Authors' Response to Reviewers' Comments: "CIMTx: An R package for causal inference with multiple treatments using observational data" (RJournal 2021-202)

June 23, 2022

Reviewer 1

Hu et al. summarize the CIMTx package, which implements several approaches for causal inference from observational data within the context of multiple treatments. The package and the paper are generally well-written and address an application area of interest to the R *community. Some parts of the package and paper appear underdeveloped, in particular the standard error calculations for multiple approaches.*

Thank you for your positive assessment of our manuscript and for your constructive comments. Below please find our point-by-point response to your comments.

Major comments

1. *The authors do a good job of briefly summarizing the estimation procedures used in CIMTx (with the exception of the VM approach – see minor comment). However, the calculation of standard errors – and thus confidence intervals – is critical to almost all potential users. In the methods section, the authors should briefly describe and justify the standard error calculation for each estimation procedure. For example, the VM functionality of CIMTx appears to use an uncharacterized bootstrapping procedure, and if boot = FALSE is specified, no confidence intervals are reported. This is especially concerning because Lopez and Gutman (2017), who describe VM, refer to the calculation of standard errors as "an open research question," and they "caution against" certain bootstrapping procedures (page 443). For the TMLE method, no confidence intervals are reported. For IPTW, it is not clear to extract effect estimates at all when boot = FALSE (see minor comment).*

   Response: Thank you for your comment. For each method, we have added a brief description of how the variance can be estimated. The following text has been added to **Regression adjustment** section on **Page 5**:

> *"Inferences about treatment effect can be obtained based on the L posterior average treatment effects. The 95% credible interval is calculated using the 2.5th percentile and the 97.5th percentile of the posterior draws (Kruschke, 2014)."*

The following text has been added to **Inverse probability of treatment weighting** on **Page 7**:

> *"For IPTW estimators, variance can be estimated via a robust sandwich-type variance estimator or a bootstrap variance estimator. In practice, a bootstrap variance estimator is often recommended. (Austin, 2016)."*

The following sentence in **paragraph 1 on Page 8** explains how uncertainty intervals can be obtained for the **Bayesian additive regression trees** method.

> *"The posterior inferences about the treatment effects can be drawn in a similar way as described in the Regression adjustment section."*

The following text has been added to **Regression adjustment with multivariate spline of GPS** section on **Page 8**:

> *"Confidence intervals of treatment effect estimates can be obtained using nonparametric bootstrap for RAMS (Hu and Gu, 2021)."*

For VM, we agree with the reviewer that nonparametric bootstrap is not an appropriate way to obtain the confidence intervals, and have removed the bootstrap feature for VM from CIMTx. Now when users set `method = "VM"` and `boot = TRUE`, they will get an informative error message which says "Bootstrap confidence intervals are not appropriate for VM based on Lopez and Gutman (2017). The current version of CIMTx does not support the standard error calculation for VM." The following text has also been added to **Vector matching** section on **Page 8**:

> *"The CIMTx does not provide confidence intervals for treatment effect estimates because the authors of this method, Lopez and Gutman (2017), did not provide an approach to estimate the sampling variance of the VM estimator."*

For the TMLE method, Rose and Normand (2019) suggests obtain confidence intervals using influence curves or bootstrap. We have added a supportive function to implement bootstrap for TMLE to obtain the confidence interval by setting `boot = TRUE` with the following codes:

```
tmle_res_boot <- ce_estimate(y = data$y, x = data$covariates,
                             w = data$w, method = "TMLE",
                             estimand = "ATE", boot = TRUE,
                             nboots = 10, sl_library = c("SL.glm",
                             "SL.glmnet", "SL.rpart"))
```

The effect estimates could be extracted using `summary(tmle_res_boot)`.

```
#> $ATE12
#>       EST   SE LOWER UPPER
#> RD -0.22 0.06 -0.32 -0.14
#> RR  0.51 0.09  0.35  0.62
#> OR  0.37 0.10  0.22  0.51

#> $ATE13
#>       EST   SE LOWER UPPER
#> RD -0.50 0.05 -0.55 -0.41
#> RR  0.31 0.05  0.23  0.39
#> OR  0.11 0.03  0.07  0.18

#> $ATE23
#>       EST   SE LOWER UPPER
#> RD -0.27 0.06 -0.39 -0.21
#> RR  0.62 0.07  0.48  0.68
#> OR  0.32 0.08  0.19  0.41
```

We have added the following text to the section of **Targeted maximum likelihood estimation** on **Page 9**:

> *"As suggested by Rose and Normand (2019), nonparametric boostrap is used in CIMTx to obtain the confidence interval of the treatment effect estimate. "*

For VM, IPTW and RAMS, when `boot = FALSE`, we added new S3 methods for the `print` and `summary` functions so that the output of the `ce_estimate()` function will be more informative and users will know how to extract the effect estimates from the output. For example, the following R codes implement RAMS with GPS estimated by multinomial logistics regression to estimate the ATE effects without bootstrap (`boot = FALSE`).

```
rams_multi_res <- ce_estimate(y = data$y, x = data$covariates,
                              w = data$w, estimand = "ATE",
                              method = "RAMS-Multinomial",
                              boot = FALSE)
```

Typing `rams_multi_res` in the R console will show the following messages.

```
#> This is ATE results from estimation method RAMS-Multinomial using
#> CIMTx. For effect estimates, please use summary function.
```

Then running `summary(rams_multi_res)` will print out the ATE effect estimates in terms of RD, RR and OR:

```
#>    ATE12 ATE13 ATE23
#> RD -0.26 -0.51 -0.26
#> RR  0.48  0.32  0.66
#> OR  0.32  0.10  0.32
```

2. *The sensitivity analysis demonstration should include enough detail so that readers can understand the basic inputs, usage, and interpretation without reading the separate 20 page methods paper (Hu et al, 2021c). Why would sensitivity analysis calculations based on BART estimation be useful to a researcher who didn't use that estimation procedure? How does a researcher use a plot like Figure 5 to make judgements contextualizing their specific findings, and how might they report those judgements in a way that is clear to non-statistical readers of applied papers (e.g. "It would take a confounder of this much strength to explain away the observed treatment effect")? The three options for priors (point mass, re-analysis over a range of point-mass priors, full prior with uncertainty specified) need to be explained – not just demonstrated – including how a researcher would go about choosing those priors.*

Response: Thank you for your comment. First, we have added Table 2 on **Page 11** to interpret the confounding function, or sensitivity parameter, used in the new sensitivity analysis method.

> *"Table 2 demonstrates the plausible assumptions about the confounding functions and their interpretations."*

The benefit of BART like sensitivity analysis approach is that it incorporates the statistical uncertainty due to sampling and the uncertainty about the values of the sensitivity parameters in a formal sensitivity analysis, while providing an accurate estimation of the confounding function adjusted treatment effect estimation. We have also added a brief explanation for the three ways to specify the priors for the confounding functions (point mass, re-analysis over a range of point-mass priors, full prior with uncertainty specified) in **Paragraph 1 on Page 11**:

> *"There are three ways in which we can specify the prior for the confounding functions: (i) point mass prior; (ii) re-analysis over a range of point mass priors (tipping point); (iii) full prior with uncertainty specified. Since the new sensitivity analysis approach was developed within the Bayesian framework, strategy (iii) offers an advantage of incorporating the statistical uncertainty due to sampling and the uncertainty about the values of the sensitivity parameters. In strategy (i), a fixed value is assumed for the sensitivity parameter. Strategy (ii) expands on strategy (i) and examines how the causal conclusion would change when a range of values are assumed for the sensitivity parameter. We will demonstrate all three cases of prior specifications with* `sa()` *function in* **CIMTx** *package."*

Figure 5 on page 15 will be useful to researchers as a sensitivity analysis example. We can examine how the causal conclusions would change if we assumed that the unmeasured confounders guiding clinicians to make treatment decisions would tend to lead them to systematically prescribe treatment $w = 1$ over $w = 3$ to healthier patients. The message to non-statistical readers is that as the magnitude of unmeasured confounding (healthier patients more likely take treatment $w = 1$) increases, the treatment benefit with $w = 3$ will become larger, indicated by the increasing adjusted effect estimates.

Finally, we have added the R implementation of key sensitivity analysis steps on **Page 11-13**.

Minor comments

1. *Table 1 seems more self-advertising than useful to readers. I suggest restricting the comparison to just the R packages which can be used for multiple treatments (since summarizing all causal inference packages is impractical). Use the columns to include information about which estimation procedures are implemented by each package, since this is likely to be of most interest to readers comparing packages.*

   Response: Thank you for your suggestion. We now only included the R packages suitable for multiple treatments and added a column showing the estimation procedure used for each package in Table 1.

2. *CIMTx only supplies methods to identify a common support for 3 of the 6 estimation procedures it implements. If identification of common support is to be emphasized as a feature of CIMTx in the introduction, abstract, and table 1, this caveat should be made explicit.*

   Response: Thank you for pointing this out. We have emphasized that VM, BART and IPTW implemented the common support procedure in our abstract, introduction and Table 1.

   In the abstract, the following text has been added:

   > *"For the positivity assumption, CIMTx demonstrates techniques to identify the common support region for retaining inferential units using inverse probability of treatment weighting, Bayesian additive regression trees and vector matching."*

   In the introduction section, the following text has been added:

   > *"In addition, CIMTx provides strategies to define a common support region to address the positivity assumption using IPTW, BART, VM."*

   In Table 1, we have added the following footnote:

   > *"*: Identification of Common Support is only for VM, BART and IPTW related methods."*

3. *Why is n cluster = 2 in the vector matching example? This means that the clustering step is just dividing the data set in two. Is this is an appropriate choice?*

   The argument `n_cluster` is used in the K-means clustering step of the vector matching method. Lopez and Gutman (Lopez and Gutman 2017) did not provide any guidance on how to choose the number of clusters used in vector matching. We tried different values for `n_cluster` using our simulated data and found that setting `n_cluster` to 3 produced the smallest relative bias; see Figure 1 Thus, we set `n_cluster = 3` in our matching example.

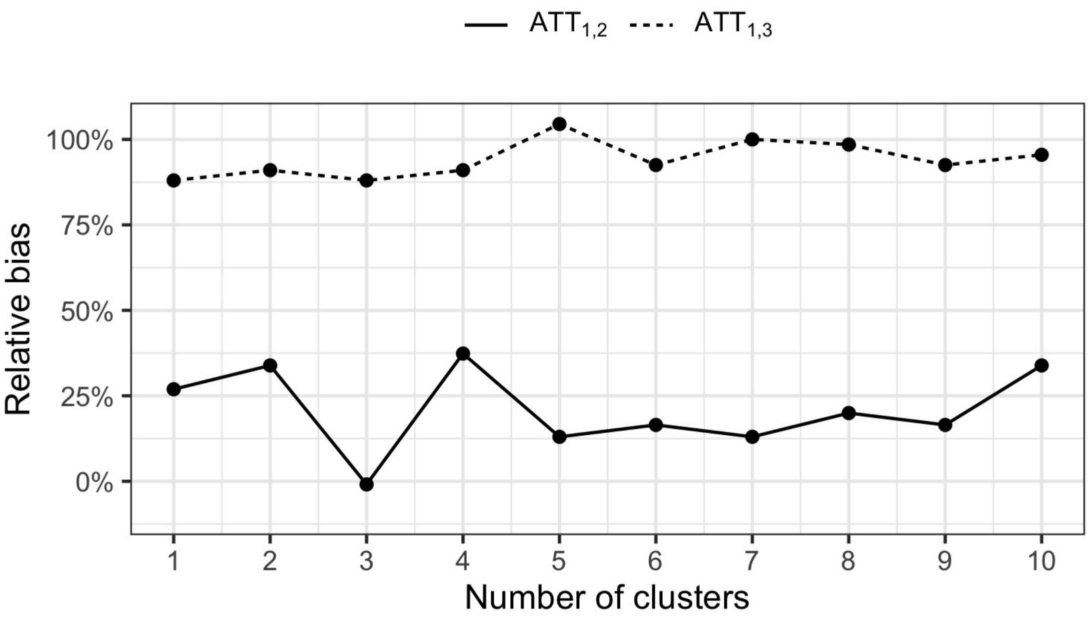

Figure 1: Relative bias by changing the argument `n_cluster` from 1 to 10.

Refs

1 Lopez, MJ, Gutman, R. Estimation of causal effects with multiple treatments: a review and new ideas. *Statistical Science* 2017; 32: 432–454.

4. *The number matched all value in the VM result seems to give the number of subjects matched in each of the bootstrap replicates. This should be stated in the manuscript and documentation.*

   Response: We have now removed the bootstrap feature for VM.

Other Comments: CIMTx Package

1. *For most results from ce estimate, estimates are extracted with summary(), but running summary on an IPTW result with boot = FALSE doesn't do anything (just prints the R default for a list object). How should the user get effect estimates?*

   Response: Thank you for the comment. We have added new S3 methods for the `summary()` function. Now when the argument `boot` is set to `FALSE` in the `ce_estimate()` function, the `summary()` function will show the point estimates of causal estimands in terms of RD, OR and RR. For example, using the simulated data, the following codes implemented ATE estimation using IPTW with weights estimated by multinomial logistic regression without bootstrap (`boot = FALSE`).

   ```
   iptw_multi_res <- ce_estimate(y = data$y, x = data$covariates,
                                 w = data$w, boot = FALSE,
                                 method = "IPTW-Multinomial",
                                 estimand = "ATE")
   ```

   Users can get the effect estimates with `summary(iptw_multi_res)`:

   ```
   #>    ATE12 ATE13 ATE23
   #> RD -0.22 -0.47 -0.25
   #> RR  0.54  0.36  0.66
   #> OR  0.38  0.13  0.34
   ```

2. *It seems like the plot method is only implemented for IPTW results. When I try to plot other objects, I get an uninformative error message. Consider implementing a plot S3 method that throws an error like "Plotting is only supported for IPTW results."*

   Response: We have added new S3 methods for the `plot()` function to implement the feature you suggested. Now when users try to plot objects other than IPTW results, they will get a more informative error message. For example, the following codes implemented ATE estimation using RA.

   ```
   ra_res <- ce_estimate(y = data$y, x = data$covariates, w = data$w,
                         ndpost = 100, method = "RA", estimand = "ATE")
   ```

   When users try to plot the output using `plot(ra_res)`, they will get the following informative error message:

   ```
   #> Plot function is not supported for estimation method using RA.
   #> Plot function is only supported for estimation method using IPTW
   #> when boot is set to FALSE.
   ```

3. *Some of the source code for CIMTx contains long lines of code ($\geq$ 80 characters) and chunks of code that are commented out (e.g. ce_estimate_iptw_ate_boot.R). To make your source code easy to read by other developers, I recommend lint package from the lintr package.*

   Response: We have tidied up the source code for our package. Now running the `lint_package()` function from **lintr** package will not show any messages regarding the long lines of codes or commented out codes in our package.

4. *To help new users get up and running quickly, it can be useful to set defaults for data_sim so that a simple command like data_sim(sample_size = 500) is all it takes to make a data set for them to experiment with.*

   Response: Thank you for the suggestion. We have added sensible default values for the arguments in the `data_sim()` function. For example, the default value for the argument `n_trt` is set to 3. The default value for the argument `delta` is set to `c(0,0)` which corresponds to an approximately equal sample sizes across three treatment groups. The default value for the argument `tau` is set to `c(0, 0, 0)`, which corresponds to an approximately equal outcome event probability across three treatment groups. The default values for other parameters in the `data_sim()` function could be found in the help page by typing `?data_sim`. Now users can generate simple data using `data_sim(sample_size = 300)`.

5. *It would greatly improve usability if the print method for results from ce estimate showed something informative (even just "This is a result from estimation method XXX using CIMTx. For effect estimates, use summary()").*

   Response: We have added new S3 methods for the `print()` function to make the output from `ce_estimate()` more informative. For example, when users run the following codes to estimate ATT effects using BART:

```
bart_res <- ce_estimate(y = data$y, x = data$covariates, w = data$w,
                        method = "BART", estimand = "ATT",
                        ndpost=100, reference_trt = 1)
```

   Then running `bart_res` will show the following message:

```
#> This is ATT results from estimation method BART with confidence
#> interval estimated by Bayesian posterior samples. For effect
#> estimates, please use summary function.
```

6. *The documentation for data sim should explain what tau, delta, and psi are (e.g. "psi should be a real number between a and b. Higer values mean...; lower values mean...")*

   Response: We have added detailed information for the arguments `tau, delta, psi` in the `ce_estimate()` function.

   The argument `psi` is a numeric value for the parameter governing the sparsity of covariate overlap. Higher values mean weaker covariate overlap; lower values mean stronger covariate overlap. The default is set to 1, which corresponds to a moderate covariate overlap.

   The argument `tau` is a numeric vector of length `n_trt` inducing different outcome event probabilities across treatment groups. Higher values mean higher outcome event probability for the treatment group; lower values mean lower outcome event probability for the treatment group. The default is set to `c(0, 0, 0)`, which corresponds to an approximately equal outcome event probability across three treatment groups.

   The argument `delta` is a numeric vector of length `n_trt-1` inducing different ratio of units across treatment groups. Higher values mean higher proportion for the treatment group; lower values mean lower proportion for the treatment group. The default is set to `c(0,0)`, which corresponds to an approximately equal sample sizes across three treatment groups.

Reviewer 2

The authors present a description of the R package CIMTx (Causal Inference with Multiple Treatments), which allows for the estimation of two types of causal effects (average treatment effect, ATE; average treatment effect among the treated, ATT) using a variety of methods. The package also provides facilities to simulate data for a multiple treatment context satisfying user-specified constraints and methods for sensitivity analysis when the causal ignorability assumption is violated. The manuscript can be improved to better highlight the package's contributions, and the code is unfortunately very poorly documented, making it difficult to thoroughly review. The help files for the user-facing functions are generally clear, and the examples are thorough. The package appears to be a useful addition to the causal inference toolkit in the multiple treatment setting and is user-friendly.

Major comments

1. *The manuscript would greatly benefit from reorganization. In its current form, description of the use of the R package does not start until page 6, halfway through the paper. Since the R package is the achievement the authors want to highlight in this paper, it should be front and center! The authors begin with a section describing the methodology the package implements; then, starting on page 6, they introduce R code and the details of the package. It is my opinion that mixing the code and the details of the method would be a more effective approach to describing what CIMTx does and*

how it is use, rather than having separate sections on methodology and the package. My suggested approach would be to introduce the multiple treatments setting, then the data generative model for simulation. This will make the setting more concrete. Then, after the model itself is introduced, describe the data_sim() function. Next, introduce the methods for estimating causal effects (regression adjustment, IPTW, etc.), and include how to use ce_estimate() to obtain estimates using that particular method before moving on to the next. This will make for a more readable manuscript that highlights the contribution more clearly. It will also eliminate the problem of discussing arguments to ce_estimate() in the methodology section before ce_estimate() has actually been introduced.

Response: Thank you for this important comment. As you suggested, we have reorganized the paper by mixing the method and the R codes together.

2. *The authors should describe how to interpret the presented output from ce_estimate(). Most of the output appears self-explanatory, but given that the analysis is being run on simulated data, it should be noted whether the output matches the true effect. Estimates (i.e., summarized output) should be shown for all methods.*

   Response: We have added new S3 methods for the `print` function to make the output from `ce_estimate()` function more informative. Users will also know how to extract the effect estimates from the printed message. For example, using the same simulated data from our manuscript, when users run the following codes to estimate ATE estimation using IPTW with weights estimated by GBM:

```
iptw_gbm_res <- ce_estimate(y = data$y, x = data$covariates,
                            w = data$w,
                            method = "IPTW-GBM",
                            estimand = "ATE")
```

   Then running `iptw_gbm_res` will show the following message:

```
#> This is ATE results from estimation method IPTW-GBM using CIMTx.
#> For effect estimates, please use summary function. To visualize
#> weight, please use plot function.
```

   To compare the estimation from `ce_estimate()` to the true values, we used the same design factors in **Design factors for data simulation** section on **Page 3-4** to simulate the data:

```
library(CIMTx)
data <- data_sim(
  sample_size = 500, n_trt = 3,
  x = c("rnorm(0, 0.5)",  # x1
        "rbeta(2, .4)",   # x2
        "runif(0, 0.5)",  # x3
        "rweibull(1, 2)", # x4
        "rbinom(1, .4)"), # x5
  # linear terms in parallel response surfaces
  lp_y = rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3),
  # nonlinear terms in parallel response surfaces
  nlp_y  = rep(".7*x1*x1  - .1*x2*x3", 3),
  align = F,# different predictors used in treatment and outcome models
  # linear terms in treatment assignment model
  lp_w = c(".4*x1 + .1*x2  - .1*x4 + .1*x5",   # w = 1
           ".2*x1 + .2*x2  - .2*x4 - .3*x5"),  # w = 2
  # nonlinear terms in treatment assignment model
  nlp_w = c("-.5*x1*x4  - .1*x2*x5", # w = 1
            "-.3*x1*x4 + .2*x2*x5"), # w = 2
  tau = c(-1.5, 0, 1.5), delta = c(0.5, 0.5), psi = 1)
```

Then we implemented different methods using `ce_estimate()` function in our **CIMTx** package and repeated the analysis 100 times. Figure 2 shows the biases in three pairwise treatment effect estimates for each method across 100 replications:

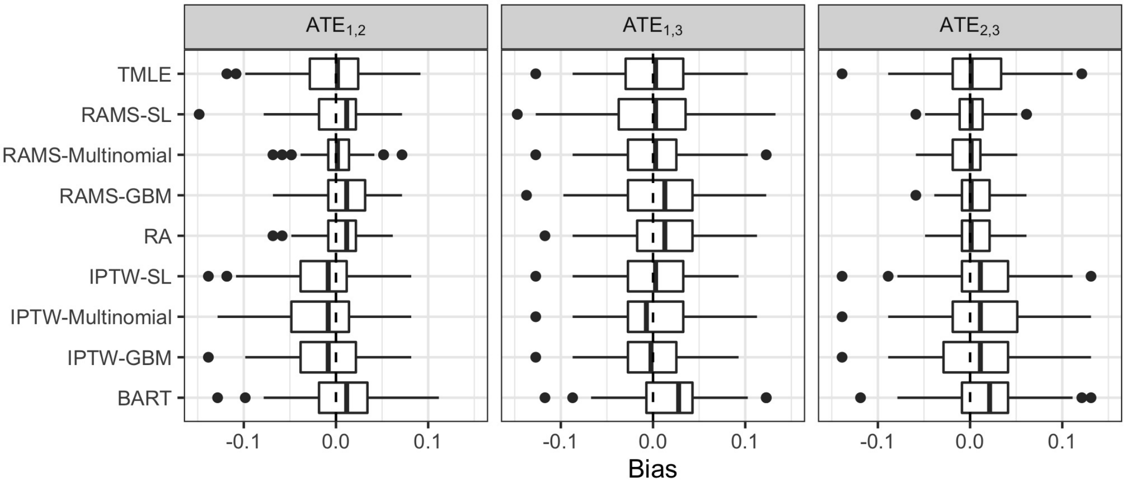

Figure 2: Biases among 100 replications using `ce_estimate()` function from **CIMTx** package. The true value for $ATE_{1,2}$ is -0.322. The true value for $ATE_{1,3}$ is -0.622. The true value for $ATE_{2,3}$ is -0.301.

Note that this is a demonstrative example using a small sample. In-depth methods comparison can be found in Hu and Gu (2021) and Hu et al. (2020).

Refs:

1 L. Hu and C. Gu. Estimation of causal effects of multiple treatments in healthcare database studies with rare outcomes. *Health Services and Outcomes Research Methodology*, 21(3):287–308, 2021.

2 L. Hu, C. Gu, M. Lopez, J. Ji, and J. Wisnivesky. Estimation of causal effects of multiple treatments in observational studies with a binary outcome. *Statistical Methods in Medical Research*, 29(11): 3218–3234, 2020.

3. *I am a bit confused by the sensitivity analysis sections. Based on the description at the beginning of the section on page 5, I would expect that the goal would be to estimate bias under certain assumptions about the unmeasured confounding. It seems, however, that the package (and the algorithm on page 5) is returning adjusted causal effects, and not bias. Then, on page 11, it is not clear how one specifies their beliefs about the confounding in the sa() function until later on page 12. Based on the example on page 11, I briefly thought that sa() was in actuality just implementing some method on the side that was robust to unmeasured confounding, which does not seem to be the case. More exposition about the sensitivity analysis procedure and its implementation is needed to make it clear what the method is doing. This should include some (brief) context and intuition for the algorithm used for sensitivity analysis. What are M1 and M2, for example?*

   Response: We wish to clarify that the goal of our sensitivity analysis approach is to construct confounding function adjusted causal effect estimators (Hu et al. (2022)). That's the reason why we return adjusted causal effects in our `sa()` function. The first example we demonstrated is intended for empirically verifying our proposed sensitivity analysis approach.

   We have added Table 2 on **page 10** to interpret the confounding functions and added the R implementation of key sensitivity analysis steps on **page 11-13**. Please see also our response to Reviewer 1 Comment 2.

   In the algorithm, M_1 is the number of posterior draws for the generalized propensity score. M_2 is the number of draws for the confounding functions $c(w, w', \boldsymbol{x})$ and we have added the definitions for the 2 parameters to **Sensitivity analysis for unmeasured confounding** on **Page 11**:

   > *"The first dimension is the number of posterior draws for the GPS (M_1)."*
   >
   > *"the number of draws (M_2) for the confounding functions $c(w, w', \boldsymbol{x})$."*

   Refs

1 L. Hu, J. Zou, C. Gu, J. Ji, M. Lopez, and M. Kale. A flexible sensitivity analysis approach for unmeasured confounding with multiple treatments and a binary outcome with application to SEER-Medicare lung cancer data. Annals of Applied Statistics, 2022. 16 (2), 1014-1037.

Minor comments

1. *The R Journal does not use numbered sections; please remove references to section numbers. The last paragraph of the introduction, for example, references sections 2.2, 2.3, and 2.4.*

   Response: Thank you for pointing it out. We have removed the references to section numbers.

2. *In Table 1, what are some things the other packages listed do that CIMTx does not? The authors should feel more comfortable discussing limitations of the package: it's okay that CIMTx does not do everything! One obvious comparator where CIMTx would not seem to receive a checkmark is support for continuous outcomes.*

   Response: We have added a column to Table 1 to indicate that our CIMTx package currently does not support continuous outcomes. However, the estimation procedure can be used in the same way to estimate the causal effects of multiple treatment on continuous outcomes. Since we focus on binary outcomes, for which there is less causal literature, we currently do not provide examples for continuous outcomes.

3. *Second sentence of "Estimation of causal effects" section: clarify that when you say "[e]ach individual is exposed to a treatment..." you mean that individuals are exposed to exactly one treatment.*

   Response: We mean each individual was exposed to one and only one treatment. We have clarified it in **Estimation of causal effects** section on **Page 5**:

   > *"Each individual was exposed to one and only one treatment, indexed by W."*

4. *Typo in equation (2): there is an extra closing parenthesis after the last X_i.*

   Response: We have removed the extra closing parenthesis.

5. *The code for the data simulation code on page 7 doesn't run when copy/pasted into R: the closing parenthesis and subsequent comma after the X argument are commented out.*

   Response: We have moved the comments after the closing parenthesis so that users can run the codes by coping and pasting from the manuscript.

6. *I get the following warning when running the VM example from the manuscript (page 9):In Matching::Matchby(Y = eval(parse(text = paste0("temp", reference_trt, :no matches found in group 2 (probably because of the exact or caliper option) continuing.*

   Response: The warning is generated in the 1:1 matching step of the VM method. It means that 1:1 matching algorithm could not find a match in stratum 2 defined by the previous k-means clustering step since the 1:1 matching is conducted within each cluster strata defined by k-means clustering.

7. *Typo in equation (7): the authors use W for treatment indicator throughout, not A.*

   Response: We have corrected the typo in the equation.

8. *Typo in equation (9): in the last line, $X\zeta^{NL}$ should be $X\zeta^{L}$.*

   Response: We have corrected the typo in the equation.

9. *In equations (9), (10), and (11), use $\vdots$ instead of repeated $\cdots$ to indicate the sequence of equations.*

   Response: We have used $\vdots$ to replace the repeated $\cdots$ in the equations.

10. *Second-to-last sentence of the "Design factors (1) – (5)" subsection (page 6): I think the statement that $Y_i = \sum_{w \in W} Y_i(w)$ is an incorrect statement of consistency. There should be an indicator for $W_i = w$ in that sum.*

    Response: We have added the indicator for $W_i = w$ for the sentence you specified.

11. *First sentence on page 10: explicitly name the "other 2 methods" (super learner, GBM).*

    Response: We have explicitly name the "other 2 methods" as super learner and GBM.

12. *First paragraph on page 10: when describing trim_perc, explain what it actually is, rather than just a vector. My understanding is that trim_perc is a 2-vector of percentiles at which to trim the weights; this should be clarified in the text.*

    Response: We have clarified that the argument `trim_perc` is a 2-vector of percentiles at which to trim the weights in both our manuscript and the package documentation.

13. *On page 11, what is true_c_fun_cal()? I cannot find it in the package.*

    Response: Previously we only defined the function `true_c_fun_cal()` in the submitted R script since it only plays a supportive role for the demonstration of our sensitivity analysis. Now we have included the function `true_c_fun_cal()` in our package.

14. *The sa() example on page 11 does not run: x1 is not an available object after simulating the data. The x argument should be changed to (I think) data_SA$covariates$V1. Similarly for the subsequent calls in this section.*

    Response: In the previous submission of this work, to limit our article's length, `x1` was only defined in our accompanying R script. We have now included these codes in our manuscript so that readers can run our codes by just coping and pasting the codes in our manuscript.

15. *The authors may want to consider a conclusion or discussion section summarizing their contribution.*

    Response: We have added a discussion section on **page 15**.

Code: Comments

1. *The code is, in general, poorly documented. In the CRAN source, there are almost no comments explaining what the code does which makes it difficult to understand how the methods are implemented. For instance, there are zero explanatory comments in the 200 lines of code contained in ce_estimate_iptw_ate.R. This makes it very difficult for curious users (and reviewers) to understand how the package works. For non-exported functions, the authors can (and should) still use roxygen2 to document the code and explicitly define imported functions to make dependencies clearer.*

   Response: We have now added explanatory comments for all the functions in our package so that readers can understand how the methods are implemented. We have also added documentation using **roxygen2** for all the non-exported functions. For each R function in our package, we have also explicitly defined the imported functions and R packages using the `@importFrom` command from **roxygen2**.

2. *Code style could be greatly improved by just adding line breaks (breaking a line around the 80-character mark is traditional) and some whitespace. The styler package may be useful here, though its defaults did not do a great job when I ran it on the package myself. General attention to readability is needed throughout the code.*

   Response: Thank your for your suggestion. To improve readability of our R source codes, we first used the **stylr** package to reformat R code. Then we manually added line breaks to our R scripts to avoid lines of code longer than 80 characters.

3. *The authors define their own logit and expit functions (the latter defined inside of the data_sim() function). Instead, use stats::qlogis() and stats::plogis() respectively to avoid needing to define these functions separately.*

   Response: We have used the `stats::qlogis()` and `stats::plogis()` to replace our defined `logit` and `expit` functions in our R package.

4. *The authors should cite important dependencies, both to give appropriate credit, and also to point out when existing methods and packages are being used to implement the causal inference methods. Such dependencies include (but are not limited to) arm, BART, nnet, Matching, twang, SuperLearner. Cite any package that is being used to implement a method critical to the causal effect estimation CIMTx does.*

   Response: Thank you for the constructive suggestion. We have added references to the dependency R packages used in each of the R function in our package with the `@references` command from the **roxygen2** package. Now when the users open up a help page for any of the functions in our package, they will see the references of the dependency R packages for the specific function under the Reference section of the help page.

5. *In the output of data_sim(), the covariates data frame should have names (x1, x2, etc.) which match the "formula" syntax used in, e.g., the lp_y argument.*

   Response: We thank the reviewer for bringing up this concern. We have changed the names of the output data frame as suggested so that they match the names in the `lp_y, nlp_y, lp_w, nlp_w` input arguments.