsection, we use our position as oracles to sample observations from the ideal experiment. We call `draw_from_experiment` with its argument `ideal` set to `TRUE` in order to numerically approximate ψ_0. By the law of large numbers, the following code approximates ψ_0 and shows it approximate value.

```
B <- 1e6 ## Antoine: 1e6 eventually
ideal_obs <- draw_from_experiment(B, ideal = TRUE)
(psi_approx <- mean(ideal_obs[, "Yone"] - ideal_obs[, "Yzero"]))
```

```
## [1] 0.08293116
```

The object `psi_approx` contains an approximation to ψ_0 based on `B` observations from the ideal experiment. The random sampling of observations results in uncertainty in the numerical approximation of ψ_0. This uncertainty can be quantified by constructing a 95% confidence interval for ψ_0. The central limit theorem and Slutsky's lemma allow us to build such an interval as follows.

[†]For $a = 0, 1$,

$$\begin{aligned} E(Y_a) &= \int \mathrm{E}_{\mathbb{P}_0}(Y_a \mid W = w) dQ_{0,W}(w) = \int \mathrm{E}_{\mathbb{P}_0}(Y_a \mid A = a, W = w) dQ_{0,W}(w) \\ &= \int \mathrm{E}_{P_0}(Y \mid A = a, W = w) dQ_{0,W}(w) = \int \bar{Q}_0(a, W) dQ_{0,W}(w). \end{aligned}$$

[‡]The second equality in the above footnote is the result of the randomization assumption, which states that $(Y_0, Y_1) \perp A \mid W$. The third equality results from two assumptions: consistency and positivity. Consistency states that $Y_a = Y \mid A = a$; that is, the counterfactual of interest Y_a is equal to the observed value Y when $A = a$. Positivity states that for $a = 0, 1$, $\Pr_{P_0}\{\ell\bar{G}_0(a, W) > 0\} = 1$, and is needed for $\mathrm{E}_{P_0}(Y \mid A = a, W = w)$ to be well-defined.

```r
sd_approx <- sd(ideal_obs[, "Yone"] - ideal_obs[, "Yzero"])
alpha <- 0.05
(psi_approx_CI <- psi_approx + c(-1, 1) * qnorm(1 - alpha / 2) * sd_approx / sqrt(B))
```

```
## [1] 0.08232477 0.08353754
```

We note that the interpretation of this confidence interval is that in 95% of draws of size `B` from the ideal data generating experiment, the true value of ψ_0 will be contained in the generated confidence interval.

2.4 Difference in covariate-adjusted quantile rewards, first pass.

The problems come within the scope of Sections 2.3.

As discussed above, parameter ψ_0 (2) is the difference in average rewards if we enforce action $a = 1$ rather than $a = 0$. An alternative way to describe the rewards under different actions involves quantiles as opposed to averages.

Let $Q_{0,Y}(y, a, w) = \int_0^y q_{0,Y}(u, a, w)du$ be the conditional cumulative distribution of reward Y given $A = a$ and $W = w$, evaluated at $y \in]0, 1[$, that is implied by P_0. For each action $a \in \{0, 1\}$ and $c \in]0, 1[$, introduce

$$\gamma_{0,a,c} \equiv \inf \left\{ y \in]0, 1[: \int Q_{0,Y}(y, a, w) dQ_{0,W}(w) \geq c \right\}. \tag{3}$$

It is not difficult to check that do we give the proof or refer to other articles?

$$\gamma_{0,a,c} = \inf \left\{ y \in]0, 1[: \Pr_{\mathbb{P}_0}(Y_a \leq y) \geq c \right\}.$$

Thus, $\gamma_{0,a,c}$ can be interpreted as a covariate-adjusted c-th quantile reward when action a is enforced. The difference

$$\delta_{0,c} \equiv \gamma_{0,1,c} - \gamma_{0,0,c}$$

is the c-th quantile counterpart to parameter ψ_0 (2).

1. Compute the numerical value of $\gamma_{0,a,c}$ for each $(a, c) \in \{0, 1\} \times \{1/4, 1/2, 3/4\}$ using the appropriate attributes of `five_obs`. Based on these results, report the numerical value of $\delta_{0,c}$ for each $c \in \{1/4, 1/2, 3/4\}$.

2. Approximate the numerical values of $\gamma_{0,a,c}$ for each $(a, c) \in \{0, 1\} \times \{1/4, 1/2, 3/4\}$ by drawing a large sample from the "ideal" data experiment and using empirical quantile estimates. Deduce from these results a numerical approximation to $\delta_{0,c}$ for $c \in \{1/4, 1/2, 3/4\}$. Confirm that your results closely match those obtained in the previous problem.

2.5 The parameter of interest, second pass.

Suppose we know beforehand that O drawn from P_0 takes its values in $\mathcal{O} \equiv [0, 1] \times \{0, 1\} \times [0, 1]$ and that $\bar{G}(W) = P_0(A = 1|W)$ is bounded away from zero and one $Q_{0,W}$-almost surely (this is the case indeed). Then we can define model $\mathcal{M}$ as the set of all laws P on $\mathcal{O}$ such that $\bar{G}(W) \equiv P(A = 1|W)$ is bounded away from zero and one Q_W-almost surely, where Q_W is the marginal law of W under P.

Let us also define generically $\bar{Q}$ as

$$\bar{Q}(A, W) \equiv \mathrm{E}_P(Y|A, W).$$

Central to our approach is viewing ψ_0 as the value at P_0 of the statistical mapping Ψ from $\mathcal{M}$ to $[0, 1]$ characterized by

$$\begin{aligned}\Psi(P) &\equiv \mathrm{E}_P\left(\mathrm{E}_P(Y \mid A=1, W) - \mathrm{E}_P(Y \mid A=0, W)\right) \\ &= \int \left(\bar{Q}(1, w) - \bar{Q}(0, w)\right) dQ_W(w),\end{aligned}$$

a clear extension of (1). For instance, although the law $\Pi_0 \in \mathcal{M}$ encoded by default (*i.e.*, with `h=0`) in `drawFromAnotherExperiment` defined below differs starkly from P_0,

```
draw_from_another_experiment <- function(n, h = 0) {
  ## preliminary
  n <- Arguments$getInteger(n, c(1, Inf))
  h <- Arguments$getNumeric(h)
  ## ## 'Gbar' and 'Qbar' factors
  Gbar <- function(W) {
    sin((1 + W) * pi / 6)
  }
  Qbar <- function(AW, hh = h) {
    A <- AW[, 1]
    W <- AW[, 2]
    expit( logit( A *  W + (1 - A) * W^2 ) +
           hh * 10 * sqrt(W) * A )
  }
  ## sampling
  ## ## context
  W <- runif(n, min = 1/10, max = 9/10)
  ## ## action undertaken
  A <- rbinom(n, size = 1, prob = Gbar(W))
  ## ## reward
  shape1 <- 4
  QAW <- Qbar(cbind(A, W))
  Y <- rbeta(n, shape1 = shape1, shape2 = shape1 * (1 - QAW) / QAW)
  ## ## observation
  obs <- cbind(W = W, A = A, Y = Y)
  attr(obs, "Gbar") <- Gbar
  attr(obs, "Qbar") <- Qbar
  attr(obs, "QW") <- function(x){dunif(x, min = 1/10, max = 9/10)}
  attr(obs, "shape1") <- shape1
  attr(obs, "qY") <- function(AW, Y, Qbar, shape1){
    A <- AW[,1]; W <- AW[,2]
    Qbar.AW <- Qbar(AW)
    dbeta(Y, shape1 = shape1, shape2 = shape1 * (1 - Qbar.AW) / Qbar.AW)
  }
  ##
  return(obs)
}
```

the parameter $\Psi(\Pi_0)$ is well defined, and numerically approximated by `psi_Pi_zero` as follows.

```
five_obs_from_another_experiment <- draw_from_another_experiment(5)
another_integrand <- function(w) {
  Qbar <- attr(five_obs_from_another_experiment, "Qbar")
  QW <- attr(five_obs_from_another_experiment, "QW")
  ( Qbar(cbind(1, w)) - Qbar(cbind(0, w)) ) * QW(w)
```

```r
}
(psi_Pi_zero <- integrate(another_integrand, lower = 0, upper = 1)$val)
```

```
## [1] 0.1966687
```

Straightforward algebra confirms that indeed $\Psi(\Pi_0) = 59/300$.

2.6 ⚙ Difference in covariate-adjusted quantile rewards, second pass. We continue with the exercise from Section 2.4. The problems come within the scope of Section 2.3.

As above, we define $q_Y(y, a, w)$ to be the (A, W)-conditional density of Y given $A = a$ and $W = w$, evaluated at y, that is implied by a generic $P \in \mathcal{M}$. Similarly, we use Q_Y to denote the corresponding cumulative distribution function. The covariate-adjusted c-th quantile reward for action $a \in \{0, 1\}$ may be viewed as a mapping $\Gamma_{a,c}$ from $\mathcal{M}$ to $[0, 1]$ characterized by

$$\Gamma_{a,c}(P) = \inf \left\{ y \in]0, 1[: \int Q_Y(y, a, w) dQ_W(w) \geq c \right\}.$$

The difference in c-th quantile rewards may similarly be viewed as a mapping Δ_c from $\mathcal{M}$ to $[0, 1]$, characterized by $\Delta_c(P) \equiv \Gamma_{1,c}(P) - \Gamma_{0,c}(P)$.

1. Compute the numerical value of $\Gamma_{a,c}(\Pi_0)$ for $(a, c) \in \{0, 1\} \times \{1/4, 1/2, 3/4\}$ using the appropriate attributes of `five_obs_from_another_experiment`. Based on these results, report the numerical value of $\Delta_c(\Pi_0)$ for each $c \in \{1/4, 1/2, 3/4\}$.

2. Approximate the value of $\Gamma_{0,a,c}(\Pi_0)$ for $(a, c) \in \{0, 1\} \times \{1/4, 1/2, 3/4\}$ by drawing a large sample from the "ideal" data experiment and using empirical quantile estimates. Deduce from these results a numerical approximation to $\Delta_{0,c}(\Pi_0)$ for each $c \in \{1/4, 1/2, 3/4\}$. Confirm that your results closely match those obtained in the previous problem.

3. Building upon the code you wrote to solve the previous problem, construct a confidence interval with asymptotic level 95% for $\Delta_{0,c}(\Pi_0)$, with $c \in \{1/4, 1/2, 3/4\}$.[§]

2.7 The parameter of interest, third pass. In the previous subsection, we reoriented our view of the target parameter to be a statistical functional of the distribution of the observed data. Specifically, we viewed the parameter as a function of specific features of the observed data distribution, namely Q_W and $\bar{Q}$. It is straightforward[¶] to show an equivalent representation of the parameter as

$$\psi_0 = \int \left(\frac{I(a=1)}{\ell\bar{G}_0(1, w)} - \frac{I(a=0)}{\ell\bar{G}_0(0, w)} \right) y \; dP_0(o) \tag{4}$$

[§]Let $X_1, \ldots, X_n$ be independently drawn from a continuous distribution function F. Set $p \in]0, 1[$ and, assuming that n is large, find $k \geq 1$ and $l \geq 1$ such that $k/n \approx p - \Phi^{-1}(1-\alpha)\sqrt{p(1-p)/n}$ and $l/n \approx p + \Phi^{-1}(1-\alpha)\sqrt{p(1-p)/n}$, where Φ is the standard normal distribution function. Then $[X_{(k)}, X_{(l)}]$ is a confidence interval for $F^{-1}(p)$ with asymptotic level $1 - 2\alpha$.

[¶] We temporarily drop the subscript P_0 to save space. For $a = 0, 1$,

$$\mathrm{E}\left(\frac{I(A=a)Y}{\ell\bar{G}(a,W)}\right) = \mathrm{E}\left\{\mathrm{E}\left(\frac{I(A=a)Y}{\ell\bar{G}(a,W)}\middle| A, W\right)\right\} = \mathrm{E}\left\{\frac{I(A=a)}{\ell\bar{G}(a,W)}\mathrm{E}(Y \mid A, W)\right\} = \mathrm{E}\left\{\frac{I(A=a)}{\ell\bar{G}(a,W)}\mathrm{E}(Y \mid A=a, W)\right\}$$
$$= \mathrm{E}\left\{\mathrm{E}\left(\frac{I(A=a)}{\ell\bar{G}(a,W)}\mathrm{E}(Y \mid A=a, W) \mid W\right)\right\} = \mathrm{E}\left(\frac{\ell\bar{G}(a,W)}{\ell\bar{G}(a,W)}\mathrm{E}(Y \mid A=a, W) \mid W\right) = \mathrm{E}\left(\mathrm{E}(Y \mid A=a, W)\right).$$

$$= \mathrm{E}_{P_0}\left\{\left(\frac{I(A=1)}{\mathrm{Pr}_{P_0}(A=1\mid W)} - \frac{I(A=0)}{\mathrm{Pr}_{P_0}(A=0\mid W)}\right)Y\right\}$$

or, viewing the parameter as a statistical functional, for a given P in the model

$$\Psi(P) = \mathrm{E}_{P}\left\{\left(\frac{I(A=1)}{\mathrm{Pr}_{P}(A=1\mid W)} - \frac{I(A=0)}{\mathrm{Pr}_{P}(A=1\mid W)}\right)Y\right\} \quad (5)$$

$$= \int \left(\frac{I(a=1)}{\ell\bar{G}(1,w)} - \frac{I(a=0)}{\ell\bar{G}(0,w)}\right) y \; dP(o) \quad (6)$$

Our reason for introducing this alternative view of the target parameter will become clear when we discuss estimation of the target parameter. Specifically, the representations (1) and (4) naturally suggest different estimation strategies for ψ_0. The former suggests building an estimator of ψ_0 using estimators of $\bar{Q}_0$ and of $Q_{W,0}$. The latter suggests building an estimator of ψ_0 using estimators of $\ell\bar{G}_0$ and of P_0. We return to these ideas in later sections.

2.8 Difference in covariate-adjusted quantile rewards, third pass.

We continue with the exercise from Section 2.4.

1. Show that for $a' = 0, 1$, $\gamma_{0,a',c}$ as defined in (3) can be equivalently expressed as

$$\inf\left\{z \in]0,1[\, : \int \frac{I(a=a')}{\ell\bar{G}(a',W)} I(y \leq z) dP_0(o) \geq c\right\}.$$