Simulation.rewrite.rst

1 Scenario 1

- W_1, W_2 and W_3 are all continuous random variables
- no interaction term in $Y \sim A + W$, i.e., $\tau(W)$ is a constant

$$W_1 \sim Unif(-1, 1)$$
$$W_2 \sim Unif(-1, 1)$$
$$W_3 \sim Unif(-1, 1)$$
$$A \sim Bernoulli(\pi_0) \ where \ \pi_0 = expit(0.5 + \frac{1}{3}W_1)$$
$$Y \sim N(\mu_0, 1)$$
$$\mu_0(A, W) = 0.1 + 0.2 * A + 0.5 * W_1 - 0.3 * W_2 + 0.1 * W_3$$
$$\mu_0(1, W) = 0.3 + 0.5 * W_1 - 0.3 * W_2 + 0.1 * W_3$$
$$\mu_0(0, W) = 0.1 + 0.5 * W_1 - 0.3 * W_2 + 0.1 * W_3$$
$$\tau(W) = 0.2$$
$$\psi_0 = 0.04$$
$$\theta_0 = 0$$

1.1 Result

- Simulation setting:
  - $n = 2000$
  - repeat 1000 times

Table 1: $\sqrt{n} * \frac{1}{1000} \sum(\psi_n - \psi_0)$ when $n = 2000$

est	SuperLearner	glm.all
plug.in1	0.057	0.0829
os.est1	0.131	0.0849
plug.in2	0.461	0.3585
os.est2	0.747	0.3583

Explanation:

- Estimator
  - estimator labeled with 1 means estimating $\hat{\mu}_1$ and $\hat{\mu}_0$ all together in one model
  - estimator labeled with 2 means estimating $\hat{\mu}_1$ and $\hat{\mu}_0$ in separate models
- Model
  - SuperLearner means: using `SuperLearner()` for both propensity score and outcome regression
    - SL.library = c("SL.mean", "SL.glm", "SL.gam", "SL.earth")
  - glm.all means: using `glm()` for both propensity score and outcome regression

- Bias is calculated by $\sqrt{2000} * \frac{1}{1000} \sum(\psi_n - \psi_0)$ and the code is

```
sqrt(n)*mean(rst$psi.plug.in - psi0)
sqrt(n)*mean(rst$psi.one.step.est - psi0)
```

- Findings (more details will be given in the following section 1.2 Detail)
  - the plug-in estimator converge to 0 faster than the one-step estimator
  - estimating $\hat{\mu}_1$ and $\hat{\mu}_0$ in separate models always includes more bias
  - the one-step estimator using `SuperLearner()` failed to converge to 0 even when $n = 2000$

Table 2: estimated standard error and empirical standard deviation of the estimator

est	SuperLearner	glm.all
plug.in1	0.8165 (0.8175)	0.8269 (0.8238)
os.est1	0.8165 (0.8371)	0.8269 (0.8228)
plug.in2	0.9076 (0.8564)	0.8906 (0.8449)
os.est2	0.9076 (0.9482)	0.8906 (0.8454)

Explanation:

- Standard error
  - the value outside the parentheses is the mean of the estimate of the standard error, which is essentially `mean(se(eif.hat))`
  - the value inside the parentheses is the empirical standard deviation of the estimator

```
mean(rst$psi.se) # 0.8269
sd(sqrt(n)*(rst$psi.plug.in - psi0)) # 0.8238
```

- Findings (more details will be given in the following section 1.2 Detail)
  - When the glm model is correct, the estimated standard error is close to the empirical standard deviation
  - estimating $\hat{\mu}_1$ and $\hat{\mu}_0$ in separate models always increases the value of estimated standard error
  - the one-step estimator using `SuperLearner()` underestimates the empirical standard deviation

1.2 Detail

Why does the plug-in estimator converge to 0 faster than the one-step estimator?
Our guess is that the model used for estimating the outcome regression is correctly specified. So `tau.hat` is asymptotically unbiased and `psi.plug.in <- mean(tau.hat^2)` is also asymptotically unbiased.

```
mu.reg <- glm(Y ~ ., data=AW, family='gaussian')
```

$$\hat{\mu}(A, W) = 0.0836 + 0.2289 * A + 0.4083 * W_1 - 0.3607 * W_2 + 0.1312 * W_3$$

This is actually very close to the correct model.

$$\mu_0(A, W) = 0.1 + 0.2 * A + 0.5 * W_1 - 0.3 * W_2 + 0.1 * W_3$$

What phenomenon will support our guess?
If there exists an interaction term $A * W$ in data generating process, then a simple glm model is not a correct model. The one-step estimator should converge to 0 faster than the plug-in estimator. See Section 2.2.

Why estimating $\hat{\mu}_1$ and $\hat{\mu}_0$ in separate models always include more bias?
My guess is that when estimating $\hat{\mu}_1$ and $\hat{\mu}_0$ separately, $\hat{\tau}$ will be a function of W. τ in this case is actually a constant. So more bias is included.

```
AW1 <- cbind(data.frame(Y=Y[which(A==1)]), data.frame(W[which(A==1),]))
mu1.reg  <- glm(Y ~ ., data=AW1, family='gaussian')
# Coefficients:
# (Intercept)           W1           W2           W3
#      0.3138       0.3825      -0.3379       0.1827
AW0 <- cbind(data.frame(Y=Y[which(A==0)]), data.frame(W[which(A==0),]))
mu0.reg  <- glm(Y ~ ., data=AW0, family='gaussian')
# Coefficients:
# (Intercept)           W1           W2           W3
#     0.08462      0.45018     -0.39852      0.04603
```

$$\hat{\tau} = 0.2292 + \hat{\gamma}_1 * W_1 + \hat{\gamma}_2 * W_2 + \hat{\gamma}_3 * W_3$$

$E(\hat{\tau}) = 0.2292$ but the variance and covariance of W will be included in $E(\hat{\tau}^2)$.

But this only explain the bias in the plug-in estimator I guess. I still can not figure out why does the one-step estimator fail to converge.

Why did the one-step estimator using `SuperLearner()` failed to converge to 0 even when $n = 2000$?

I am not sure whether 0.131 should be considered as converging to 0 or not. (Add a plot maybe) But from my understanding, the bias from the one-step estimator should be very close to the one from the plug-in estimator like the result from using `glm()` (0.0829 vs 0.0849 in Table 1).

the one-step estimator using `SuperLearner()` underestimates the empirical standard deviation

This is actually also the biggest issue we faced when we were trying to do our previous complicated simulation setting. (underestimating the empirical standard deviation leads to the undercoverage of the Confidence Intervals)

My initial guess is that `SuperLearner()` use a couple of models, hence the variance of the estimator is decreasing especially when the model is weakly correlated. But the contradictory issue is that the empirical standard deviation is also the variance of the estimator, it should be decreasing at the same time. So confusing and still working on this part.

2 Scenario 2

- W_1, W_2 and W_3 are all continuous random variables
- add interaction term in $Y \sim A + W_1 + W_2 + W_3 + AW_1$, i.e., $\tau(W)$ is a function of W_1

$$W_1 \sim Unif(-1, 1)$$
$$W_2 \sim Unif(-1, 1)$$
$$W_3 \sim Unif(-1, 1)$$
$$A \sim Bernoulli(\pi_0) \ where \ \pi_0 = expit(0.5 + \frac{1}{3}W_1)$$
$$Y \sim N(\mu_0, 1)$$
$$\mu_0(A, W) = 0.1 + 0.2 * A + 0.5 * A * W_1 - 0.3 * W_2 + 0.1 * W_3$$
$$\mu_0(1, W) = 0.3 + 0.5 * W_1 - 0.3 * W_2 + 0.1 * W_3$$
$$\mu_0(0, W) = 0.1 - 0.3 * W_2 + 0.1 * W_3$$
$$\tau(W) = 0.2 + 0.5 * W_1$$
$$\psi_0 = 0.2^2 + (0.5^2)/3 = 0.1233$$
$$\theta_0 = (0.5^2)/3 = 0.083$$

2.1 Result

- Simulation setting:
  - $n = 2000$
  - repeat 1000 times

Table 3: $\sqrt{n} * \frac{1}{1000} \sum(\psi_n - \psi_0)$ when $n = 2000$

est	glm	glm.interaction1
plug.in1	-3.8729	
os.est1	-3.6471	
plug.in2	0.4239	
os.est2	0.4199	

Table 4: estimated standard error and empirical standard deviation of the estimator

est	glm
plug.in1	0.78 (0.77)
os.est1	0.78 (0.82)
plug.in2	1.53 (1.47)
os.est2	1.53 (1.47)

2.2 Detail

Support our guess in Section 1.2
This time a simple glm model is not a correct model. The one-step estimator has smaller bias than the plug-in estimator. Also, estimating $\hat{\mu}_1$ and $\hat{\mu}_0$ in separate models can provide a `tau.hat` more close to the true $\tau(W)$.