DoWhy: An End-to-End Library for Causal Inference

Amit Sharma, Emre Kiciman

Microsoft Research

Abstract

*In addition to efficient statistical estimators of a treatment's effect, successful application of causal inference requires specifying assumptions about the mechanisms underlying observed data and testing whether they are valid, and to what extent. However, most libraries for causal inference focus only on the task of providing powerful statistical estimators. We describe DoWhy, an open-source Python library that is built with causal assumptions as its first-class citizens, based on the formal framework of causal graphs to specify and test causal assumptions. DoWhy presents an API for the four steps common to any causal analysis—1) **modeling** the data using a causal graph and structural assumptions, 2) **identifying** whether the desired effect is estimable under the causal model, 3) **estimating** the effect using statistical estimators, and finally 4) **refuting** the obtained estimate through robustness checks and sensitivity analyses. In particular, DoWhy implements a number of robustness checks including placebo tests, bootstrap tests, and tests for unoberved confounding. DoWhy is an extensible library that supports interoperability with other implementations, such as EconML and CausalML for the the estimation step. The library is available at https://github.com/microsoft/dowhy.*

1 INTRODUCTION

Many questions in data science are fundamentally causal questions, such as the impact of a marketing campaign or a new product feature, the reasons for customer churn, which drug may work best for which patient, and so on. As the field of data science has grown, many practitioners are realizing the value of causal inference in providing insights from data. However, unlike the streamlined experience for supervised machine learning with libraries like Tensorflow (Abadi et al., 2016) and PyTorch (Paszke et al., 2019), it is non-trivial to build a causal inference analysis. Software libraries that implement state-of-the art causal inference methods can accelerate the adoption of causal inference among data analysts in both industry and academia.

However, we find that for data scientists and machine learning engineers familiar with non-causal methods and unpracticed in the use of causal methods, one of the biggest challenges is the practice of modeling assumptions (i.e., translating domain knowledge into a causal graph) and the implications of these assumptions for causal identification and estimation. *What is the right model?* Another challenge is in the shift in verification and testing practicalities. Unlike supervised machine learning models that can be validated using held-out test data, causal tasks often have no ground truth answer available. Thus, checking core assumptions and applying sensitivity tests is critical to gaining confidence in results. *But how to check those assumptions?*

Therefore, we built DoWhy, an end-to-end library for causal analysis that builds on the latest research in modeling assumptions and robustness checks (Athey and Imbens, 2017; Kıcıman and Sharma, 2018), and provides an easy interface for analysts to follow the best practices of causal inference. Specifically, DoWhy's API is organized around the four key steps that are required for any causal analysis: Model, Identify, Estimate, and Refute. **Model** encodes prior knowledge as a formal causal graph, **identify** uses graph-based methods to identify the causal effect, **estimate** uses statistical methods for estimating the identified estimand, and finally **refute** tries to refute the obtained estimate by testing robustness to initial model's assumptions.

The focus on all the four steps, going from data to the final causal estimate (along with a measure of its robustness) is the key differentiator for DoWhy, compared to many existing libraries for causal inference in Python and R that only focus on estimation (the third step). These libraries expect an analyst to have already figured out how to build a reasonable causal model from data and domain knowledge, and to have identified the correct estimand. More critically, they also assume that the analyst may perform their own sensitivity and robustness checks, but provide no guidance on their own; which makes it hard to verify and build robust causal analyses.

Under the hood, DoWhy builds on two of the most powerful frameworks for causal inference: graphical models (Pearl, 2009) and potential outcomes (Imbens and Rubin, 2015). It uses graph-based criteria and do-calculus for modeling assumptions and identifying a non-parametric causal effect. For estimation, it switches to methods based primarily on potential outcomes. DoWhy is also built to be interoperable with other libraries that implement the estimation step. It currently supports calling EconML (Microsoft-Research, 2019) and CausalML (Chen et al., 2020) estimators.

To summarize, DoWhy provides a unified interface for causal inference methods and automatically tests many assumptions, thus making inference accessible to non-experts. DoWhy is available open-source on Github, ***https://github.com/microsoft/dowhy***, and has a growing community, including over 2300 stars and 31 contributors. Many people have made key contributions that are improving the usability and functionality of the library such as an integrated Pandas interface for DoWhy's four steps, and we welcome more community contributions. The library makes three key contributions:

1. Provides a principled way of modeling a given problem as a causal graph so that all assumptions are explicit, and identifying a desired causal effect.
2. Provides a unified interface for many popular causal inference *estimation* methods, combining the two major frameworks of graphical models and potential outcomes.
3. Automatically tests for the validity of causal assumptions if possible and assesses the robustness of the estimate to violations.

2 DOWHY AND THE FOUR STEPS OF CAUSAL INFERENCE

DoWhy is based on a simple unifying language for causal inference. Causal inference may seem tricky, but almost all methods follow four key steps. Figure 1 shows a schematic of the DoWhy analysis pipeline.

I. Model the causal question. DoWhy creates an underlying causal graphical model (Pearl, 2009) for each problem. This serves to make each causal assumption explicit. This graph need not be complete—an analyst may provide a partial graph, representing prior knowledge about some of the variables. DoWhy automatically considers the rest of the variables as potential confounders.

II. Identify the causal estimand. Based on the causal graph, DoWhy finds all possible ways of identifying a desired causal effect based on the graphical model. It uses graph-based criteria and do-calculus to find potential ways find expressions that can identify the causal effect.

Supported identification criteria are,

- Back-door criterion
- Front-door criterion
- Instrumental Variables
- Mediation (Direct and indirect effect identification)

III. Estimate the causal effect. DoWhy supports methods based on both back-door criterion and instrumental variables. It also provides a non-parametric confidence intervals and a permutation test for testing the statistical significance of obtained estimate.

Supported estimation methods include,

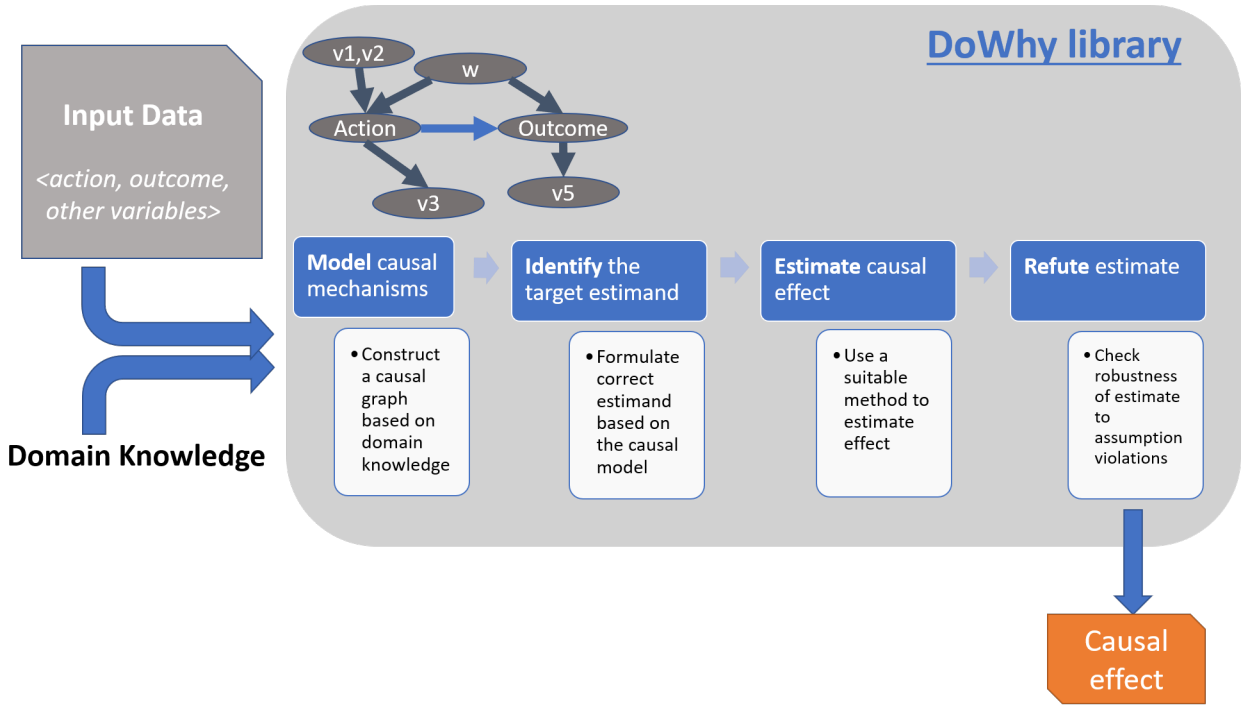

Figure 1: The four-step analysis pipeline in DoWhy.

- Methods based on estimating the treatment assignment: Propensity-based Stratification, Propensity Score Matching, Inverse Propensity Weighting
- Methods based on estimating the outcome model: Linear Regression, Generalized Linear Models
- Methods based on the instrumental variables identification: Binary Instrument/Wald Estimator, Two-stage least squares, Regression discontinuity
- Methods for front-door criterion and general mediation: Two-stage linear regression

In addition, DoWhy support integrations with the EconML and CausalML packages for estimating the conditional average treatment effect (CATE). All estimators from these libraries can be directly called from DoWhy.

IV. Refute the obtained estimate. Having access to multiple refutation methods to validate an effect estimate from a causal estimator is a key benefit of using DoWhy.

Supported refutation methods include:

- **Add Random Common Cause:** Does the estimation method change its estimate after we add an independent random variable as a common cause to the dataset? (Hint: It should not)
- **Placebo Treatment**: What happens to the estimated causal effect when we replace the true treatment variable with an independent random variable? (Hint: the effect should go to zero)
- **Dummy Outcome:** What happens to the estimated causal effect when we replace the true outcome variable with an independent random variable? (Hint: The effect should go to zero)
- **Simulated Outcome:** What happens to the estimated causal effect when we replace the outcome with a simulated outcome based on a known data-generating process closest to the given dataset? (Hint: It should match the effect parameter from the data-generating process)
- **Add Unobserved Common Causes:** How sensitive is the effect estimate when we add an additional common cause (confounder) to the dataset that is correlated with the treatment and the outcome? (Hint: It should not be too sensitive)

- **Data Subsets Validation:** Does the estimated effect change significantly when we replace the given dataset with a randomly selected subset? (Hint: It should not)
- **Bootstrap Validation**: Does the estimated effect change significantly when we replace the given dataset with bootstrapped samples from the same dataset? (Hint: It should not)

Many of the above methods aim to refute the full causal analysis, including modeling, identification and estimation (as in Placebo Treatment or Dummy Outcome) whereas others refute a specific step (e.g., Data Subsets and Bootstrap Validation that test only the estimation step).

3 AN EXAMPLE CAUSAL ANALYSIS

In this section, we show how causal inference using DoWhy simplifies to four lines of code, each corresponding to one of the four steps. Each analysis starts with a building a causal model. The assumptions can be viewed graphically or in terms of conditional independence statements. Wherever possible, DoWhy can also automatically test for stated assumptions using observed data.

```
# I. Create a causal model from the data and given graph.
model = CausalModel(
    data=data["df"],
    treatment=data["treatment_name"],
    outcome=data["outcome_name"],
    graph=data["gml_graph"])
```

Given the model, identification is a causal problem. Estimation is simply a statistical problem. DoWhy respects this boundary and treats them separately. This focuses the causal inference effort on identification, and frees up estimation using any available statistical estimator for a target estimand. In addition, multiple estimation methods can be used for a single identified estimand and vice-versa.

```
# II. Identify causal effect and return target estimands
identified_estimand = model.identify_effect()

# III. Estimate the target estimand using a statistical method.
estimate = model.estimate_effect(identified_estimand,
                                 method_name="backdoor.propensity_score_stratification")
```

For data with high-dimensional confounders, machine learning-based estimators may be more effective. Therefore, DoWhy supports calling estimators from other libraries like EconML. Here is an example of using the double machine learning estimator (Chernozhukov et al., 2017).

```
import econml
dml_estimate = model.estimate_effect(
                identified_estimand, method_name="backdoor.econml.dml.DMLCateEstimator",
                confidence_intervals=False,
                method_params={
                    "init_params":{
                        'model_y':GradientBoostingRegressor(),
                        'model_t': GradientBoostingRegressor(),
                        'model_final':LassoCV(),
                        'featurizer':PolynomialFeatures(degree=1, include_bias=True)},
                    "fit_params":{}})
print(dml_estimate)
```

The most critical, and often skipped, part of causal analysis is checking the robustness of an estimate to unverified assumptions. DoWhy makes it easy to automatically run sensitivity and robustness checks on the obtained estimate.

```
# IV. Refute the obtained estimate using multiple robustness checks.
refute_results = model.refute_estimate(identified_estimand, estimate,
                                       method_name="random_common_cause")
```

Finally, DoWhy is easily extensible, allowing other implementations of the four verbs to co-exist. The four verbs are mutually independent, so their implementations can be combined in any way. Example notebooks of using DoWhy for different causal problems are available at https://github.com/microsoft/dowhy/tree/master/docs/source/example_notebooks.

4 CONCLUSION

We presented DoWhy, an extensible and end-to-end library for causal inference. Unlike most other libraries, DoWhy focuses on helping an analyst devise the correct causal model and test its assumptions, in addition to estimating the causal effect. We look forward to extending DoWhy with more refutation and robustness analyses, and supporting more estimation methods with its 4-step API.

ACKNOWLEDGEMENTS

A big thanks to all the open-source contributors to DoWhy that continue to make important additions to the library's functionality and usability. The list of contributors is updated at https://github.com/microsoft/dowhy/blob/master/CONTRIBUTING.md.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In: *12th {usenix} symposium on operating systems design and implementation ({osdi} 16)*. 2016, 265–283.

Athey, S., & Imbens, G. W. (2017). The state of applied econometrics: Causality and policy evaluation. *Journal of Economic Perspectives*, *31*(2), 3–32.

Chen, H., Harinen, T., Lee, J.-Y., Yung, M., & Zhao, Z. (2020). Causalml: Python package for causal machine learning.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., & Newey, W. (2017). Double/debiased/neyman machine learning of treatment effects. *American Economic Review*, *107*(5), 261–65.

Imbens, G. W., & Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

Kıcıman, E., & Sharma, A. (2018). Tutorial on causal inference and counterfactual reasoning. https://causalinference.gitlab.io/kdd-tutorial/

Microsoft-Research. (2019). EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation [Version 0.6].

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In: *Advances in neural information processing systems*. 2019, 8026–8037.

Pearl, J. (2009). *Causality*. Cambridge university press.