The Journal of Open Source Software

Interfere: Intervention Response Simulation and Prediction for Stochastic Non-Linear Dynamics

D. J. Passey [1*] **and Peter J. Mucha** [2*]

1 University of North Carolina at Chapel Hill, United States **2** Dartmouth College, United States
* These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Scientific models possess several properties that make them notoriously difficult to test, including a complex input space, long execution times, and non-determinism, rendering existing testing techniques impractical. In fields such as epidemiology, where researchers seek answers to challenging causal questions, a statistical methodology known as Causal Inference (CI) (Hernán & Robins, 2020; Pearl, 2009) has addressed similar problems, enabling the inference of causal conclusions from noisy, biased, and sparse observational data instead of costly randomised trials. CI works by using domain knowledge to identify and mitigate for biases in the data, enabling them to answer causal questions that concern the effect of changing some feature on the observed outcome. The Causal Testing Framework (CTF) is a software testing framework that uses CI techniques to establish causal effects between software variables from pre-existing runtime data rather than having to collect bespoke, highly curated datasets especially for testing.

Statement of need

Metamorphic Testing (Chen et al., 1998) is a popular technique for testing computational models (and other traditionally "hard to test" software). Test goals are expressed as *metamorphic relations* that specify how changing an input in a particular way should affect the software output. Nondeterministic software can be tested using Statistical Metamorphic Testing (Guderlei & Mayer, 2007), which uses statistical tests over multiple executions of the software to determine whether the specified metamorphic relations hold. However, this requires the software to be executed repeatedly for each set of parameters of interest, so is computationally expensive, and is constrained to testing properties over software inputs that can be directly and precisely controlled. Statistical Metamorphic Testing cannot be used to test properties that relate internal variables or outputs to each other, since these cannot be controlled a priori.

By employing domain knowledge in the form of a causal graph—a lightweight model specifying the expected relationships between key software variables—the CTF overcomes the limitations of Statistical Metamorphic Testing by enabling models to be tested using pre-existing runtime data. The CTF is written in Python but is language agnostic in terms of the system under test. All that is required is a set of properties to be validated, a causal model, and a set of software runtime data.

Causal Testing

Causal Testing (Clark, Foster, Prifling, et al., 2023) has four main steps, outlined in Figure 1. Firstly, the user supplies a causal model, which takes the form of a directed acyclic graph

The Journal of Open Source Software

(DAG) (Pearl, 2009) where an edge $X \rightarrow Y$ represents variable X having a direct causal effect on variable Y. Secondly, the user supplies a set of causal properties to be tested. Such properties can be generated from the causal DAG (Clark, Foster, Walkinshaw, et al., 2023): for each $X \rightarrow Y$ edge, a test to validate the presence of a causal effect is generated, and for each missing edge, a test to validate independence is generated. The user may also refine tests to validate the nature of a particular relationship. Next, the user supplies a set of runtime data in the form of a table with each column representing a variable and rows containing the value of each variable for a particular run of the software. Finally, the CTF automatically validates the causal properties by using the causal DAG to identify a statistical estimand (Pearl, 2009) (essentially a set of features in the data which must be controlled for), calculate a causal effect estimate from the supplied data, and validating this against the expected causal relationship.

Causal Testing workflow.

Figure 1: Causal Testing workflow.

Test Adequacy

Because the properties being tested are completely separate from the data used to validate them, traditional coverage-based metrics are not appropriate here. The CTF instead evaluates the adequacy of a particular dataset by calculating a statistical metric (Foster et al., 2024) based on the stability of the causal effect estimate, with numbers closer to zero representing more adequate data.

Missing Variables

Causal Testing works by using the causal DAG to identify the variables that need to be statistically controlled for to remove their biassing effect on the causal estimate. This typically means we need to know their values. However, where such biassing variables are not recorded in the data, the Causal Testing Framework can still sometimes estimate unbiased causal effects by using Instrumental Variables (Hernán & Robins, 2020), an advanced Causal Inference technique.

Feedback Over Time

Many scientific models involve iterating several interacting processes over time. These processes often feed into each other, and can create feedback cycles. Traditional CI cannot handle this, however the CTF uses a family of advanced CI techniques, called g-methods (Hernán & Robins, 2020), to enable the estimation of causal effects even when there are feedback cycles between variables.

Related Work

The Dagitty tool (Textor et al., 2017) is a browser-based environment for creating, editing, and analysing causal graphs. There is also an R package for local use, but Dagitty cannot be used to estimate causal effects. For this, doWhy (Blöbaum et al., 2024; Sharma & Kiciman, 2020) is a free, open source Python package, and cStruture is a paid low code CI platform. However, these packages are intended for general CI. Neither explicitly supports causal software testing, nor do they support temporal feedback loops.

Ongoing and Future Research

The CTF is the subject of several publications (Clark, Foster, Walkinshaw, et al., 2023; Clark, Foster, Prifling, et al., 2023; Foster et al., 2024; Somers et al., 2024). We are also in the

process of preparing scientific publications concerning how the CTF handles missing variables and feedback over time. Furthermore, we are working to develop a plug-in for the DAFNI platform to enable national-scale infrastructure models to be easily tested.

Acknowledgements

This work was supported by the EPSRC CITCoM grant EP/T030526/1.

References

Blöbaum, P., Götz, P., Budhathoki, K., Mastakouri, A. A., & Janzing, D. (2024). DoWhy-GCM: An extension of DoWhy for causal inference in graphical causal models. *Journal of Machine Learning Research*, *25*(147), 1–7.

Chen, T. Y., Cheung, S. C., & Yiu, S. M. (1998). *Metamorphic testing: A new approach for generating next test cases* (HKUST-CS98-01). The Hong Kong University of Science; Technology.

Clark, A. G., Foster, M., Prifling, B., Walkinshaw, N., Hierons, R. M., Schmidt, V., & Turner, R. D. (2023). Testing causality in scientific modelling software. *ACM Transactions on Software Engineering Methodology*, *33*(1). https://doi.org/10.1145/3607184

Clark, A. G., Foster, M., Walkinshaw, N., & Hierons, R. M. (2023). Metamorphic testing with causal graphs. *2023 IEEE Conference on Software Testing, Verification and Validation (ICST)*, 153–164. https://doi.org/10.1109/ICST57152.2023.00023

Foster, M., Wild, C., Hierons, R. M., & Walkinshaw, N. (2024). Causal test adequacy. *2024 IEEE Conference on Software Testing, Verification and Validation (ICST)*, 161–172. https://doi.org/10.1109/ICST60714.2024.00023

Guderlei, R., & Mayer, J. (2007). Statistical metamorphic testing testing programs with random output by means of statistical hypothesis tests and metamorphic testing. *Seventh International Conference on Quality Software (QSIC 2007)*, 404–409. https://doi.org/10.1109/QSIC.2007.4385527

Hernán, M. A., & Robins, J. M. (2020). *Causal Inference: What if*. Chapman & Hall/CRC.

Pearl, J. (2009). *Causality: Models, reasoning, and inference*. Cambridge university press. ISBN: 9780521895606

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

Somers, R., Walkinshaw, N., Hierons, R., Elliott, J., Iqbal, A., & Walkinshaw, E. (2024). *Configuration testing of an artificial pancreas system using a digital twin*. https://doi.org/10.2139/ssrn.4732706

Textor, J., Zander, B. van der, Gilthorpe, M. S., Liśkiewicz, M., & Ellison, G. T. H. (2017). Robust causal inference using directed acyclic graphs: The R package "dagitty." *International Journal of Epidemiology*, dyw341. https://doi.org/10.1093/ije/dyw341