# `rjaf`: Regularized Joint Assignment Forest with Treatment Arm Clustering

Wenbo Wu [1,¶], **Xinyi Zhang** [1], **Jann Spiess** [2], **and Rahul Ladhania** [3]

1 Department of Population Health, NYU Grossman School of Medicine, USA **2** Graduate School of Business, Stanford University, USA **3** Departments of Health Management and Policy and Biostatistics, University of Michigan School of Public Health, USA **¶** Corresponding author

Summary

Learning optimal assignment of treatments is an important problem in economics, public health, and related fields, particularly when faced with a variety of treatment strategies. The problem arises, for example, in settings where randomized controlled trials (RCT) are conducted to evaluate various behavioral science-informed interventions aimed at fostering behavior change (Milkman, Gromet, et al., 2021). Such interventions have been studied across diverse domains, including encouraging gym attendance and increasing vaccine uptake for influenza or COVID-19 (Dai et al., 2021; Milkman, Gromet, et al., 2021; Milkman, Patel, et al., 2021; Milkman et al., 2022). While most studies focus on identifying interventions that perform best on average, this approach often overlooks effect heterogeneity. Ignoring heterogeneity can be a missed opportunity to tailor interventions for maximum effectiveness and may even exacerbate disparities (Bryan et al., 2021). Subject-specific covariates, such as sociodemographics can be harnessed to identify which interventions work best for different segments of the population, allowing for more impactful intervention assignments. The `rjaf` package provides a user-friendly implementation of the regularized joint assignment forest (RJAF) (Ladhania et al., 2023), a regularized forest-type assignment algorithm based on greedy recursive partitioning (Athey et al., 2019) that shrinks effect estimates across treatment arms. The algorithm is augmented by outcome residualization to reduce baseline variation, and employs a clustering scheme (Hartigan & Wong, 1979) that combines treatment arms with consistently similar outcomes. Personalized treatment learning is achieved by optimizing a regularized empirical analogue of the expected outcome. The integration of R (R Core Team, 2024) and C++ (Stroustrup, 2013) substantially boosts computational efficiency in tree partitioning and aggregating. It is especially suitable in RCT settings with numerous treatment arms and constrained sample sizes, making it a powerful tool for learning personalized intervention strategies.

Statement of Need

There is an ever-growing literature at the intersection of machine learning and causal inference attempting to address the problem of optimal treatment assignment through heterogeneous treatment effect estimation (Athey et al., 2019; Athey & Imbens, 2016; Athey & Wager, 2021; Hitsch & Misra, 2018; Sverdrup et al., 2020; Wager & Athey, 2018; Zhao et al., 2012; Xin Zhou et al., 2017). Among them, the `policytree` approach (Sverdrup et al., 2020) pursues doubly robust estimation that relies on the `grf` package, while the outcome (or residual) weighted learning approach (Zhao et al., 2012; Xin Zhou et al., 2017) is based on support vector machines. Other methods focus on maximizing the benefit (empirical welfare) from treatment assignment (e.g., Kitagawa & Tetenov, 2018), or the chance of assigning an individual to an optimal treatment arm (e.g., Murphy, 2005; Xuan Zhou et al., 2018). Most of these approaches perform well with a limited number of treatment and control groups. A large number of arms

renders the identification of best arm increasingly difficult, and assignments based on separate arm-wise estimation are inefficient. Commonly used implementations such as the multi-arm causal forest (implemented through the `grf` package) (Tibshirani et al., 2022) and random forest (Wright & Ziegler, 2017) might lead to suboptimal assignment, particularly in settings with high noise. By contrast, the RJAF (Ladhania et al., 2023) yields elevated empirical outcome estimates closer to the optimal level from the oracle assignment than the multi-arm causal forest approach in high noise settings, and performs at the same level in low-noise ones. Despite the methodological advantage over existing approaches, the incorporation of machine learning and causal inference techniques such as recursive tree partitioning, bootstrap aggregating, and treatment arm clustering makes it challenging to implement the RJAF from scratch even for well-trained data scientists. The `rjaf` is an open-source software package in R and C++ that efficiently implements the RJAF, offering data scientists a user-friendly analytic toolbox for learning personalized treatment rules in real-world settings.

Workflow

Figure 1 outlines the workflow of the `rjaf` package. The process begins with partitioning the input data—consisting of outcomes, treatment arms, covariates, individual identifiers, and optional probabilities of treatment assignment—into two parts, one for model training and estimation, and the other is the held-out set on which personalized assignment rules are obtained. The `rjaf` function first checks whether outcome residualization for reducing baseline variation should be performed via the `residualize` function, using the `resid` argument. If `resid` is set to TRUE (the default), a new column of residualized outcomes is added to the input data and used for tree growing on the training set. Next, the `rjaf` function evaluates whether treatment clustering should be performed on the training-estimation set during tree growing using the `clus.tree.growing` argument. If `clus.tree.growing` is TRUE, an Rcpp function is employed to estimate cross-validated counterfactual outcomes for the $K + 1$ treatment arms, after which k-means clustering is used to learn $M + 1$ treatment arm clusters. The optimal number of treatment clusters is determined using the elbow method. After clustering, the Rcpp function is reapplied to the preprocessed data, with assignment forest fitted on $M + 1$ treatment clusters and counterfactual outcomes estimated for the original $K + 1$ arms. If `clus.tree.growing` is FALSE, the Rcpp function is employed to estimate counterfactual outcomes for the $K + 1$ arms. Lastly, the Rcpp function is used to obtain optimal treatment arms and predicted counterfactual outcomes under all treatment arms for individuals in the held-out set.
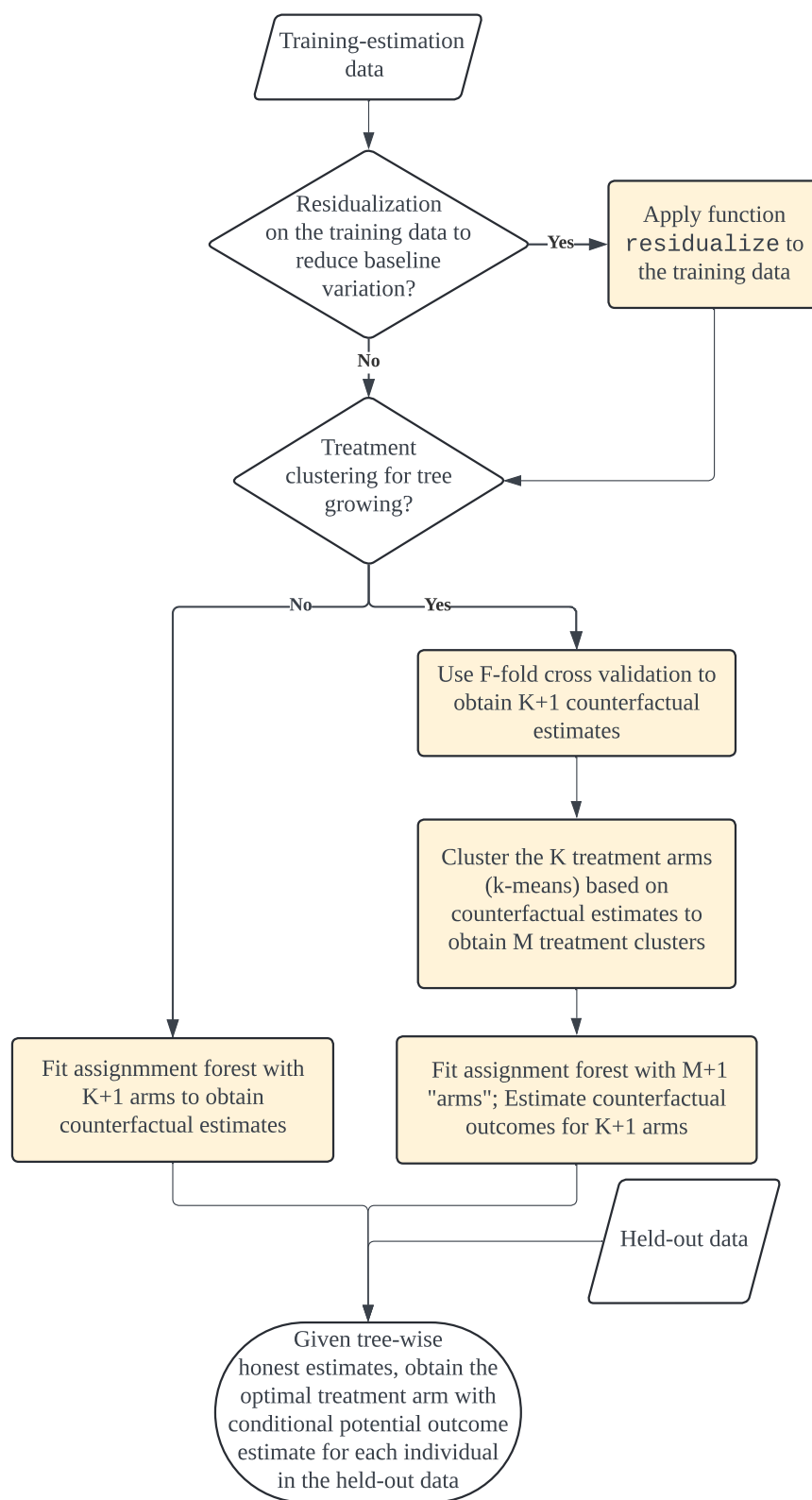
Figure 1: A sketch of the `rjaf` package.

Figure 2 describes the RJAF algorithm. Tree growing begins by taking the training-estimation data set as input, randomly splitting it into separate training and estimation subsets proportionally by treatment arms (or clusters). Initially, utility is set at the root node, where optional Horvitz–Thompson estimator-based inverse probability weighting (IPW) can be applied. A tree is then grown via recursive partitioning of the training subset based on covariate splits. Each potential split is generated by an internal `Rcpp` function, where regularization specified by the `lambda1` parameter can be performed along with IPW to calculate weighted average outcomes by treatment arms or clusters. A potential split is retained if it meets three criteria: (1) each child node contains at least the minimum number of units specified by the `nodesize` argument, (2) the utility gain is at least `eps` times the empirical standard deviation of outcomes in the entire input data, and (3) the child nodes have different optimal treatment arm (or cluster) assignments from the parent node. Recursive partitioning ends when no further splits meet these criteria. Once terminal nodes are determined in the training subset, the learned splitting rules are applied to the estimation subset to assign its units to the terminal nodes. Outcomes from units in the estimation subset are used to calculate treatment-arm-specific average outcomes for each terminal node, with optional regularization specified by the `lambda2` parameter and imputation controlled by `impute`. On the held-out data set, treatment-arm-specific outcomes from the estimation subset are assigned to corresponding terminal nodes to achieve honest outcome estimates, thus concluding the tree growing process. The final step is bootstrap aggregation of a large number of trees, where the total number of trees is set by the `ntree` parameter.
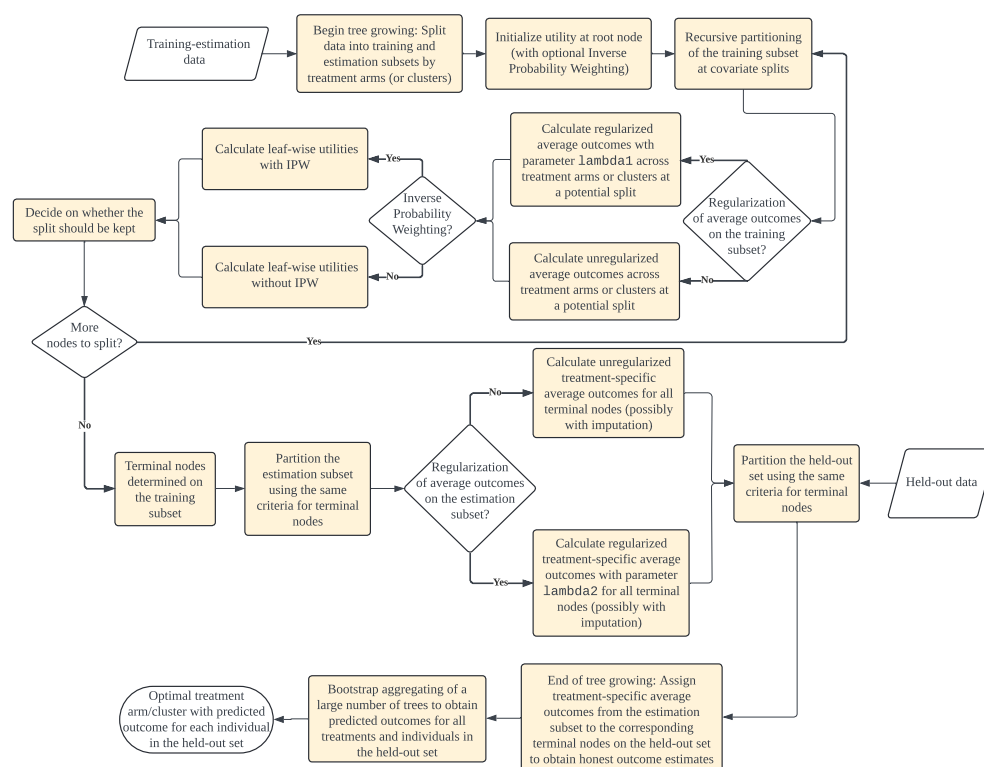

Figure 2: A description of the RJAF algorithm.

Acknowledgments

References

Athey, S., & Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, *113*(27), 7353–7360. https://doi.org/10.1073/pnas.1510489113

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *Annals of Statistics*, *47*(2), 1148–1178. https://doi.org/10.1214/18-AOS1709

Athey, S., & Wager, S. (2021). Policy learning with observational data. *Econometrica*, *89*(1), 133–161. https://doi.org/10.3982/ECTA15732

Bryan, C. J., Tipton, E., & Yeager, D. S. (2021). Behavioural science is unlikely to change the world without a heterogeneity revolution. *Nature Human Behaviour*, *5*(8), 980–989. https://doi.org/10.1038/s41562-021-01143-3

Dai, H., Saccardo, S., Han, M. A., Roh, L., Raja, N., Vangala, S., Modi, H., Pandya, S., Sloyan, M., & Croymans, D. M. (2021). Behavioural nudges increase COVID-19 vaccinations. *Nature*, *597*(7876), 404–409. https://doi.org/10.1038/s41586-021-03843-2

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *28*(1), 100–108. https://doi.org/10.2307/2346830

Hitsch, G. J., & Misra, S. (2018). Heterogeneous treatment effects and optimal targeting policy evaluation. *Available at SSRN*. https://doi.org/10.1007/s11129-023-09278-5

Kitagawa, T., & Tetenov, A. (2018). Who should be treated? Empirical welfare maximization methods for treatment choice. *Econometrica*, *86*(2), 591–616. https://doi.org/10.3982/ECTA13288

Ladhania, R., Spiess, J., Ungar, L., & Wu, W. (2023). *Personalized assignment to one of many treatment arms via regularized and clustered joint assignment forests*. https://doi.org/10.48550/arXiv.2311.00577

Milkman, K. L., Gandhi, L., Patel, M. S., Graci, H. N., Gromet, D. M., Ho, H., Kay, J. S., Lee, T. W., Rothschild, J., Bogard, J. E., & others. (2022). A 680,000-person megastudy of nudges to encourage vaccination in pharmacies. *Proceedings of the National Academy of Sciences*, *119*(6), e2115126119. https://doi.org/10.1073/pnas.2115126119

Milkman, K. L., Gromet, D., Ho, H., Kay, J. S., Lee, T. W., Pandiloski, P., Park, Y., Rai, A., Bazerman, M., Beshears, J., & others. (2021). Megastudies improve the impact of applied behavioural science. *Nature*, *600*(7889), 478–483. https://doi.org/10.1038/s41586-021-04128-4

Milkman, K. L., Patel, M. S., Gandhi, L., Graci, H. N., Gromet, D. M., Ho, H., Kay, J. S., Lee, T. W., Akinola, M., Beshears, J., & others. (2021). A megastudy of text-based nudges encouraging patients to get vaccinated at an upcoming doctor's appointment. *Proceedings of the National Academy of Sciences*, *118*(20), e2101165118. https://doi.org/10.1073/pnas.2101165118

Murphy, S. A. (2005). A generalization error for q-learning. *Journal of Machine Learning Research*, *6*(37), 1073–1097. https://www.jmlr.org/papers/volume6/murphy05a/murphy05a.pdf

R Core Team. (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing; https://www.R-project.org.

Stroustrup, B. (2013). *The C++ Programming Language* (Fourth). Pearson Education. https://www.stroustrup.com/4th.html

Sverdrup, E., Kanodia, A., Zhou, Z., Athey, S., & Wager, S. (2020). Policytree: Policy learning via doubly robust empirical welfare maximization over trees. *Journal of Open Source Software*, *5*(50), 2232. https://doi.org/10.21105/joss.02232

Tibshirani, J., Athey, S., Friedberg, R., Hadad, V., Hirshberg, D., Miner, L., Sverdrup, E., Wager, S., & Wright, M. (2022). *grf: Generalized Random Forests*. https://CRAN.R-project.org/package=grf. https://doi.org/10.32614/cran.package.grf

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*(523), 1228–1242. https://doi.org/10.1080/01621459.2017.1319839

Wright, M. N., & Ziegler, A. (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *Journal of Statistical Software*, *77*(1), 1–17. https://doi.org/10.18637/jss.v077.i01

Zhao, Y., Zeng, D., Rush, A. J., & Kosorok, M. R. (2012). Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, *107*(499), 1106–1118. https://doi.org/10.1080/01621459.2012.695674

Zhou, Xin, Mayer-Hamblett, N., Khan, U., & Kosorok, M. R. (2017). Residual weighted learning for estimating individualized treatment rules. *Journal of the American Statistical Association*, *112*(517), 169–187. https://doi.org/10.1080/01621459.2015.1093947

Zhou, Xuan, Wang, Y., & Zeng, D. (2018). *Sequential outcome-weighted multicategory learning for estimating optimal individualized treatment rules*. http://www.columbia.edu/~yw2016/SOM_ITR.pdf