The multicate R Package: An Introduction

Kyungeun Jeon, Carly Brantner, Daniel Obeng, Elizabeth Stuart

2025-07-29

This guide provides an introduction to the multicate package – an R package for estimation and prediction of heterogeneous treatment effects across one or more studies. The background below is further outlined in the Brantner et al., 2024 paper entitled *Comparison of methods that combine multiple randomized trials to estimate heterogeneous treatment effects*.[1](#fn1) The package can be found [here](https://github.com/dobengjhu/multicate).

Purpose of multicate

Randomized controlled trials (RCTs) are considered the gold standard for unbiased estimation of treatment effects. However, RCTs often have limited power to detect heterogeneous treatment effects (HTE) due to sample size constraints, and they may not reflect populations to which findings will be applied. To address these limitations, researchers have developed various methods to combine data from multiple RCTs to improve treatment effect estimation, including approaches such as meta-analysis. However, these techniques often do not explicitly target conditional average treatment effects and typically rely on aggregate-level data, making it challenging to estimate treatment effects conditional on individual-level characteristics.

Enter multicate. The multicate package implements methodologies that combine multiple studies using individual, patient-level data (IPD). The focal *estimand* in this package is the conditional average treatment effect (CATE), defined under Rubin’s potential outcomes framework.[2](#fn2) We define 𝐴 to represent a binary treatment indicator, 𝑆 a categorical variable for the trial in which the individual participated (from 1 to 𝐾, with a total of 𝐾 RCTs), **𝐗** the covariates, and 𝑌 a continuous outcome. Then, we let 𝑌(0) and 𝑌(1) be the potential outcomes under control and treatment, respectively. Generally, the CATE is defined as the expected difference in the potential outcomes under treatment versus control, *conditional on covariates,* **𝐗**:

𝜏(**𝐗**)=𝐸(𝑌(1)|**𝐗**)−𝐸(𝑌(0)|**𝐗**)

We can also conceptualize the CATE as conditional not only on covariates, but also on study membership, where differences in the treatment effect due to study membership may reflect differences in the distributions of unobserved effect moderators. Here, the CATE can be expressed as

𝜏𝑆(**𝐗**)=𝐸(𝑌(1)|**𝐗**,𝑆)−𝐸(𝑌(0)|**𝐗**,𝑆)

The multicate package can estimates either of these CATE expressions, depending on the specific estimation (or prediction) technique chosen. Notably, while the methods in multicate were originally developed to target CATE estimation across multiple RCTs, they can also be utilized in non-experimental studies. We elaborate upon this in the sections to follow.

Features of multicate

The multicate package enables researchers to combine multiple RCTs to estimate the CATE using various estimation and aggregation methods, while effectively handling heterogeneity in the data. This approach not only increases statistical power but also extends findings from a single study to multiple studies, providing a more robust foundation for personalized decision-making in diverse populations.

* **Goal:** Guide treatment decision-making in a health care or other practical setting by estimating heterogeneous treatment effects
* **Data:** multiple RCTs (or other studies), as well as optionally a new target population
* **Key features:**
  + *Estimation* of CATEs with different combinations of estimation and aggregation methods
  + *Visualization* of CATEs and relevant models
  + *Prediction* of CATEs for a target population

multicate incorporates (1) estimation across multiple studies, (2) estimation of the CATE (beyond the overall average treatment effect), (3) visualizations to increase interpretability of machine learning methods for CATE estimation, and (4) prediction techniques for predicting the CATE in a new target population.

Compared to other packages [3](#fn3) [4](#fn4) [5](#fn5) that more focus primarily on meta-analysis, multicate is unique in that it is specially designed to estimate the CATE by combining multiple studies and predict CATEs in a target population of interest. The aim of this package is to make informed decisions on which treatment may be preferable for a set covariate profiles – profiles of patients defined based on their characteristics. To our knowledge, no other R package currently supports both estimation and prediction of CATEs specifically when combining studies with machine learning methods. A more detailed explanation of the methods, their contributions and limitations is provided in Brantner et al. (2024).[6](#fn6)

A screenshot of a test

AI-generated content may be incorrect.

A comparison table of R packages related to estimation of heterogeneous treatment effects

Process of multicate: Estimation

multicate relies on two primary functions, estimate\_cate() and predict(). They do exactly what you’d expect - estimate and predict the CATE, respectively. Below displays a potential workflow for a two-step approach to use multicate. Notably, researchers do not have to proceed to Step 2 - often a goal is solely focused on estimation of the CATE across multiple studies (Step 1).

A diagram of steps to a training

AI-generated content may be incorrect.

A two-step approach for using the multicate package.

Assumptions

Estimating the CATE across multiple trials rely on a few key assumptions. These include the Stable Unit Treatment Value Assumption (SUTVA), unconfoundedness, consistency, and positivity.

1. **Stable Unit Treatment Value Assumption (SUTVA)**
2. **Unconfoundedness of each study:** 𝑌(0),𝑌(1)⊥𝐴∣**𝐗**
3. **Consistency in each study:** 𝑌=𝐴𝑌(1)+(1−𝐴)𝑌(0) almost surely
4. **Positivity of treatment assignment in each study:** There exists a constant 𝑏>0 such that 𝑏<𝑃(𝐴=1|**𝐗**)<1−𝑏 for all **𝐗**
5. **Positivity of study membership:** There exists a constant 𝑐>0 such that 𝑐<𝑃(𝑆=𝑠|**𝐗**)<1−𝑐 for all **𝐗** and 𝑠.

Unconfoundedness and positivity of treatment assignment are met by design in an RCT, and they can be evaluated in observational studies. Assumption 5, which requires that any covariate **𝐗** is possible to be observed in all studies, may be relaxed depending on the estimation method used.

When multicate is used for predicting the CATE in a target population, we rely on estimation of the CATE first, using the same above assumptions as above. Our only addition is;

1. **Positivity of target population:** There exists a constant 𝑑>0 such that 𝑑<𝑃(𝑆=𝑠|**𝐗**)<1−𝑑 for all **𝐗** *in the target population* and for all 𝑠.

Now to estimate CATEs across multiple studies, we rely on both *estimation* and *aggregation* methods. The *estimation* method is the approach originally geared towards CATE estimation in a single study. The *aggregation* method is the approach that extends the application to multiple studies at once. multicate is currently set up to implement one of multiple options for both estimation and aggregation.

Estimation Methods

When the goal is estimation of the CATE, real-world clinical data often involve complex structures that violate parametric assumptions. Unlike parametric models that require pre-specification of effect moderators and distributional assumptions, nonparametric methods offer greater flexibility, especially in modeling nonlinear relationships between covariates and treatment effects. multicate is set up to apply non-parametric estimation methods to get the CATE using the estimate\_cate() function.

1) S-learner

The S-learner is a ‘meta-learner’ in that it combines base learners of different forms in a special way, and we use this learner to estimate a conditional outcome mean function given observed covariates and assigned treatment as 𝜇(𝑋,𝐴)=𝐸(𝑌|𝑋,𝐴). Then by plugging in 0 and 1 for 𝐴, we can obtain predicted outcomes under treatment and control for each individual and calculate

𝜏̂ =𝜇̂ (𝑋,1)−𝜇̂ (𝑋,0).

In this package, we applied Bayesian additive regression trees (BART) for this base learner. [7](#fn7)

BART, introduced by Chipman et al. (2007, 2010), is a Bayesian nonparametric model that combines a sum of trees approach with regularization priors. [8](#fn8) [9](#fn9) [10](#fn10) These priors shrink the model toward additive structures, reducing overfitting and provides posterior draws of outcomes, enabling the construction of credible intervals. Although the S-learner framework can incorporate various algorithms (e.g., random forests, neural networks), this package specifically uses BART for its flexibility and performance in estimating treatment effect heterogeneity. More technical details and parameter settings can be found in Sparapani et al. (2021) [11](#fn11).

2) Causal forest

The causal forest is similar to a traditional random forest, but the primary estimand is the treatment effect itself, not the outcome mean function. It recursively partitions covariates to best split based on treatment effect heterogeneity. The treatment effect is estimated as the difference in average outcomes between the treatment and control units within each leaf. In other words, the causal forest is the weighted aggregation of many causal trees.[12](#fn12)

Aggregation Methods

This package is especially well-suited for combining multiple studies with various aggregation methods. However, our package also supports settings where individual-level data can be shared across trials.

1) No pooling

When trials are too heterogeneous to justify combining information across studies, it may be preferable to estimate effects separately for each trial. In this case, fitting models within each study independently would be most appropriate. Note that this is not technically an ‘aggregation approach’ since each study is analyzed independently, and no cross-study information is used. You can specify this setting with aggregation\_method = “*studyspecific*”.

2) Pooling with trial indicator

A ‘complete pooling approach’ - combining all data and treating it as a single study - requires strong assumptions. To relax these, our package implements ‘pooling with a trial indicator’. Basically, all of the individual data from all RCTs is combined into one comprehensive dataset, but a categorical study variable is included. This allows researchers to apply single-study approaches while accounting for full covariates including membership indicator. This will yield trial-specific CATE estimates. You can use aggregation\_method = “*studyindicator*” to apply this method.

A diagram of a diagram

AI-generated content may be incorrect.

Overview of the Pooling with Trial Indicator Aggregation Method

3) Ensemble forest method

This method is based on Tan and colleagues’ methods[13](#fn13) for federated learning, devised for scenarios in which individual data cannot be shared across trial sites. First, it builds localized models for CATE within each trial, and then applies these models to all individuals in all studies to get each individual trial-specific CATE estimate. Then, an ensemble model is trained using these estimates as the response variable, with individual covariates and trial indicators as predictors. This method can be selected with aggregation\_method = “*ensembleforest*”.

A diagram of a model

AI-generated content may be incorrect.

Overview of the Ensemble Forest Aggregation Method

Process of multicate: Prediction

Once we have estimated the CATE, a unique feature of the multicate package is its ability to predict CATEs for a target population — such as a new group of patients at baseline in a healthcare system — using models trained on multiple studies. This is particularly useful when models trained on previously conducted studies need to be applied to individuals outside of the original study samples, such as patients from electronic health records (EHR). When multiple studies involve the same treatment options, you can use this package to apply fitted models to these external individuals and guide real-world treatment decisions.

To predict the CATE, we can use the predict() functionality applied to an object resulting from estimate\_cate(). Applying the function to this object alongside a new target setting dataset will yield prediction intervals for CATEs in the new target setting. In predicting the CATE, we leverage prediction interval techniques from two-stage meta-analysis. Briefly, the approach involves two stages:

1. **Stage 1:** Estimating the CATE in each study separately (using estimate\_cate() with aggregation\_method = "studyspecific").
2. **Stage 2:** Summarizing aggregate CATE estimates across studies in a meta-analysis.

From this meta-analysis, we can form prediction intervals around the inverse variance-weighted CATE for a particular covariate profile, estimating the within-study variance from Stage 1 and the between-study variance using restricted maximum likelihood estimation (REML). We replicate the second stage and fit a meta-analysis for every covariate profile (**𝐗\***) of interest in the target setting data to form prediction intervals for each 𝜏(**𝐗\***).

Q&A

We will finish off this introductory vignette with some common questions about multicate.

**Question 1:** What type of studies can I use in estimate\_cate()? How does the package handle observational studies?

You can estimate the CATE using any combination of RCTs and/or observational studies. Notably, the prior simulation work assessing performance of these methods in multiple studies has focused on the use of RCTs. However, the estimation methods of multicate (S-learner with BART, causal forest) can inherently handle confounding in treatment assignment, and users can allow the methods to automatically address this confounding. If desired, propensity scores can be separately estimated and included in one of two ways: as a covariate for S-learner with BART, or as the W.hat argument in the causal forest (see grf documentation).

If you are using just a set of observational studies or combining observational studies with RCTs, exercise caution in implementing the aggregation methods that involve pooling or ensembling (see **Question 2** for more on assumptions). It is important to check assumptions and to ensure that the studies are comparable enough that combining data is reasonable.

**Question 2:** How do I pick which estimation and aggregation method to use?

The answer to this question is somewhat subjective and still in need of future work to verify. When it comes to estimation methods, both the S-Learner with BART and the Causal Forest have been shown to perform well in previous simulations (cite Kunzel paper and Brantner et al 2024). In Kunzel’s paper, the S-Learner was shown to sometimes be biased towards 0, and they recommend using the method if the CATE is hypothesized to be in many instances 0. Otherwise though, either approach can be utilized, and both can be compared to assess sensitivity of results.

For the aggregation method, all three have also been shown to perform well in simulations (cite Brantner et al 2024 paper). We recommend three major factors to consider when deciding which aggregation method to use:

1. **Assumptions:** Assumptions 1-4 above are required for all aggregation methods, but Assumption 5 (positivity of study membership) can be relaxed depending on the approach. Specifically, we do not require that all covariate profiles can be found in all studies for “studyspecific” or “studyindicator”, but we do require this assumption for “ensembleforest” since in “ensembleforest”, each study’s model is applied to all participants across all studies. Therefore, if you have studies without strong overlap in the covariate distributions, you will want to avoid “ensembleforest”.
2. **Pooling data:** Your a priori hypotheses for your data and the variability of the CATE across studies can help inform your model selection. If your studies are very heterogeneous in their covariate distributions and in their hypothesized treatment effects (and treatment effect heterogeneity), it might make sense to not pool the data whatsoever (the “studyspecific” aggregation method). However, the other two aggregation methods allow for some pooling but still pick up on study-level heterogeneity, so for many estimation goals, they will be effective. Of those two methods, the “ensembleforest” method is the most computationally intensive without a notable benefit in simulations over the “studyindicator” technique. But if you are particularly interested in fitting study-specific models before doing any pooling, that is the approach taken by the “ensembleforest”.
3. **Prediction:** If the main goal of your use of multicate is to predict the CATEs for a target population, you should use the “studyspecific” aggregation method. This is what is required for use of the predict() functionality because it applies a two-stage meta-analysis, where the first stage requires estimation of the CATE within each trial on its own. You can always explore the other aggregation approaches within your estimation first, but ultimately you will rely on “studyspecific” in the prediction step.

**Question 3:** When should I use the predict() functionality?

The predict() functionality is meant for a scenario when you are interested in predicting CATEs on a group who is distinct from the original studies that you have complete data on. Often, we may be interested in predicting patients or units at baseline - in this scenario, we can rely on the previously conducted studies to estimate the CATE, and we can predict in our new patients without having seen any treatment or outcomes.

An ideal use case of the prediction would be: using previously conducted studies, we estimate the CATE and subsequently predict in a set of patient profiles who may receive one of two interventions. Using our predictions, we can inform our intervention decision-making in these patients in a more personalized manner (since we are using the CATE as opposed to the ATE), and without relying on them trying the interventions in advance.

**Question 4:** What kind of visualizations does multicate provide?

multicate provides five types of visualization plots, along with a covariate distribution plot across studies: a histogram of estimated conditional average treatment effects (CATEs), a boxplot of CATEs stratified by study membership, a plot of 95% confidence intervals for all CATEs sorted by their estimated values, a best linear projection (available only when estimation\_method = "causalforest"), and an interpretation tree. Additionally, a covariate distribution plot across studies highlights how the distribution of a selected covariate varies across different studies. This helps researchers assess the extent of covariate imbalance or heterogeneity between studies.

A close-up of several graphs

AI-generated content may be incorrect.

Examples of Visualization Outputs from multicate

**Question 5:** What packages does multicate rely on?

multicate builds on established R packages depending on the method used:

* S-learner is implemented using the dbarts package. Default setting include: keeptrees = TRUE, and verbose = FALSE (only when aggregation\_method = "ensembleforest".
* Causal forest is based on the grf package, using its causal\_forest() function with defaults: importance = "impurity" and keep.inbag = TRUE.

For additional details and parameter options, refer to the package documentation. [14](#fn14) [15](#fn15)

Next Up:

If you are ready to try implementing multicate, please see our next vignette here.

1. Brantner, C. L., Nguyen, T. Q., Tang, T., Zhao, C., Hong, H., & Stuart, E. A. (2024). Comparison of methods that combine multiple randomized trials to estimate heterogeneous treatment effects. Statistics in medicine, 43(7), 1291-1314. <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.9955>[↩︎](#fnref1)
2. Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. Journal of educational Psychology, 66(5), 688. <https://psycnet.apa.org/record/1975-06502-001>[↩︎](#fnref2)
3. CausalMetaR CRAN manual <https://cran.r-project.org/web/packages/CausalMetaR/index.html>[↩︎](#fnref3)
4. metacart CRAN manual <https://cran.r-project.org/web/packages/metacart/index.html>[↩︎](#fnref4)
5. personalized CRAN manual <https://cran.r-project.org/web/packages/personalized/index.html>[↩︎](#fnref5)
6. Brantner, C. L., Nguyen, T. Q., Tang, T., Zhao, C., Hong, H., & Stuart, E. A. (2024). Comparison of methods that combine multiple randomized trials to estimate heterogeneous treatment effects. Statistics in medicine, 43(7), 1291-1314. <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.9955>[↩︎](#fnref6)
7. Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. Proceedings of the national academy of sciences, 116(10), 4156-4165. <https://doi.org/10.1073/pnas.1804597116>[↩︎](#fnref7)
8. Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). BART: Bayesian additive regression trees. <https://doi.org/10.1214/09-AOAS285>[↩︎](#fnref8)
9. Hill, J., Linero, A., & Murray, J. (2020). Bayesian additive regression trees: A review and look forward. Annual Review of Statistics and Its Application, 7(1), 251-278. <https://doi.org/10.1146/annurev-statistics-031219-041110>[↩︎](#fnref9)
10. Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. Journal of Computational and Graphical Statistics, 20(1), 217-240. <https://doi.org/10.1198/jcgs.2010.08162>[↩︎](#fnref10)
11. Sparapani, R., Spanbauer, C., & McCulloch, R. (2021). Nonparametric machine learning and efficient computation with Bayesian additive regression trees: The BART R package. Journal of Statistical Software, 97, 1-66. <https://www.jstatsoft.org/article/view/v097i01>[↩︎](#fnref11)
12. Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. <https://www.jstor.org/stable/26581894>[↩︎](#fnref12)
13. Tan, X., Chang, C. C. H., Zhou, L., & Tang, L. (2022, June). A tree-based model averaging approach for personalized treatment effect estimation from heterogeneous data sources. In International Conference on Machine Learning (pp. 21013-21036). PMLR. <https://proceedings.mlr.press/v162/tan22a.html>[↩︎](#fnref13)
14. dbarts CRAN manual <https://cran.r-project.org/web/packages/dbarts/index.html>[↩︎](#fnref14)
15. grf CRAN manual <https://cran.r-project.org/web/packages/grf/index.html>[↩︎](#fnref15)