R Lab 5 - TMLE

Introduction to Causal Inference

Goals:
1. Review the causal roadmap.
2. Code TMLE for the G-computation estimand.
3. Understand the basics of the `ltmle` package.
4. Use the `ltmle` package to explore the double robustness of TMLE.

Next lab:
We will implement the non-parametric bootstrap to estimate the standard error of the estimators. We will also use the sample variance of the estimated influence curve to obtain inference for TMLE.

1 Background

Dr. Alan Grant: "T-Rex doesn't want to be fed. He wants to hunt. Can't just suppress 65 million years of gut instinct." - Michael Crichton

We are interested in estimating the causal effect of prior experience with Dinosaurs on injury severity on Isla Nublar, the location of the InGen lab. Suppose we have data on the following variables:

- $W1$: gender (1 for male; 0 for female)
- $W2$: intelligence (scale from 0 to 1; with higher values for smarter)
- $W3$: handy/inventiveness (continuous and scaled; with larger, positive values for more MacGyver-ness)
- $W4$: running speed (continuous and scaled; with larger, positive values for faster)
- A: prior Dinosaur experience (1 for yes; 0 for no)
- Y: seriousness of injury (scale from 0 to 1; with higher values for more severe)

Let $W = (W1, W2, W3, W4)$ be the vector of baseline covariates.

http://www.thesambarnes.com/web-project-management/account-management-for-the-web-project-manager-part-1/

2 Causal Roadmap Rundown

1. **Specify the Question:**
What is the causal effect of prior experience on injury severity in Jurassic Park?

2. **Specify the structural causal model (SCM) $\mathcal{M}^{\mathcal{F}}$:**
- Endogenous nodes: $X = (W, A, Y)$, where $W = (W1, W2, W3, W4)$ is the set of baseline covariates (gender, intelligence, MacGyver-ness, running speed), A is prior Dinosaur experience, and Y is injury severity. For simplicity, we have condensed the baseline characteristics into a single node.
- Background variables (Exogenous nodes): $U = (U_W, U_A, U_Y) \sim \mathbb{P}_U$. We place no assumptions on the distribution $\mathbb{P}_U$.
- Structural equations $\mathcal{F}$:
$$\begin{aligned} W &= f_W(U_W) \\ A &= f_A(W, U_A) \\ Y &= f_Y(W, A, U_Y) \end{aligned}$$
We have not placed any restrictions on the functional forms.

3. **Specify the causal parameter of interest:**
We are interested in the causal effect of prior Dinosaur experience on expected injury severity on Isla Nublar (i.e. the average treatment effect):
$$\Psi^{\mathcal{F}}(\mathbb{P}_{U,X}) = \mathbb{E}_{U,X}(Y_1) - \mathbb{E}_{U,X}(Y_0)$$
where Y_a is the counterfactual outcome (injury severity), if possibly contrary to fact, the subject had Dinosaur-experience $A = a$.

4. **Specify the link between the structural causal model (SCM) and the observed data:**
We assume that the observed data $O = (W, A, Y) \sim \mathbb{P}_0$ were generated by sampling n times from a data generating described by the SCM. The statistical model $\mathcal{M}$ for the set of allowed distributions of the observed data is non-parametric.

5. **Assess identifiability:**
In the original structural causal model $\mathcal{M}^{\mathcal{F}}$, the target causal parameter is not identified from the observed data distribution. We need make assumptions about the independence of the background factors: $U_A \perp\!\!\!\perp U_Y$ and (i) $U_A \perp\!\!\!\perp U_W$, or (ii) $U_Y \perp\!\!\!\perp U_W$. Then the backdoor criteria will hold conditionally on the covariates $W = (W1, W2, W3, W4)$. We use $\mathcal{M}^{\mathcal{F}^*}$ to denote the original SCM augmented by the convenience-based assumptions needed for identifiability.

To identify $\mathbb{E}_{U,X}(Y_a)$ with the G-Computation formula, we also need the positivity assumption to hold
$$min_{a \in \mathcal{A}} \ \mathbb{P}_0(A = a | W = w) > 0$$
for all w for which $\mathbb{P}_0(W = w) > 0$. In terms of our example, there must be a positive probability of being dinosaur-experienced and not being dinosaur-experienced within strata of baseline covariates.

6. **Specify the statistical estimand:**
The target parameter of the observed data distribution (which equals the causal parameter in the augmented causal model $\mathcal{M}^{\mathcal{F}^*}$) is given by the G-Computation formula:
$$\begin{aligned} \Psi(\mathbb{P}_0) &= \mathbb{E}_0\big[\mathbb{E}_0(Y|A=1,W) - \mathbb{E}_0(Y|A=0,W)\big] \\ &= \sum_w \big[\bar{Q}_0(1,w) - \bar{Q}_0(0,w)\big]\mathbb{P}_0(W=w) \end{aligned}$$
This is our statistical estimand.

7. **Estimate the chosen parameter of the observed data distribution:**

(a) **Simple substitution estimator based on the G-Computation formula:**

$$\hat{\Psi}_{SS}(\hat{\mathbb{P}}) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\bar{Q}}(1,W_i) - \hat{\bar{Q}}(0,W_i)\right)$$

where $\hat{\mathbb{P}}$ is the empirical distribution and $\hat{\bar{Q}}(A,W)$ is the estimate of the conditional mean outcome given the exposure (experience with Dinosaurs or not) and baseline covariates $\bar{Q}_0(A,W) \equiv \mathbb{E}_0(Y|A,W)$.
- Consistency of the simple (non-targeted) substitution estimator depends on consistent estimation of the conditional mean outcome $\bar{Q}_0(A,W)$.

(b) **Standard (unstabilized) inverse probability weighted estimator (IPTW)**:

$$\hat{\Psi}_{IPTW}(\hat{\mathbb{P}}) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\mathbb{I}(A_i=1)}{\hat{g}(1|W_i)} - \frac{\mathbb{I}(A_i=0)}{\hat{g}(0|W_i)}\right]Y_i$$

where $\hat{g}(1|W_i) = \hat{\mathbb{P}}(A_i = 1|W_i)$ is an estimate of the exposure mechanism (i.e. the conditional probability of having Dinosaur experience, given the baseline covariates).
- Consistency of IPTW estimators depends on consistent estimation of the exposure mechanism $g_0(1|W) \equiv \mathbb{P}_0(A = 1|W)$.

(c) **Targeted maximum likelihood estimation (TMLE):**

$$\hat{\Psi}_{TMLE}(\hat{\mathbb{P}}) = \frac{1}{n}\sum_{i=1}^{n}\left(\bar{Q}_n^*(1,W_i) - \bar{Q}_n^*(0,W_i)\right)$$

where $\bar{Q}_n^*(A,W)$ denotes the targeted estimate of the conditional mean outcome, given the exposure and baseline covariates $\bar{Q}_0(A,W)$.
- Implementation requires estimation of both the conditional mean function $\bar{Q}_0(A,W)$ and the exposure mechanism $g_0(A|W)$.
- Double robust estimators are consistent if either $\bar{Q}_0(A,W)$ or $g_0(A|W)$ is estimated consistently.
- If both $\bar{Q}_0(A,W)$ and $g_0(A|W)$ are estimated consistently (and at reasonable rates), TMLE will be efficient and achieve the lowest possible asymptotic variance over a large class of estimators.
- These asymptotic properties describe what happens when sample size goes to infinity and also translate into lower bias and variance in finite samples.

If we apply an estimator to our observed data (n i.i.d. copies of O drawn from $\mathbb{P}_0$), we get an estimate (a number). The estimator is function of random variables; so it is a random variable. It has a distribution, which we can study theoretically or using simulations.

Note: An estimator is *consistent* if the point estimates converge (in probability) to the estimand as sample size $n \to \infty$.

8. **Inference and interpret results:**
In the next lab, we will implement the non-parametric bootstrap for variance estimation for the three types of estimators. We will use the sample variance of the estimated influence curve to obtain inference for the TMLE.

3 Import and explore data set `RLab5.TMLE.csv`.

1. Set the seed to 252.
2. Use the `read.csv` function to import the dataset and assign it to dataframe `ObsData`.
3. Use the `head` and `summary` functions to explore the data.

4. Use the `nrow` function to count the number of subjects in the data set. Assign this number as `n`.

Solution:

```
> set.seed(252)

> #  Import the data set and assign it to object ObsData; explore
> ObsData<- read.csv("RLab5.TMLE.csv")
> names(ObsData)

[1] "W1" "W2" "W3" "W4" "A"  "Y"

> head(ObsData)

  W1         W2          W3          W4 A           Y
1  0 0.53080879  1.33425653  0.84889241 1 0.002010378
2  0 0.68486090  1.59585299  0.54299062 0 0.001031028
3  1 0.38328339 -1.28106043 -0.39391379 0 0.591290943
4  1 0.95498800  0.06046723  0.34488307 0 0.500392409
5  0 0.11835658  0.08203773  0.05144746 1 0.371959296
6  1 0.03910006 -1.78980628 -1.40836264 0 0.078282481

> summary(ObsData)

       W1             W2                    W3                  W4
 Min.   :0.00   Min.   :0.0006053   Min.   :-3.26337   Min.   :-3.08419
 1st Qu.:0.00   1st Qu.:0.2282640   1st Qu.:-0.67585   1st Qu.:-0.69567
 Median :0.00   Median :0.4809893   Median :-0.06790   Median : 0.01021
 Mean   :0.48   Mean   :0.4902962   Mean   :-0.01983   Mean   :-0.01145
 3rd Qu.:1.00   3rd Qu.:0.7570638   3rd Qu.: 0.70675   3rd Qu.: 0.68237
 Max.   :1.00   Max.   :0.9988775   Max.   : 3.57008   Max.   : 3.43078
       A               Y
 Min.   :0.000   Min.   :0.00000
 1st Qu.:0.000   1st Qu.:0.05332
 Median :0.000   Median :0.30312
 Mean   :0.271   Mean   :0.29550
 3rd Qu.:1.000   3rd Qu.:0.50466
 Max.   :1.000   Max.   :0.77292

> # can get the dimensions
> dim(ObsData)

[1] 1000    6

> n<- nrow(ObsData)
```

4 Implement TMLE for the G-computation estimand

1. **Load the `SuperLearner` package. Then specify the Super Learner library with the following algorithms: `SL.glm`, `SL.step` and `SL.gam`.** In practice, we would want to use a larger library with a mixture of simple (e.g. parametric) and more aggressive libraries.

```
> library("SuperLearner")
> # specify the library
> SL.library<- c("SL.glm", "SL.step", "SL.gam")
```

2. **Use Super Learner to estimate $\mathbb{E}_0(Y|A,W) = \bar{Q}_0(A,W)$, which is the expected injury severity given the exposure (prior experience) and baseline covariates.**

   (a) Create dataframe `X` consisting of the covariates $(W1, W2, W3, W4)$ and the exposure A. Also create dataframe `X1` where A has been set to 1, and create dataframe `X0` where A has been set to 0.

   (b) Estimate $\bar{Q}_0(A,W)$ by running `SuperLearner`. Call this object `QbarSL`. Be sure to specify the `SL.library` and the appropriate `family`.

```
> QbarSL<- SuperLearner(Y=ObsData$Y, X=X, SL.library=SL.library, family="binomial")
```

   (c) Use the `predict` function to obtain initial estimates of the expected outcome, given the observed exposure and covariates $\hat{\bar{Q}}(A,W)$.

```
> QbarAW <- predict(QbarSL, newdata=ObsData)$pred
```

   The argument `newdata=ObsData` specifies that we want to predict the outcome using as input the observed exposure and covariates.

   (d) Also obtain the initial estimates of the expected outcome for all units under the exposure $\hat{\bar{Q}}(1,W)$. Now we specify `newdata=X1` to predict the outcome using as input `X1`, where $A = 1$ for all units.

```
> Qbar1W<- predict(QbarSL, newdata=X1)$pred
```

   (e) Finally, obtain the initial estimates of the expected outcome for all units under no exposure $\hat{\bar{Q}}(0,W)$. Now we specify `newdata=X0` to predict the outcome using as input `X0`, where $A = 0$ for all units.

```
> Qbar0W<- predict(QbarSL, newdata=X0)$pred
```

   (f) Evaluate the simple substitution estimator by plugging the estimates $\hat{\bar{Q}}(1,W)$ and $\hat{\bar{Q}}(0,W)$ into the target parameter mapping:

$$\hat{\Psi}_{SS}(\hat{\mathbb{P}}) = \frac{1}{n}\sum_{i=1}^{n} \hat{\bar{Q}}(1,W_i) - \hat{\bar{Q}}(0,W_i)$$

   Note: This step is not part of the TMLE algorithm, but done for comparison.

3. **Estimate the exposure mechanism $g_0(1|W) = \mathbb{P}_0(A=1|W)$, which is the conditional probability of having Dinosaur experience, given baseline covariates.**

   (a) Estimate $g_0(A|W)$ by running `SuperLearner`. Call this object `gHatSL`. Since we are estimating the exposure mechanism, specify the-outcome-for-prediction with `Y=ObsData$A` and the predictors as the baseline covariates with `X=subset(ObsData, select= -c(A,Y))`. Use the same library.

```
> gHatSL<- SuperLearner(Y=ObsData$A, X=subset(ObsData, select= -c(A,Y)),
+                            SL.library=SL.library, family="binomial")
```

   (b) The predicted probability of being Dinosaur experienced, given the subject's baseline characteristics $\hat{g}(A=1|W)$, can be accessed with `gHatSL$SL.predict`

      i. Assign the predicted probability of being experienced $\hat{g}(A=1|W)$ to `gHat1W`:

```
> gHat1W<- gHatSL$SL.predict
```

      ii. Assign the predicted probability of not being experienced $\hat{g}(A=0|W)$ to `gHat0W`.

      iii. Look at the distribution of estimated probabilities: $\hat{g}(1|W)$ and $\hat{g}(0|W)$.

iv. Create empty vector `gHatAW`. Among subjects with $A = 1$, assign the predicted probabilities $\hat{g}(1|W)$. Among subjects with $A = 0$, assign the predicted probabilities $\hat{g}(0|W)$.

4. **Use these estimates to create the clever covariate:**

$$\hat{H}(A,W) = \left(\frac{\mathbb{I}(A=1)}{\hat{g}(1|W)} - \frac{\mathbb{I}(A=0)}{\hat{g}(0|W)}\right)$$

(a) Calculate `H.AW` for each subject:

```
> H.AW<- as.numeric(ObsData$A==1)/gHat1W - as.numeric(ObsData$A==0)/gHat0W
```

For subjects with $A = 1$, the clever covariate is 1 over the predicted probability of being experienced, given the baseline covariates. Among subjects with $A = 0$, the clever covariate is -1 over the predicted probability of not being experienced, given the baseline covariates $\hat{g}(A = 0|W)$.

(b) Also evaluate the clever covariate at $A = 1$ and $A = 0$ for all subjects. Call the resulting vectors `H.1W` and `H.0W`, respectively.

(c) Evaluate the IPTW estimator by taking the empirical mean of the weighted observations:

$$\begin{aligned}\hat{\Psi}_{IPTW}(\hat{\mathbb{P}}) &= \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbb{I}(A_i=1)}{\hat{g}(1|W_i)}Y_i - \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbb{I}(A_i=0)}{\hat{g}(0|W_i)}Y_i \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\mathbb{I}(A_i=1)}{\hat{g}(1|W_i)} - \frac{\mathbb{I}(A_i=0)}{\hat{g}(0|W_i)}\right]Y_i \\ &= \frac{1}{n}\sum_{i=1}^{n}\hat{H}(A_i,W_i)\times Y_i\end{aligned}$$

As before, this is not part of the TMLE algorithm, but implemented for comparison.

5. **Target the initial estimator of the conditional mean outcome $\hat{\bar{Q}}(A,W)$ with information in the estimated exposure mechanism $\hat{g}(1|W)$.**

(a) Run a univariate regression of the outcome Y on the clever covariate $\hat{H}(A,W)$ with the (logit of the) initial estimator as offset. Specifically, we estimate the coefficient ϵ by fitting the following logistic regression model

$$logit\big[\hat{\bar{Q}}^*(A,W)\big] = logit\big[\hat{\bar{Q}}(A,W)\big] + \epsilon\hat{H}(A,W).$$

Note there is no intercept (i.e. there is no β_0 term), and the coefficient on the (*logit* of the) initial estimator is set to 1.

```
> logitUpdate<- glm(ObsData$Y ~ -1 +offset(qlogis(QbarAW)) + H.AW,
+                    family='binomial')
```

- We are again calling the `glm` function to fit a generalized linear model.
- On the left hand side of the formula, we have the outcome Y.
- On the right hand side of the formula, we suppress the intercept by including -1 and use as `offset` the *logit* of our initial Super Learner estimates `QbarAW`.
- In `R`, $logit(x) = log(x/(1-x))$ function is given by `qlogis(x)`.
- The only main term in the regression is the clever covariate $\hat{H}(A,W)$.
- Including `family='binomial'` runs logistic regression.
- Again ignore any warning message.

```
> # we can examine the output by typing
> summary(logitUpdate)
```

(b) Let `epsilon` denote the resulting maximum likelihood estimate of the coefficient on the clever covariate `H.AW`.

```
> epsilon<- logitUpdate$coef
> epsilon
```

(c) Update the initial estimate of $\hat{\bar{Q}}(A,W)$ according to the fluctuation model:

$$\begin{aligned} logit\big[\hat{\bar{Q}}^*(A,W)\big] &= logit\big[\hat{\bar{Q}}(A,W)\big] + \hat{\epsilon}\hat{H}(A,W) \\ \hat{\bar{Q}}^*(A,W) &= logit^{-1}\Big[logit\big[\hat{\bar{Q}}(A,W)\big] + \hat{\epsilon}\hat{H}(A,W)\Big] \end{aligned}$$

```
> QbarAW.star<- plogis(qlogis(QbarAW)+ epsilon*H.AW)
```

In `R`, the inverse-*logit* function is given by `plogis(x)`.

(d) Plug-in the estimated coefficient $\hat{\epsilon}$ to yield targeted estimates of the expected outcome under the exposure $\hat{\bar{Q}}^*(1,W)$ and under no exposure $\hat{\bar{Q}}^*(0,W)$:

$$\begin{aligned} \hat{\bar{Q}}^*(1,W) &= logit^{-1}\Big[logit\big[\hat{\bar{Q}}(1,W)\big] + \hat{\epsilon}\hat{H}(1,W)\Big] \\ \hat{\bar{Q}}^*(0,W) &= logit^{-1}\Big[logit\big[\hat{\bar{Q}}(0,W)\big] + \hat{\epsilon}\hat{H}(0,W)\Big] \end{aligned}$$

Recall $\hat{H}(1,W)$ is the clever covariate evaluated for all units under the exposure, and $\hat{H}(0,W)$ is the clever covariate evaluated for all units under no exposure.

```
> Qbar1W.star <- plogis( qlogis(Qbar1W.star)+ epsilon*H.1W)
> Qbar0W.star <- plogis( qlogis(Qbar0W.star)+ epsilon*H.0W)
```

(e) *Optional*: Try updating again. What is updated $\hat{\epsilon}$?

6. **Step 6. Estimate the statistical parameter by substituting the targeted predictions into the G-Computation formula.**

Estimate $\Psi(\mathbb{P}_0)$ by averaging the difference in the targeted predictions:

$$\Psi_{TMLE}(\hat{\mathbb{P}}) = \frac{1}{n}\sum_{i=1}^{n}\Big[\hat{\mathbb{E}}^*(Y_i|A_i=1,W_i) - \hat{\mathbb{E}}^*(Y_i|A_i=0,W_i)\Big]$$

```
> PsiHat.TMLE <- mean(Qbar1W.star- Qbar0W.star)
> PsiHat.TMLE
```

Solution:

```
> #------------------------------------------
> # 1. Load the Super Learner package and specify the library
> #------------------------------------------
> library("SuperLearner")
> # specify the library
> SL.library<- c("SL.glm", "SL.step", "SL.gam")


> #------------------------------------------
> # 2. Estimate Qbar_0(A,W) with Super Learner
> #------------------------------------------
> # dataframe X with baseline covariates and exposure
> X<-subset(ObsData, select=c(A, W1, W2, W3, W4))
> # set the exposure=1 in X1 and the exposure=0 in X0
> X1 <- X0<-X
> X1$A <- 1         # under exposure
> X0$A <- 0        # under control
> # call Super Learner
> QbarSL<- SuperLearner(Y=ObsData$Y, X=X, SL.library=SL.library, family="binomial")
> QbarSL

Call:
SuperLearner(Y = ObsData$Y, X = X, family = "binomial", SL.library = SL.library)



                  Risk Coef
SL.glm_All  0.04650504    0
SL.step_All 0.04669083    0
SL.gam_All  0.01380438    1

> # get the expected injury severity, given the observed exposure and covariates
> QbarAW <- predict(QbarSL, newdata=ObsData)$pred
> # expected injury severity, given A=1 and covariates
> Qbar1W<- predict(QbarSL, newdata=X1)$pred
> # expected injury severity, given A=0 and covariates
> Qbar0W<- predict(QbarSL, newdata=X0)$pred

> # the fitted value at the observed exposure should equal the fitted value
> # under when A=a
> tail(data.frame(A=ObsData$A, QbarAW, Qbar1W, Qbar0W))

     A    QbarAW    Qbar1W    Qbar0W
995  0 0.1518816 0.1025812 0.1518816
996  1 0.4317119 0.4317119 0.5434104
997  0 0.1703507 0.1158743 0.1703507
998  0 0.3545315 0.2595846 0.3545315
999  0 0.1177309 0.0784898 0.1177309
1000 0 0.4483481 0.3415719 0.4483481

> # note the simple substitution estimator would be
> PsiHat.SS<-mean(Qbar1W - Qbar0W)
> PsiHat.SS

[1] -0.07844085

> #-----------------------------------------
> # 3.  Estimate g_0(A|W) with Super Learner
> #-----------------------------------------

> # call Super Learner for the exposure mechanism
> gHatSL<- SuperLearner(Y=ObsData$A, X=subset(ObsData, select= -c(A,Y)),
+                        SL.library=SL.library, family="binomial")
> gHatSL

Call:
SuperLearner(Y = ObsData$A, X = subset(ObsData, select = -c(A, Y)), family = "binomial",
    SL.library = SL.library)


                 Risk Coef
SL.glm_All  0.1986571    0
SL.step_All 0.1981094    0
SL.gam_All  0.1864084    1


> # generate the predicted prob of being experienced, given baseline cov
> gHat1W<- gHatSL$SL.predict
> # generate the predicted prob of not being experienced, given baseline cov
> gHat0W<- 1- gHat1W

> # summary of propensity scores
> summary(data.frame(gHat1W, gHat0W))


     gHat1W           gHat0W
 Min.   :0.1882   Min.   :0.1209
 1st Qu.:0.2120   1st Qu.:0.7031
 Median :0.2347   Median :0.7653
 Mean   :0.2710   Mean   :0.7290
 3rd Qu.:0.2969   3rd Qu.:0.7880
 Max.   :0.8791   Max.   :0.8118


> # generate the predicted prob of the obs experience, given baseline cov
> gHatAW<- rep(NA, n)
> gHatAW[ObsData$A==1]<- gHat1W[ObsData$A==1]
> gHatAW[ObsData$A==0]<- gHat0W[ObsData$A==0]


> # check that the pred prob of the obs exposure equals the pred prob
> # when A=a
> tail(data.frame(ObsData$A, gHatAW, gHat1W, gHat0W))


     ObsData.A    gHatAW    gHat1W    gHat0W
995          0 0.7313503 0.2686497 0.7313503
996          1 0.1988351 0.1988351 0.8011649
997          0 0.6928286 0.3071714 0.6928286
998          0 0.7587797 0.2412203 0.7587797
999          0 0.6946935 0.3053065 0.6946935
1000         0 0.8051628 0.1948372 0.8051628


> #-----------------------------------------------
> # 4. Create the clever covariate H(A,W) for each subject
> #-----------------------------------------------
> H.AW<- as.numeric(ObsData$A==1)/gHat1W - as.numeric(ObsData$A==0)/gHat0W
> # equiv: H.AW<- (2*ObsData$A-1)/ gHatAW
>
> # also want to evaluate the clever covariates at A=1 and A=0 for all subjects
> H.1W<- 1/gHat1W
> H.0W<- -1/gHat0W
> tail(data.frame(ObsData$A, H.AW, H.1W, H.0W))


     ObsData.A      H.AW     H.1W      H.0W
995          0 -1.367334 3.722319 -1.367334
996          1  5.029293 5.029293 -1.248183
997          0 -1.443358 3.255511 -1.443358
998          0 -1.317906 4.145589 -1.317906
999          0 -1.439484 3.275397 -1.439484
1000         0 -1.241985 5.132490 -1.241985


> #IPTW estimator of the G-computation formula:
> PsiHat.IPTW <-mean( H.AW*ObsData$Y)
> PsiHat.IPTW


[1] -0.08494535


> # equiv
> mean(as.numeric(ObsData$A==1)/gHat1W*ObsData$Y) -
+   mean(as.numeric(ObsData$A==0)/gHat0W*ObsData$Y)


[1] -0.08494535


> #------------------------------------------
> # 5. Update the initial estimator of Qbar_0(A,W)
> # run logistic regression of Y on H.AW using the logit of the esimates as offset
> #------------------------------------------
> logitUpdate<- glm(ObsData$Y ~ -1 +offset(qlogis(QbarAW)) + H.AW, family='binomial')
> epsilon <- logitUpdate$coef
> epsilon


      H.AW
0.01254328


> # obtain the targeted estimates
> QbarAW.star<- plogis(qlogis(QbarAW)+ epsilon*H.AW)
> Qbar1W.star<- plogis(qlogis(Qbar1W)+ epsilon*H.1W)
> Qbar0W.star<- plogis(qlogis(Qbar0W)+ epsilon*H.0W)


> # since the clever cov is not changing, updating will not have any effect
> coef(glm(ObsData$Y ~ -1 +offset(qlogis(QbarAW.star)) + H.AW, family=binomial))


        H.AW
2.738074e-17


> # 6. Estimate Psi(P_0) as the empirical mean of the difference in the targeted
> # outcomes under A=1 and A=0
> PsiHat.TMLE<- mean(Qbar1W.star - Qbar0W.star)
> # comparing the estimates...
> c(PsiHat.SS, PsiHat.IPTW, PsiHat.TMLE)

[1] -0.07844085 -0.08494535 -0.06638246
```

The point estimate from the simple substitution estimator, using Super Learner for $\bar{Q}_0(A, W)$, was -7.8%. The point estimate from IPTW, using Super Learner for $g_0(A|W)$, was -8.5%. The point estimate from TMLE was -6.6%. The true value of the statistical estimand was -6.2%. To evaluate the performance of these estimators (e.g. bias and variance), we would draw another independent sample of size n, implement the 3 estimators (with the same Super Learner library), and repeat 500 or so times.

5 The basics of the `ltmle` package

The `ltmle` package expands the previous `tmle` package. The `ltmle` package estimates parameters corresponding to point-treatment exposures, longitudinal exposures, marginal structural working models, dynamic treatment regimes, and much more!

1. **Load the `SuperLearner` and `ltmle` packages.**

```
> library('SuperLearner')
> library('ltmle')
> # we can learn a lot more about the function by reading the help file
> ?ltmle
```

   - The basic input to the function is the dataset `data`, the exposure variable(s) `Anodes`, the outcome(s) `Ynodes`, and the exposure levels of interest `abar`.
   - The user can also specify censoring variables `Cnodes`, time-dependent covariates `Lnodes`, weights `observation.weights`, and the independent unit `id`. (See the help file for more information.)
   - Initial estimates of the conditional mean outcome $\bar{Q}_0(A, W)$ can be estimated according to a user-specified regression formula (`Qform`) or estimated with Super Learner (`SL.library`).
   - Initial estimates of the propensity score $g_0(A = 1|W)$ can be estimated according to a user-specified regression formula (`gform`) or estimated with Super Learner (`SL.library`).

2. **Call the `ltmle` function using Super Learner to estimate the conditional mean outcome $\bar{Q}_0(A, W)$ and the exposure mechanism $g_0(A = 1|W)$. Use the `summary` function to obtain point estimates and get inference.**

```
> ltmle.SL<- ltmle(data=ObsData, Anodes='A', Ynodes='Y', abar=list(1,0),
+                        SL.library=SL.library)
> summary(ltmle.SL)
```

   Here, `abar=list(1,0)` specifies the comparison of interest: all exposed ($A = 1$) vs. all unexposed ($A = 0$).

3. **Use the `ltmle` package to explore performance under model mis-specification**

   (a) Using main terms parametric regression

```
> ltmle.parametric<- ltmle(data=ObsData, Anodes='A', Ynodes='Y', abar=list(1,0),
+                        Qform=c(Y="Q.kplus1 ~ A+W1+W2+W3+W4"), gform="A~W1+W2+W3+W4")
> summary(ltmle.parametric)
```

   (b) Using unadjusted estimators

```
> # adding a dummy variable to observed data
> ObsData<- data.frame(U=1, ObsData)
> ltmle.unadj <- ltmle(data=ObsData, Anodes='A', Ynodes='Y', abar=list(1,0),
+                      Qform=c(Y="Q.kplus1 ~ A"), gform="A~U")
> summary(ltmle.unadj)
```

4. **Use the `ltmle` package to explore double robustness.**

Solution:

```
> #-------------------------------------------
> # 0. Re-loading the observed data for the workshop & resetting the seed
> #-------------------------------------------
> ObsData<- read.csv("RLab5.TMLE.csv")
> set.seed(252)


> #-------------------------------------------
> # 1. Load the Super Learner & ltmle packages
> #-------------------------------------------
> library("SuperLearner")
> library("ltmle")


> ?ltmle


> #-------------------------------------------
> # 2. call ltmle with Super Learner (same libraries)
> #-------------------------------------------
> ltmle.SL<- ltmle(data=ObsData, Anodes='A', Ynodes='Y', abar=list(1,0),
+                      SL.library=SL.library)
> summary(ltmle.SL)


Estimator:  tmle
Call:
ltmle(data = ObsData, Anodes = "A", Ynodes = "Y", abar = list(1,
    0), SL.library = SL.library)

Treatment Estimate:
   Parameter Estimate:  0.24379
    Estimated Std Err:  0.0087318
              p-value:  <2e-16
    95% Conf Interval: (0.22668, 0.2609)

Control Estimate:
   Parameter Estimate:  0.31021
    Estimated Std Err:  0.007714
              p-value:  <2e-16
    95% Conf Interval: (0.29509, 0.32533)

Additive Treatment Effect:
   Parameter Estimate:  -0.06642
    Estimated Std Err:  0.0080769
```

```
             p-value:  <2e-16
    95% Conf Interval: (-0.08225, -0.050589)
```

The point estimates from `ltmle` package might differ from our code for several reasons. First, the `ltmle` package uses a two-dimensional clever covariate in updating step. This allows us to obtain estimates and inference for - under the identifiability assumptions - the expected outcome under the exposure $\mathbb{E}_{U,X}(Y_1)$, expected outcome under the control $\mathbb{E}_{U,X}(Y_0)$, average treatment effect $\mathbb{E}_{U,X}(Y_1 - Y_0)$. If the outcome is binary, the package will also return estimates of the risk ratio and odds ratio. (See Example1 in the help file.) In our code, we used a one-dimensional clever covariate for simplicity and to focus on the G-computation formula (equal to the average treatment effect). Second, the `ltmle` package incorporate the clever covariates in the weights (as opposed to covariates in the fluctuation model.) Third, the `ltmle` package bounds the estimated propensity scores. This bounding is included to deal with theoretical and practical positivity violations. Finally, the Super Learner algorithm could split the data into different folds. (This is why we reset the seed.)

```
> #-------------------------------------------
> # 3a. call ltmle with main terms parametric regression for both QbarAW & g(A|W)
> #-------------------------------------------
> ltmle.parametric<- ltmle(data=ObsData, Anodes='A', Ynodes='Y', abar=list(1,0),
+                        Qform=c(Y="Q.kplus1 ~ A+W1+W2+W3+W4"), gform="A~W1+W2+W3+W4")
> summary(ltmle.parametric)

Estimator:  tmle
Call:
ltmle(data = ObsData, Anodes = "A", Ynodes = "Y", Qform = c(Y = "Q.kplus1 ~ A+W1+W2+W3+W4"),
    gform = "A~W1+W2+W3+W4", abar = list(1, 0))

Treatment Estimate:
   Parameter Estimate:  0.18893
    Estimated Std Err:  0.012821
              p-value:  <2e-16
    95% Conf Interval: (0.1638, 0.21406)

Control Estimate:
   Parameter Estimate:  0.33387
    Estimated Std Err:  0.0083281
              p-value:  <2e-16
    95% Conf Interval: (0.31755, 0.35019)

Additive Treatment Effect:
   Parameter Estimate:  -0.14494
    Estimated Std Err:  0.015209
              p-value:  <2e-16
    95% Conf Interval: (-0.17475, -0.11513)

> #-------------------------------------------
> # 3b. call ltmle with unadjusted
> # adding a dummy variable to observed data
> #-------------------------------------------
> ObsData<- data.frame(U=1, ObsData)
> ltmle.unadj <- ltmle(data=ObsData, Anodes='A', Ynodes='Y', abar=list(1,0),
+                        Qform=c(Y="Q.kplus1 ~ A"), gform="A~U")
> summary(ltmle.unadj)
```

```
Estimator:  tmle
Call:
ltmle(data = ObsData, Anodes = "A", Ynodes = "Y", Qform = c(Y = "Q.kplus1 ~ A"),
    gform = "A~U", abar = list(1, 0))

Treatment Estimate:
   Parameter Estimate:  0.18777
    Estimated Std Err:  0.012423
              p-value:  <2e-16
    95% Conf Interval: (0.16342, 0.21212)

Control Estimate:
   Parameter Estimate:  0.33554
    Estimated Std Err:  0.0083973
              p-value:  <2e-16
    95% Conf Interval: (0.31909, 0.352)

Additive Treatment Effect:
   Parameter Estimate:  -0.14777
    Estimated Std Err:  0.014995
              p-value:  <2e-16
    95% Conf Interval: (-0.17716, -0.11839)


> #------------------------------------------
> # 4a. explore double robustness using misspecified regression for QbarAW
> #------------------------------------------
> ltmle.DR<- ltmle(data=ObsData, Anodes='A', Ynodes='Y', abar=list(1,0),
+                      SL.library=SL.library,
+                    Qform=c(Y="Q.kplus1 ~ A"))
> summary(ltmle.DR)


Estimator:  tmle
Call:
ltmle(data = ObsData, Anodes = "A", Ynodes = "Y", Qform = c(Y = "Q.kplus1 ~ A"),
    abar = list(1, 0), SL.library = SL.library)

Treatment Estimate:
   Parameter Estimate:  0.23599
    Estimated Std Err:  0.012911
              p-value:  <2e-16
    95% Conf Interval: (0.21068, 0.26129)

Control Estimate:
   Parameter Estimate:  0.31763
    Estimated Std Err:  0.0087212
              p-value:  <2e-16
    95% Conf Interval: (0.30054, 0.33473)

Additive Treatment Effect:
   Parameter Estimate:  -0.081647
    Estimated Std Err:  0.015581
              p-value:  1.6028e-07
    95% Conf Interval: (-0.11218, -0.05111)
```

```
> #------------------------------------------
> # 4b. explore double robustness using misspecified regression for gAW
> #------------------------------------------
> ltmle.DRb<- ltmle(data=ObsData, Anodes='A', Ynodes='Y', abar=list(1,0),
+                  SL.library=SL.library,
+                  gform="A~U")
> summary(ltmle.DRb)


Estimator:  tmle
Call:
ltmle(data = ObsData, Anodes = "A", Ynodes = "Y", gform = "A~U",
    abar = list(1, 0), SL.library = SL.library)

Treatment Estimate:
   Parameter Estimate:  0.23462
    Estimated Std Err:  0.0080797
              p-value:  <2e-16
    95% Conf Interval: (0.21879, 0.25046)

Control Estimate:
   Parameter Estimate:  0.31307
    Estimated Std Err:  0.0077779
              p-value:  <2e-16
    95% Conf Interval: (0.29783, 0.32831)

Additive Treatment Effect:
   Parameter Estimate:  -0.078446
    Estimated Std Err:  0.0075134
              p-value:  <2e-16
    95% Conf Interval: (-0.093172, -0.06372)
```

Formally, an estimator is *consistent* if the point estimates converge (in probability) to the estimand as sample size $n \to \infty$. This is an asymptotic property. Here, we only have one sample of size $n = 1,000$. To evaluate the consistency of TMLE, we would need to do multiple runs at increasing samples sizes, e.g. $n = 500$, $n = 5,000$, $n = 50,000$, $n = 500,000$.

Solution:

Appendix: A specific data generating process

The following code was used to generate the data set `RLab5.TMLE.csv`. In this data generating process (one of many compatible with the SCM $\mathcal{M}^{\mathcal{F}}$), *all exogenous errors are independent.*

```
> library('MASS')
> #-------
> # generateData - function to generate the data
> # input: number of draws, whether or not there is a treatment effect
> # output: observed data + counterfactuals
> generateData<- function(n, effect=T){
+
+   W1 <- rbinom(n, size=1, prob=0.5)
+   W2<- runif(n, min=0, max=1)
+
+   # W3 and W4 are drawn from a multivariate normal (i.e. correlated)
+   s=1
+   Sigma<- matrix(0.85*s*s, nrow=2, ncol=2)
+   diag(Sigma)<- s^2
+   Z<- mvrnorm(n, rep(0,2), Sigma)
+   W3<- Z[,1]; W4<- Z[,2];
+
+   # generate the propensity score P(A=1|W)
+   pscore<- plogis(-1.25 - .25*(W1+W2) +.5*W3*W4)
+
+   A<- rbinom(n, size=1, prob= pscore)
+
+   U.Y<- rnorm(n, 0, s)
+   # generate the counterfactual outcome with A=0
+   Y.0<- generateY(W1=W1, W2=W2, W3=W3, W4=W4, A=0, U.Y=U.Y)
+
+   if(!effect){ # if there is no effect, the counterfactual under txt =
+     # the counterfactual under the control
+     Y.1<- Y.0
+   }else{ # otherwise, generated the counterfactual outcome with A=1
+     Y.1<- generateY(W1=W1, W2=W2, W3=W3, W4=W4, A=1, U.Y=U.Y)
+   }
+
+   # assign the observed outcome based on the observed exposure
+   Y<- rep(NA, n)
+   Y[A==1]<- Y.1[A==1]
+   Y[A==0]<- Y.0[A==0]
+
+   data<- data.frame(W1, W2, W3, W4, A, Y, Y.1, Y.0)
+   data
+ }
> #---------
> # generateY: function to generate the outcome given the
> #   baseline covariates, exposure and background error U.Y
> #------------------
> generateY<- function(W1, W2, W3, W4, A, U.Y){
+   W1*plogis(0.25 +.5*W2 -1*W4 -0.5*A  -2*W4*W4 -.5*W4*A + .25*U.Y) +
+     (1-W1)*plogis(.25  -.5*W2 -1*W3 -0.5*A  -2*W3*W3 -.5*W3*A - .25*U.Y)
+ }


> #------------
> # Creation of the dataset
> #--------------
> set.seed(1)
> FullData<- generateData(n=1000, effect= T)
> # remove unobservable counterfactuals
> ObsData<- subset(FullData, select=c(W1,W2,W3,W4, A, Y) )
> write.csv(ObsData, file="RLab5.TMLE.csv", row.names=F)
> #------------
```

We could obtain the true value of the causal parameter $\Psi^{\mathcal{F}}(\mathbb{P}_{U,X})$ by drawing a huge number of observations and taking the difference in the means of the counterfactual outcomes.

```
> set.seed(252)
> TrueData<- generateData(n=100000)
> # Simply take the difference in mean the counterfactuals
> Psi.F<- mean(TrueData$Y.1 - TrueData$Y.0)
> Psi.F
```

```
[1] -0.06235098
```

The average treatment effect $\Psi^{\mathcal{F}}(\mathbb{P}_{U,X})$ is -6.2%. The expected injury severity would be 6.2% lower if all subjects had prior Dinosaur experience than if none were experienced.