Double Machine Learning at Scale to Predict Causal Impact of Customer Actions

Sushant More[0000−0002−3746−2431]✉, Priya Kotwal[0009−0004−6599−359X], Sujith Chappidi[0009−0009−3310−6067], Dinesh Mandalapu[0009−0007−2984−859X], and Chris Khawand[0009−0000−5283−9391]

Amazon, Seattle WA, USA
{morsusha,kotwalp,jcchappi,mandalap,khawandc}@amazon.com

Abstract. Causal Impact (CI) measurement is broadly used across the industry to inform both short- and long-term investment decisions of various types. In this paper, we apply the double machine learning (DML) methodology to estimate average and conditional average treatment effects across 100s of customer action types for ecommerce and digital businesses and 100s of millions of customers that can be used in decisions supporting those busiensses. We operationalize DML through a causal machine learning library. It uses distributed computation on Spark and is configured via a flexible, JSON-driven model configuration approach to estimate causal impacts at scale (i.e., across hundred of actions and millions of customers). We outline the DML methodology and implementation. We show examples of average treatment effect and conditional average treatment effect (i.e., customer-level) estimates values along with confidence intervals. Our validation metrics show a 2.2% gain over the baseline methods and a 2.5X gain in the computational time. Our contribution is to advance the scalable application of CI, while also providing an interface that allows faster experimentation, ability to onboard new use cases, and improved accessibility of underlying code for partner teams.

Keywords: Double Machine Learning · Potential Outcomes · Heterogeneous treatment effect · Invserse propenity weighting · Placebo tests.

1 Introduction

Causal Impact (CI) is a measure of the incremental change in a customer's outcomes (usually spend or profit) from a customer event or action (e.g, signing up for a paid membership). Business leaders commonly use some version of CI values as important signals for driving various decisions, such as marketing content ranking or capital investments.

The CI values are leveraged by partner teams to understand and improve the value they generate through customer behaviors. Some examples include customer actions such as 'first purchase in category X', 'first stream in service Y', or 'sign up for program Z'[1]. For many of these customer actions, we are unable

[1] We use placeholder X,Y,Z to maintain business confidentiality

to conduct A/B experiments due to practical or legal constraints. We instead use observational data, effectively leveraging rich customer data to isolate causal relationships in the absence of a randomized experiment.

In this paper, we provide results for average treatment effects and conditional average treatment effects (i.e., customer-level CI values) estimated using a variant on the Double Machine Learning (DML) methodology [1]. The paper is arranged as follows. In Sec. 2, we give a brief overview of the use of causal measurement. In Sec. 2.1, we introduce an example of a conventional regression-and-propensity-adjustment system used for calculating CI values. We discuss the shortcomings of the traditional method and the advantages of moving to DML.

Sec. 3 covers the details of our DML implementation for calculating CI values. Our contributions include improving the robustness of CI estimates through inverse propensity weighting, adding the ability to produce heterogeneous CI values, implementing customer-level confidence intervals with various assumptions, and making available the JSON Machine Learning interface to accelerate experimentation. We present results in Sec. 4 for a few customer actions and conclude with the takeaways and ideas for future work in Sec. 6.

2 Causal impact estimation in industry

Causal impact estimation drives a large number of business decisions across industry. This includes multiple organizations such as retail, search, devices, streaming services, and operations. To this end, most companies have invested in developing and deploying models that vend CI values for the cutsomer actions under consideration. In the next section, we give an overview of the traditional potential-outcome based model which is widely used in the industry for CI estimation. This will be the baseline model.

2.1 CI: P-score Binning and Regression adjustment framework

CI framework applies the principles of observational causal inference. We rely on it because A/B testing is not possible to evaluate the impact of certain treatments due to practical constraints (e.g., the treatment is not effectively assignable, or would be too expensive to assign at scale). Observational causal inference methods rely on eliminating potential confounders through adjustment on observed variables. Under a "selection on observables" assumption, we believe we can estimate the causal effect correctly on average. Applied to the customer's next 365 days of spending, for example, the CI value represents the incremental spending that a customer makes because of participating in a certain action compared to the counterfactual case where they didn't take that particular action. The formal framework for this kind of counterfactual reasoning is the "potential outcomes" framework, sometimes known as the Neyman-Rubin causal framework [3], [4], [5].

There are many procedures aimed at estimating potential outcomes. One example estimator is a combination of propensity score stratification and regression adjustment:

1. Propensity binning. Group the customer based on their propensity to participate in the action. This is done based on features that relate to recency, frequency, and the monetary behavior of customers along with their other characteristics such as their tenure type.
2. Regression adjustment. In each of the groups, we build a regression model on the control customers with customer-spend as the target. The trained model is applied on the treatment customers to predict the counterfactual spend (how much would customer have spent if they didn't participate in the action). The difference between the predicted counterfactual and the actual spend is the CI value. We take a weighted average across different groups to get the final CI value for the customer action.

In addition, we require the CI model to be able to scale to the business use case. For instance, we may want generate CI values for hundreds of customer actions in an automated way. In the rest of the paper, we refer to this estimation procedure as "CI-PB" (short for "propensity binning") and the DML-based estimator as "CI-DML".

3 CI: DML framework

Note that one of the challenges in validating the causal estimates is posed by the *Fundamental Problem of Causal Inference* [2]. The lack of observable ground truth makes it difficult to validate the output of a causal model, but well-constructed procedures can at least provide some guarantees of causally interpretable estimates under certain assumptions. The Double/Debiased Machine learning (DML) method proposed by Chernozhukov et al. [1] leverages the predictive power of modern Machine Learning (ML) methods in a principled causal estimation framework that is free of regularization bias asymptotically.

For treatment D, features X, we express the outcome Y as an additively separable function of D and arbitrary function of features X:

$$Y = D\beta + g(X) + \epsilon \tag{1}$$

DML's estimation strategy is motivated by writing out the residualized representation of Eq. (1) and its parts:

$$\tilde{Y} = Y - E(Y|X) \tag{2}$$

$$\tilde{D} = D - E(D|X) \tag{3}$$

$$\tilde{Y} = \tilde{D}\beta + \tilde{\epsilon} \tag{4}$$

We use ML models to estimate $E(Y|X)$ and $E(D|X)$. The residuals from outcome equation (Eq. (3)) are regressed on residuals from propensity equation (Eq. (4)) to obtain the causal parameter β. We use ML models to predict $E(Y|X)$ and $E(D|X)$. We leverage K-fold sample splitting so that training and scoring of the ML models happens on different folds. We use a 3-fold sample split and follow the "DML2" approach [1] where we pool the residuals outcome and propensity residuals across all the folds to fit a single, final regression of the residualized outcome on the residualized treatment (Eq. (4)).

3.1 Inverse Propensity Treatment Weighting

We use a weighted ordinary least squares to solve the residual regression equation (Eq. (4)). where the weights are determined by the Inverse Propensity Treatment Weighting (IPTW or IPW) [12]. Our IPTW weights correspond to the Horvitz-Thompson (HT) weight [13], in which the weight for each unit is the inverse of the probability of that unit being assigned to the observed group. In Table 1 we define the weights that balance the distributions of covariates between comparison groups for two widely used estimands, the Average Treatment Effect (ATE) and Average Treatment Effect on the Treated (ATT). Weighting helps achieve additional robustness, bringing us closer to a conventional Doubly Robust estimator. Applying these weights when conducting statistical tests or regression models helps reduce impact of confounders over and above what we get from the regression adjustment [14]. Secondly, the weights allow us to target the estimand; we prefer the ATT since it represents the treatment effects for those customers actually treated historically who are marginally closer to those who will be treated next. We refer to the customer-level counterparts of ATE, ATT estimands as HTE and HTT respectively.

Table 1. IPW weights for different estimands. D is the treatment assignment and $\hat{e}(X)$ is the treatment propensity, $E(D|X)$.

Estimand	IPW weight
Average treatment effect (ATE)	$\frac{D}{\hat{e}(X)} + \frac{1-D}{1-\hat{e}(X)}$
Average treatment effect on the treated (ATT)	$D + (1-D)\frac{\hat{e}(X)}{1-\hat{e}(X)}$

3.2 Common support and propensity score trimming

For many treatments, propensity distribution has significant mass near 1 for the treated group and near 0 for the control group (see an example histogram in Fig. 1). Scores near the boundary can create instability in weighting methods. In addition, these scores often represent units for whom we cannot make an adequate treatment-control comparison. We limit analysis to the common support region, where propensity score distributions overlap between treated and untreated samples.

We also use trimming to exclude customers whose estimated propensity is outside of the range $[\alpha, 1-\alpha]$. We experimented with different thresholds on various customer actions and observed that $\alpha = 0.001$ with rescaled propensity scores works the best.

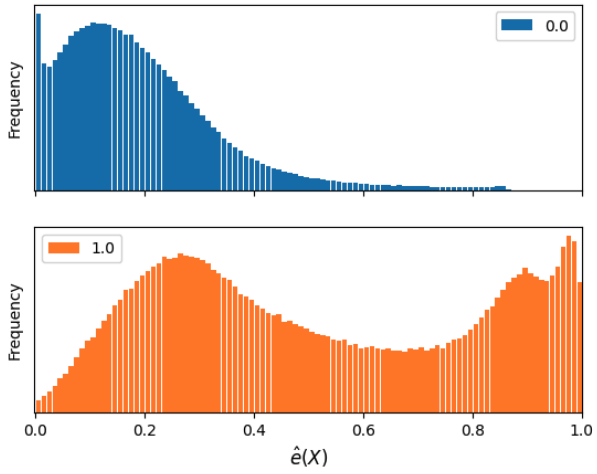

Fig. 1. Representiative propensity scores distribution for control (top panel) and treatment (bottom panel) groups.

Normalizing and rescaling weights When using the IPW, we normalize the weights by rescaling the propensity scores for each customer i as in Eq. (5).

$$\hat{e}(X_i)_{\text{scaled}} = \left(\frac{\overline{D}}{\overline{\hat{e}(X)}} \right) * \hat{e}(X_i) \tag{5}$$

$\overline{D}$ and $\overline{\hat{e}(X)}$ in Eq. (5) are the averages of treatment assignment and propensity score respectively taken over both the treatment and control population combined. Propensity trimming and rescaling reduces variance, leads to more stable estimates, and tighter confidence intervals as seen in Fig. 2.

trimming boundary	CI-PO ATT	CI-DML ATT	CI-DML confidence interval	outcome model: R-squared	propensity model: ROC-AUC
(0.001, 0.999)	230.29	239.93	(224.47, 255.39)	0.52	0.79
(0.005, 0.995)	230.29	236.64	(222.31, 250.96)	0.52	0.79
no trimming and rescaling	230.29	844.13	(-243.58, 1931.84)	0.52	0.79

Fig. 2. Effect of propensity scores trimming and rescaling on estimated CI for a certain customer action.

3.3 Heterogeneity in DML

CI-DML implements a version of the heterogenous effects modeling proposed in [6], by leveraging the treatment-feature interactions in the final stage of DML

to identify heterogenous (customer-level) responses. The general form of Eq. (4) can be written as

$$\tilde{Y} = h(X, \tilde{D}) + \tilde{\epsilon}\ . \tag{6}$$

In fact, Eq. (4) is a special case of Eq. (6) with $h(X, \tilde{D}) = \tilde{D}\beta$. We interact treatment with the features and define $h(X, \tilde{D}) \equiv \psi(X) * \tilde{D}\,\beta$, where '$*$' represents element-wise multiplication. Thus, the heterogeneous residual regression becomes:

$$\tilde{Y} = \psi(X) * \tilde{D}\,\beta + \tilde{\epsilon} \tag{7}$$

We want $\psi(X)$ to be low-dimensional so that we are able to extract the coefficient β in Eq. (7) reliably.

Let N and M be the number of customers and features respectively. If the dimension of $\psi(X)$ is $N \times K$, we want $K \ll M$. In our use case, M is typically 2000 and K is typically around 20. To get the low-dimensional representation, $\psi(X)$ we proceed as follows:

1. We project the original features onto an orthogonal space through Principal Component Analysis (PCA).
2. We run a K-means clustering algorithm on the highest-signal Principal Components. Dimension reduction from PCA helps to reduce dimensionality-related problems when computing Euclidean distance for K-means clustering.
3. We calculate the K cluster scores for each customer, as $\psi_{i,c} = \dfrac{1/d_{i,c}}{\sum_{k=1}^{K} 1/d_{i,k}}$ where $d_{i,c}$ is the distance of customer i's value from centroid of cluster c (see schematic in Fig. 3).

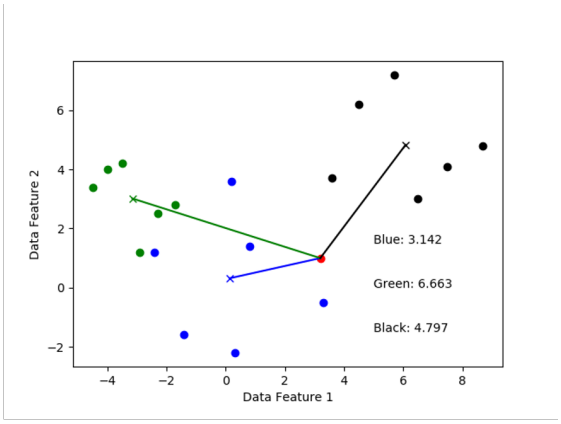

Fig. 3. Schematic for calculation of distance from cluster centroids. The red dot is represented by three features which is the distance from centroids from blue, green, and black clusters.

Once we have calculated the distance features $\psi(X)$ for each customer, we interact them with the propensity residuals and fit a linear regression model using

IPW (refer sec. 3.1) to extract the coefficients β in Eq. (7). The heterogenous estimates are given by

$$h = \psi(X)\beta \ . \tag{8}$$

E.g., for $K = 3$, $h = \psi_1\beta_1 + \psi_2\beta_2 + \psi_3\beta_3$.

A schematic of CI-DML workflow is shown in Fig. 4.

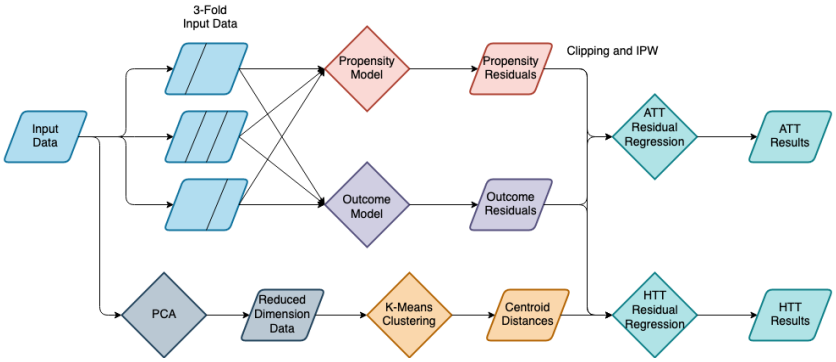

Fig. 4. Schematic of the CI-DML modeling framework.

3.4 Confidence intervals in DML

One of the disadvantages of CI-PB is that generating confidence intervals requires bootstrapping around the multi-step process and is computationally expensive. Obtaining confidence intervals in CI-DML is straightforward. For a single ATT parameter estimate, we obtain the confidence interval simply by calculating the variance of the estimate of β in Eq. (4). We also estimate Huber-White heteroscedasticity consistent standard errors [7], [8]. For the ATT case, the steps for calculating variance of the coefficient $\hat{\beta}$ are as follows:

$$Var(\beta) = H\hat{\Sigma}H'. \tag{9}$$

For a customer, 'p':

$$\hat{\sigma}_p^2 = \hat{U}_p^2 = (\tilde{Y}_p - \tilde{D}_p\hat{\beta})'(\tilde{Y}_p - \tilde{D}_p\hat{\beta}) \ , \tag{10}$$

where $\hat{\beta}$ is the value of coefficient from solving Eq. (4). Note that $\tilde{Y}_p$ and $\tilde{D}_p$ are scalars. Σ in Eq. (9) is a diagonal matrix with the squared prediction error $\hat{\sigma}_p^2$ for each customer on its diagonal and H in Eq. (9) is defined as

$$H = (\tilde{D}' * W\tilde{D})^{-1}\tilde{D}' * W \tag{11}$$

where W are the IPW weights as defined in Sec. 3.1.

We compute the confidence intervals on the causal estimate β using $Var(\beta)$.

Customer-level confidence intervals CI-DML also provides the ability to obtain customer-level confidence intervals. From Eq. (8), we can write

$$Var(h) = Var(\psi\beta) = \sum_k \psi_k^2 Var(\beta_k) + \sum_{k \neq l} \psi_k \psi_l Cov(\beta_k, \beta_l). \tag{12}$$

We calculate variance of the heterogeneous coefficients following similar approach as in Eqs. 9, 10, and 11. The only difference is we replace $\tilde{D}_p \rightarrow \psi(X_p) * \tilde{D}_p$ in Eq. (10) and $\tilde{D} \rightarrow \psi(X) * \tilde{D}$ in Eq. (11).

For the ATT case, $Var(\beta)$ is a scalar whereas for the HTT case, $Var(\beta)$ is a $K \times K$ matrix. The first and second terms in the summation in Eq. (12) are the diagonal and the off-diagonal terms of the $Var(\beta)$ matrix respectively.

3.5 DML implementation

We developed a causal ML library with JSON driven modeling configuration (see Fig. 5). JSON ML Interpreter (JMI) translates JSON configuration to executable Python ML application.

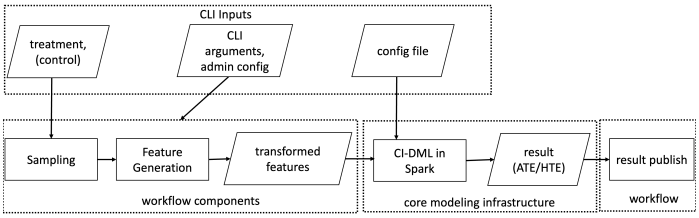

Fig. 5. JSON-Machine Learning Stage Interpreter modeling stages

The main advantages of JMI approach are:
Flexibility: Business questions from various domains cannot always be addressed through a single unified configuration of a causal model. We address this in our system where users can invoke different causal analysis frameworks (DML, Causal Forests) and prediction algorithm type (regression, classification, clustering).
Scalability: CI-DML utilizes distributed implementation of algorithms and file system via Apache Spark which helps causal modeling at the big-data scale (100 millions customers, multiple targets, and time horizons)
Persistence: CI-DML inherits SparkML serialization and deserialization methods to persist and instantiate fitted models for live or batch inference.
Compatibility: In addition to Spark, interfaces to adapt ML libraries from scikit-learn, tensorflow, and MXNet, and other communities can be onboarded using the configurable abstraction support by JMI.

In the system, we dockerize the JMI Causal ML library which is platform agnostic and has the flexibility to extend and utilize different compute engines like AWS EMR, Sagemaker or AWS Batch based on the use case and will abstract

the computation information from the user. Dockerization also helps version control and the build environment via standard software development tooling.

4 Results

Next we present the results for CI-DML. The target variable we focus on is customer spending, but our framework on can be leveraged to obtain the causal impact on any other target variable of interest (e.g., net profit, units bought etc). For every CI run, we produce both the population-level ATT values (Eq. (4)) and the customer-level HTT (Eq. (8)) values.

We compared the CI-PB and CI-DML results for 100+ customer actions. As noted earlier, two major advantages of CI-DML are the availability of customer grain results (aka. HTT) and confidence intervals. In Fig. 6, we present population-level (ATT) and customer-level values for selected representative customer actions [2]. The reported confidence intervals are for both homoscedastic and heteroscedastic error variances. To get a sense of the level of variance in customer-grain results, we report the percentage of customers where the customer-level confidence interval crosses zero. We also report the out-of-sample fit metrics for outcome and propensity models in DML.

Action	CI-PO (ATT)	CI-DML (ATT)	CI-DML (HTT mean)	CI-DML confidence interval (homoscedastic)	CI-DML confidence interval (heteroscedastic)	%customer-level conf. intervals crossing zero (heteroscedastic)	R-squared (outcome model)	ROC-AUC (propensity model)
Action 1	230.3	235.0	229.1	(233.9, 243.5)	(220.1, 249.9)	0.001	0.523	0.790
Action 2	186.5	197.3	181.2	(192.2, 200.0)	(178.0, 216.6)	1.99	0.499	0.818
Action 3	116.3	108.9	103.8	(92.5, 101.9)	(87.2, 130.6)	16.37	0.472	0.943
Action 4	180.6	186.5	229.8	(185.3, 195.9)	(161.9, 211.1)	5.96	0.153	0.957
Action 5	125.7	117.0	160.9	(173.4, 180.7)	(154.2, 218.3)	36.70	0.663	0.869
Action 6	12.3	21.0	21.1	(16.4, 24.4)	(9.0, 33.0)	76.95	0.748	0.885
Action 7	7.6	19.1	6.4	(14.1, 24.2)	(-1.7, 36.4)	83.17	0.738	0.913
Action 8	146.8	138.7	137.9	(129.8, 147.6)	(109.1, 160.8)	20.88	0.674	0.946
Action 9	89.3	86.5	101.1	(82.9, 93.1)	(61.0, 112.1)	45.63	0.726	0.903
Action 10	383.7	298.9	291.3	(295.5, 306.4)	(285.1, 312.6)	0.14	0.671	0.854

Fig. 6. CI values and confidence intervals for selected customer actions.

Our takeaways from the analysis of 100+ actions are:

1. Population-level CI-PB and CI-DML values are aligned for 86% of actions.
2. When the customer-level CI values are aggregated up, they are generally aligned with the population-level CI-DML values.

[2] We anonymize actions to preserve business confidentiality

3. The difference between the CI-PB and CI-DML values are larger either when the data is noisy and/or we have a small sample size. For such cases, we also see large confidence intervals and the mean of HTT values is farther away from the CI-DML ATT values.
4. The homoscedastic confidence intervals are tighter than the heteroscedastic confidence interval as expected. However, the homoscedastic confidence intervals likely under-predict the variance. We recommend business stakeholders to use the heteroscedasticity-robust confidence intervals.
5. The customer-level confidence intervals are economically reasonable. The percentage of customer-level confidence interval crossing zero increases for data with lower participation and is small for customer actions with a long history.
6. The ML model metrics shown in Fig. 6 are using ridge regression for the outcome model and logistic regression for the propensity model. We noticed that the model metrics as well as the CI values are relatively insensitive to the choice of ML model at the outcome/propensity stage. Accordingly, we leverage ridge and logistic models due to their favorable compute time.

4.1 Hyperparameter tuning

The hyperparameters (e.g., regularization strength) in the outcome and propensity model are chosen based on the out-of-sample performance. For the HTT estimates, the two main hyperparameters are the number of principal components and the number of clusters.

We choose the number of principal components (PC) based on the percentage of variance explained. We find that around 300 PC, about 80% of the variance is explained (Fig. 7). The amount of variance explained grows much slowly as we add more number of PC. To avoid sparsity issues in the downstream K-means calculation, we choose the number of PC components to be 300.

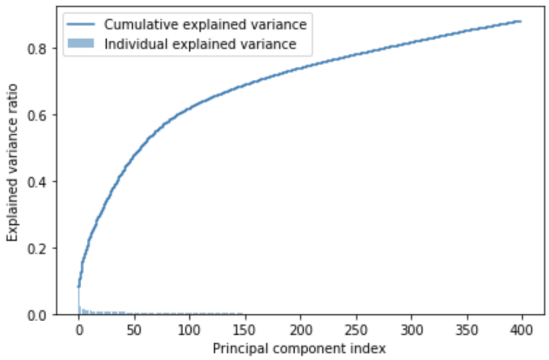

Fig. 7. Amount of variance explained as a function of principal components.

Choosing the number of clusters is less straightforward. Standard tools such as elbow method and Silhouette score do not yield a clear answer for the optimal cluster choice. In the current work, we choose 20 clusters, since we do not see much change in any form of out of sample fit statistics beyond 20. We also find that the mean of HTT values is robust with respect to the choice for number of clusters. In future work, we aim to make this choice in a more data-driven way (e.g., by evaluating how output scores perform in a downstream use case measured through A/B tests), since there may be important variation in the quality of output for decisions that is not picked up by conventional fit statistics.

4.2 Spread of customer-level CI values

So far, we have only looked at the mean of customer-level values in Fig 6. In Fig. 8, we look at the customer-level CI scores for an example customer action. We see that most of the customers have CI value close to the average HTT value. We see that there are few customers with a low CI value which shifts the mean to the left.

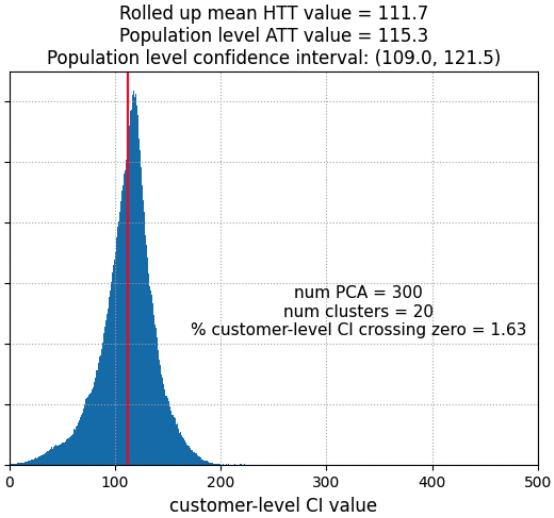

Fig. 8. Spread of customer-level CI values. The red line is the mean of customer-level CI values.

5 Validation

5.1 Placebo tests

Placebo tests help us understand the relative ability of competing causal estimates to account for selection bias. Selection bias occurs when customers who take

an action (e.g., stream video) have unobserved characteristics not included in the model that makes them systematically more or less likely to take the action (e.g., high income, low age etc.). In placebo tests, we take the treatment group customers and simulate as though they took the action a year before the actual date. This is achieved by shifting the event date by one year and recalculating the features based on the shifted event date. The CI estimated in this set up is the "placebo error". Since, this is a fake event, a model with a lower placebo error than another on the same underlying data suggests that it has smaller contribution from selection bias in its estimate. Running placebo tests on all events is computationally expensive, so we selected a few events for placebo analysis. The results are shown in Fig. 9.

Customer Action	Placebo CI-PO	Placebo CI-DML	% Improvement
Action A	191.1	190.7	0.22
Action B	327.6	320.9	2.05
Action C	100.2	98.8	1.40
Action D	204.2	192.7	5.67
Action E	23.7	21.9	7.56
Action F	315.7	390.8	1.86
Action G	143.5	137.3	4.29
Action H	615.5	602.8	2.07
Action I	272.5	269.9	0.94

Fig. 9. Placebo results for selected customer actions for CI-PB and CI-DML model.

The key takeaways from Fig. 9 are:

- Selection bias is inherently event-dependent. When averaged across the selected customer actions, we see a 2.24% improvement in placebo estimates when going from CI-PB to CI-DML.
- Selection bias is primarily impacted by the modeling features. As CI-PB and CI-DML use the same features, we did not expect big improvements in placebo tests. The consistent improvement across the events shows that double machine learning methodology is better able to adjust for observables even when the same features are used.

5.2 Confidence interval comparison

One of the major wins in CI-DML is that we provide heteroskedasticity-consistent confidence intervals at both a customer and aggregate level for every CI analysis in a scalable and lower-cost fashion. We compare the uncertainty estimates (specifically the width of confidence intervals) from CI-DML with the bootstrap results in CI-PB for a few events in Fig. 10.

We find confidence interval width to be comparable among the two approaches. On average, the CI-DML width (scaled with CI-PB point estimate) is 1.5% smaller

Action	CI-PO	confidence interval width CI-PO (bootstrap)	confidence interval width CI-DML (heterosckedastic)	% diff. in conf. interval over CI-PO point estimate
Action I	201.4	15.6	10.4	2.6
Action II	271.9	21.5	27.7	-2.3
Action III	123.7	8.2	11.5	-2.6
Action IV	80.0	39.3	20.2	23.9
Action V	340.5	41.6	56.6	-4.4
Action VI	180.4	7.0	7.7	-0.4
Action VII	221.6	11.4	10.5	0.4
Action VIII	209.2	10.8	8.2	1.3
Action IX	619.8	17.6	29.9	-2.0
Action X	268.0	12.6	10.3	0.8

Fig. 10. CI-PB bootstrap vs. CI-DML confidence interval width comparison

when compared to bootstrap-based confidence interval. A bootstrap-enabled CI-PB run takes about 2.5X more time than a CI-DML run. Bootstrap also does not scale for events with large number of customers. As CI-DML approach for confidence intervals is based on a closed form implementation, we do not have any scalability issues. In addition, note that bootstrapping has theoretical limitations when used for matching estimators [16].

6 Conclusion and Future work

In this work, we introduced a state-of-the-art methodology used for calculating CI values. We noted that a DML based framework eliminates bias, allows us to extract heterogeneity in CI values, and provides a scalable way to construct heteroscedastic confidence intervals. We also made a case for using IPW and common support to refine the CI estimates. We demonstrated how leveraging PCA followed by K-means clustering allowed us to introduce customer-level heterogeneity. Using JSON based config allows flexibility to experiment with a wide variety algorithms and can take us from experimentation to production in minimal steps.

We presented results for few anonymized customer actions across different domains. Both the population-level and customer-level results for the customer actions we have looked at so far are aligned with the CI-PB results and our expectations, but we now can take advantage of convenient calculation of confidence intervals, estimates of heterogeneous treatment effects, and greater scalability.

Note that estimation of heterogeneous or context-aware treatment effects is an active area of research with wide applications ranging from marketing to health care. Distribution of treatment effects across different subgroups, or as a function of specific individual-level characteristics provides researchers with additional insights about the treatment/ intervention analyzed. Our work showcases a scalable real-world application for extracting average as well heterogeneous causal effects which we believe will be of interest to the broader scientific community.

6.1 Future work

Validating the causal estimates is challenging due to lack of ground truth. In the current work, we relied on the model fit metrics in the DML steps, placebo tests, and on bridging the CI-DML and CI-PB outputs. In the future, we plan to include metrics which focus on the validation of heterogeneous treatment effects. Examples of these include metrics based on Generic Machine Learning [17] and empirically calibrated Monte Carlo resampling techniques [18].

Appendix

A Sample JSON config

We show a snippet of JSON config in Fig. 11. We can swap the specified models in the outcome and propensity step with any ML model. Likewise we can easily configure pre/post-processing steps and hyperparameters through JSON files.

```
{
    "macros": {
        "FEATURE_COLS":              {"regex": "^pre_.+$"},
        "STANDARDIZED_FEATURE_COLS": {"regex": "^standardized_pre_.+$"},
        "OUTCOME_COLS":              {"regex": "^post_.+$"},
        "TREATMENT_COLS":            {"regex": "^treatment_.+$"}
    },

    "stages": [
        "standardizer", "propensity_model", "propensity_metrics", "outcome_model",
        "outcome_metrics", "residualer"
        ],

    "standardizer": {
        "module": "sklearn.preprocessing.StandardScaler",
        "with_mean": true, "with_std": true, "inference": "transform",
        "featureCols":    "FEATURE_COLS",
        "predictionCols": {"reference": "featureCols", "format": "standardized_%s"}
    },

    "propensity_model": {
        "module": "sklearn.linear_model.LogisticRegression",
        "fit_intercept": true, "solver": "sag", "n_jobs": -1, "inference": "probability",
        "featureCols": "STANDARDIZED_FEATURE_COLS", "targetCols": "TREATMENT_COLS",
        "predictionCols": {"reference": "targetCols", "format": "propensity_%s"}
    },

    "propensity_metrics": {"include": "configuration/metrics/propensity_sklearn.json"},

    "outcome_model": {
        "module": "sklearn.linear_model.Ridge",
        "alpha": 0.01, "fit_intercept": true, "normalize": false, "solver": "cholesky",
        "featureCols": "STANDARDIZED_FEATURE_COLS", "targetCols": "OUTCOME_COLS",
        "predictionCols": {"reference": "targetCols", "format": "prediction_%s"}
    },

    "outcome_metrics": {"include": "configuration/metrics/outcome_sklearn.json"},

    "residualer": {
        "module": "transformers.BinaryOperatorPandas", "operator": "-",
        "leftOperandCols":  ["OUTCOME_COLS", "TREATMENT_COLS"],
        "rightOperandCols": [{"regex": "^prediction_.+$"}, {"regex": "^propensity_.+$"}],
        "resultCols":       {"reference": "leftOperandCols", "format": "residual_%s"}
    }
}
```

Fig. 11. A sample JSON config where we are using Ridge regression for the outcome model and the logistic regression for the propensity model.

References

1. Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, Volume 21, Issue 1, 1 February 2018, Pages C1–C68, doi.org/10.1111/ectj.12097
2. Sekhon, Jasjeet. The Neyman–Rubin Model of Causal Inference and Estimation via Matching Methods. 2007. *The Oxford Handbook of Political Methodology.*
3. Holland, Paul W. Statistics and Causal Inference. 1986. *J. Amer. Statist. Assoc.* 81 (396): 945–960. doi:10.1080/01621459.1986.10478354
4. Neyman, Jerzy. Sur les applications de la theorie des probabilites aux experiences agricoles: Essai des principes. Master's Thesis (1923). Excerpts reprinted in English, *Statistical Science*, Vol. 5, pp. 463–472.
5. Rubin, Donald. Causal Inference Using Potential Outcomes. 2005. *J. Amer. Statist. Assoc.* 81 (396): 945–960. doi:10.1080/01621459.1986.10478354
6. Chernozhukov, V., Goldman, M., Semenova, V., and Taddy, M. (2017). Orthogonal Machine Learning for Demand Estimation: High Dimensional Causal Inference in Dynamic Panels. ArXiv:1712.09988 [Stat].
7. Huber, Peter J. The behavior of maximum likelihood estimates under nonstandard conditions. (1967) *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability.* Vol. 5. pp. 221–233
8. White, Halbert. A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. *Econometrica* 48 (4): 817–838
9. Horvitz, D. G., and Thompson, D. J. . A Generalization of Sampling Without Re-placement From a Finite Universe (1952). *Journal of the American Statistical Association*, 47(260), 663-685
10. Nie, X., and Wager, S. Quasi-Oracle Estimation of Heterogeneous Treatment Effects (2017). ArXiv:1712.04912 [Econ, Math, Stat]
11. Edward H. Kennedy, Optimal doubly robust estimation of heterogeneous causal effects (2020). ArXiv:2004.14497 [math.ST]
12. Estimating exposure effects by modelling the expectation of exposure conditional on confounders. Biometrics 48 479–495. MR1173493
13. Ruth C, Brownell M, Isbister J, MacWilliam L, Gammon H, Singal D, Soodeen R, McGowan K, Kulbaba C, Boriskewich E. Long-Term Outcomes Of Manitoba's Insight Mentoring Program: A Comparative Statistical Analysis . Winnipeg, MB: Manitoba Centre for Health Policy, 2015
14. P. C. Austin and E. A. Stuart, Moving towards Best Practice When using Inverse Probability of Treatment Weighting (IPTW) Using the Propensity Score to Estimate Causal Treatment Effects in Observational Studies, vol. 34, pp. 3661-3679, 2015.
15. Hirano, K., Imbens, G.W. Estimation of Causal Effects using Propensity Score Weighting: An Application to Data on Right Heart Catheterization. Health Services and Outcomes Research Methodology 2, 259–278 (2001). doi.org/10.1023/A:1020371312283
16. A. Abadie and G. Imbens, On the failure of bootstrap for matching estimators, Econometrica, Vol. 76, No. 6 (2008), 1537-1157
17. Chernozhukov, Victor, Mert Demirer, Esther Duflo, and Ivan Fernandez-Val (2022). Generic machine learning inference on heterogenous treatment effects in randomized experiments. aXiv:1712.04802v6 [stat.ML]

18. Knaus, M. C., Lechner, M., & Strittmatter, A. (2021). Machine learning estimation of heterogeneous causal effects: Empirical monte carlo evidence. The Econometrics Journal, 24(1), 134-161

Ethical Implication

The data presented in the paper is completely anonymized. It cannot be used for inference of personal information of any kind.

The method presented in this paper falls in the domain of observational causal inference. Observational causal inference methods are used to gauge impact of things already happened. The inference methods by itself do not aid in any wrong doing. But in the unfortunate case of bad things happening to an individual (e.g., unfair economic policy/ smoking/ abuse), the causal methods can help identify the impact and help guide the recovery methods. In that sense, work presented here can be used to seek justice for the victim.

Of course, as a society we want to make sure that we do not subject individuals to an unscrupulous treatment to extract the causal impact of that treatment. Because the impact of such treatment could be adverse in some cases. But again the work presented here is used to analyze the aftermath of an action/ treatment. The type of treatments a person can be subjected to is outside the scope of current work.