The Journal of Open Source Software

dte_adj: A Python Package for Estimating Distributional Treatment Effects in Randomized Experiments

Tomu Hirata [1,3¶], **Undral Byambadalai** [1], **Tatsushi Oka** [1,2], **and Shota Yasui** [1]

1 CyberAgent, Inc., Japan **2** Keio University, Japan **3** Databricks, Inc., United States ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

dte_adj is a Python package for estimating distributional treatment effects (DTEs) in randomized experiments (RCTs, also known as A/B tests). Unlike traditional approaches that focus on average treatment effects, dte_adj enables researchers to analyze the full distributional impact of interventions across different outcome levels. The package implements machine learning-enhanced regression adjustment methods for variance reduction, supports multiple experimental designs including simple randomization, covariate-adaptive randomization, and settings with imperfect compliance, and provides a scikit-learn compatible API with comprehensive functionality for computing distribution functions, probability treatment effects, and quantile treatment effects with confidence intervals.

Statement of Need

Randomized experiments have been fundamental to scientific inquiry since Fisher (1935), providing the gold standard for causal inference. While most experimental analyses focus on average treatment effects (ATEs), many research questions require understanding how treatments affect the entire distribution of outcomes. Distributional treatment effects (DTEs) capture these richer patterns, revealing heterogeneous impacts across different outcome levels that averages can mask. For example, a policy intervention might have no effect on average income while substantially reducing poverty rates at lower quantiles, or a medical treatment might benefit patients at the tails of the distribution differently than those near the median.

Despite the growing importance of distributional analysis in economics, medicine, and technology, the Python ecosystem lacks comprehensive tools for DTE estimation with modern variance reduction techniques. Researchers often resort to basic empirical CDFs or manual implementations that lack statistical rigor. dte_adj fills this gap by providing a unified framework for distributional treatment effect analysis that integrates state-of-the-art machine learning methods for improved precision, rigorous confidence interval construction, and support for complex experimental designs.

State of the Field

Several Python packages address causal inference, but none focus on distributional treatment effects with machine learning-based variance reduction:

- **SciPy** (Virtanen et al., 2020): Provides basic empirical cumulative distribution functions but offers no functionality for treatment effect estimation or confidence interval construction in experimental settings.

- **DoWhy** (Sharma & Kiciman, 2020): Focuses on causal graph-based inference and average treatment effects, without distributional analysis capabilities.
- **EconML** (Battocchi et al., 2021): Incorporates machine learning for heterogeneous treatment effect estimation (CATE) but does not address distributional effects.
- **causal-curve** (Kobrosly, 2020): Estimates dose-response curves but targets continuous treatments rather than distributional outcomes.

In the R ecosystem, packages like `qte` provide quantile treatment effect estimation but lack machine learning integration for variance reduction. `dte_adj` uniquely combines: (1) distributional treatment effect estimation across the full outcome distribution, (2) machine learning-enhanced regression adjustment for precision gains, and (3) support for multiple experimental designs including covariate-adaptive randomization and imperfect compliance settings.

Software Design

`dte_adj` follows a class-based architecture with a template method pattern, where a base class defines the algorithm structure and subclasses implement design-specific computations:

- **`SimpleDistributionEstimator`** and **`AdjustedDistributionEstimator`**: For simple randomized experiments, implementing methods from Byambadalai et al. (2024).
- **`SimpleStratifiedDistributionEstimator`** and **`AdjustedStratifiedDistributionEstimator`**: For covariate-adaptive randomization designs, implementing methods from Byambadalai et al. (2025b).
- **`SimpleLocalDistributionEstimator`** and **`AdjustedLocalDistributionEstimator`**: For settings with imperfect compliance, implementing methods from Byambadalai et al. (2025a).

All estimators implement a consistent API with three primary methods: `predict_dte()` for distributional treatment effects, `predict_pte()` for probability treatment effects over intervals, and `predict_qte()` for quantile treatment effects. The adjusted estimators use K-fold cross-fitting to prevent overfitting and support both single-task and multi-task learning modes (Hirata et al., 2025) for computational efficiency. Bootstrap methods provide confidence intervals with multiple variance estimation approaches.

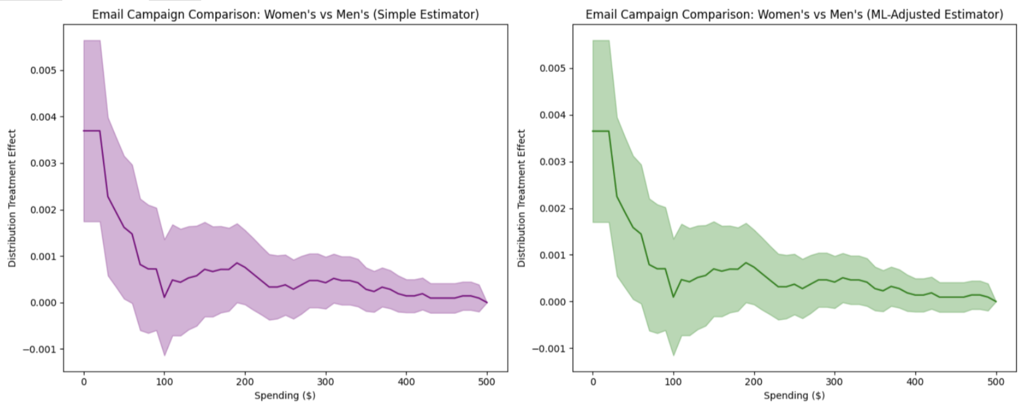

Figure 1: Distributional treatment effects for the Hillstrom email marketing dataset (Hillstrom, 2008), comparing Women's vs Men's email campaigns. The simple estimator (left, purple) and ML-adjusted estimator (right, green) show that adjustment substantially tightens confidence bands, demonstrating the variance reduction benefit of regression adjustment.

The Journal of Open Source Software

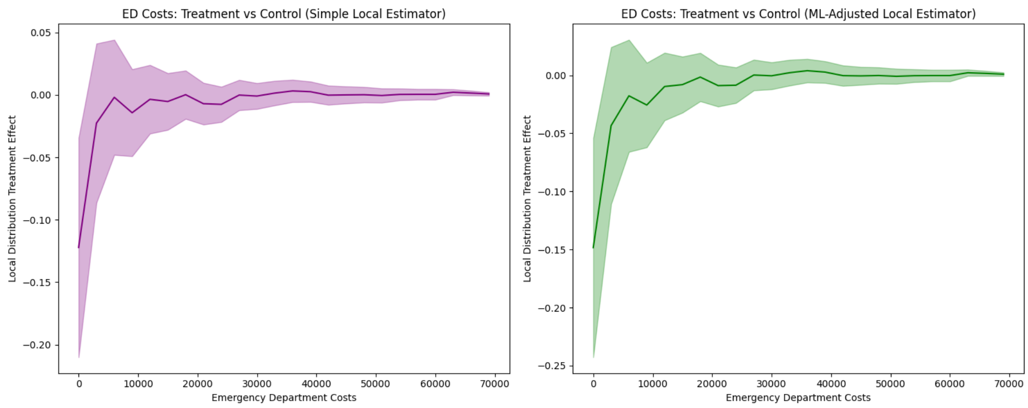

Figure 2: Local distributional treatment effects for emergency department costs in the Oregon Health Insurance Experiment (Finkelstein et al., 2012), estimated using `SimpleLocalDistributionEstimator` (left) and `AdjustedLocalDistributionEstimator` (right). Health insurance coverage shifts the distribution of ED costs, with ML adjustment again yielding narrower confidence intervals.

Research Impact Statement

The methods implemented in `dte_adj` have been published at top machine learning venues: ICML 2024 (Byambadalai et al., 2024) and ICML 2025 (Byambadalai et al., 2025b). The package has been used internally at CyberAgent, Inc. for analyzing A/B tests where distributional impacts are critical, such as evaluating interventions on user engagement metrics where tail behavior matters more than averages. The documentation includes tutorials demonstrating applications to the Hillstrom email marketing dataset (Figure 1) and the Oregon Health Insurance Experiment (Figure 2), facilitating adoption by researchers in economics, marketing, and healthcare.

AI Usage Disclosure

Generative AI tools (Claude) were used to assist with documentation writing and code review during development. All AI-generated content was reviewed and validated by the human authors.

Acknowledgements

We thank CyberAgent, Inc. for supporting this research and the open-source community for valuable feedback during development.

References

Battocchi, K., Dillon, E., Hei, M., Lewis, G., Ling, M., Rao, J., Shyr, D., & Syrgkanis, V. (2021). Causal inference and machine learning in practice with EconML and CausalML: Industrial use cases at Microsoft, TripAdvisor, Uber. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 4072–4073.

Byambadalai, U., Hirata, T., Oka, T., & Yasui, S. (2025a). *Beyond the average: Distributional causal inference under imperfect compliance*. https://arxiv.org/abs/2509.15594

Byambadalai, U., Hirata, T., Oka, T., & Yasui, S. (2025b). On efficient estimation of distributional treatment effects under covariate-adaptive randomization. *Proceedings of the*

42nd International Conference on Machine Learning. https://arxiv.org/abs/2506.05945

Byambadalai, U., Oka, T., & Yasui, S. (2024). *Estimating distributional treatment effects in randomized experiments: Machine learning for variance reduction*. https://arxiv.org/abs/2407.16037

Finkelstein, A., Taubman, S., Wright, B., Bernstein, M., Gruber, J., Newhouse, J. P., Allen, H., Baicker, K., & Oregon Health Study Group. (2012). The oregon health insurance experiment: Evidence from the first year. *The Quarterly Journal of Economics*, *127*(3), 1057–1106. https://doi.org/10.1093/qje/qjs020

Fisher, R. A. (1935). *The design of experiments*. Oliver; Boyd.

Hillstrom, K. (2008). *The MineThatData e-mail analytics and data mining challenge*. https://blog.minethatdata.com/2008/03/minethatdata-e-mail-analytics-and-data.html

Hirata, T., Byambadalai, U., Oka, T., Yasui, S., & Uto, S. (2025). *Efficient and scalable estimation of distributional treatment effects with multi-task neural networks*. https://arxiv.org/abs/2507.07738

Kobrosly, R. W. (2020). Causal-curve: A python causal inference package to estimate causal dose-response curves. *Journal of Open Source Software*, *5*(52), 2523. https://doi.org/10.21105/joss.02523

Sharma, A., & Kiciman, E. (2020). DoWhy: An end-to-end library for causal inference. *arXiv Preprint arXiv:2011.04216*. https://arxiv.org/abs/2011.04216

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. van der, Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2