Causal Discovery Toolbox:
Uncover causal relationships in Python

Diviyan Kalainathan   diviyan.kalainathan@inria.fr
TAU, LRI, INRIA, Université Paris-Sud
660 Rue Noetzlin, 91190 Gif-Sur-Yvette, France

Olivier Goudet   olivier.goudet@univ-angers.fr
LERIA, Université d'Angers,
2 boulevard Lavoisier, 49045 Angers, France

Abstract

This paper presents a new open source Python framework for causal discovery from observational data and domain background knowledge, aimed at causal graph and causal mechanism modeling. The CDT package implements the end-to-end approach, recovering the direct dependencies (the skeleton of the causal graph) and the causal relationships between variables. It includes algorithms from the 'Bnlearn' (Scutari, 2018) and 'Pcalg' (Kalisch et al., 2018) packages, together with algorithms for pairwise causal discovery such as ANM (Hoyer et al., 2009). CDT is available under the MIT License at
`https://github.com/Diviyan-Kalainathan/CausalDiscoveryToolbox`.

Keywords: Causal Discovery, Graph recovery, open source, constraint-based methods, score-based methods, pairwise causality, Markov blanket.

1. Introduction

Causal modeling is key to understand physical or artificial phenomenons and make recommendations. Most softwares for causal discovery have been developed in the R programming language (Kalisch et al., 2018; Scutari, 2018), and a few causal discovery algorithms are available in Python (RCC (Lopez-Paz et al., 2015), CGNN (Goudet et al., 2017) and SAM (Kalainathan et al., 2018)), which supports many current machine learning frameworks such as PyTorch (Paszke et al., 2017).

The proposed CDT package is concerned with observational causal discovery, aimed at learning both the causal graph and the associated causal mechanisms from samples of the joint probability distribution of the data. CDT is supported by PyTorch.

Formally, the **Causal Discovery Toolbox** (CDT) is a open-source Python package including many state-of-the-art causal modeling algorithms (most of which are imported from R), that supports GPU hardware acceleration and automatic hardware detection.

Compared to other causal discovery packages, CDT unifies pairwise and score-based multi-variate approaches within a single package, implementing an end-to-end, step-by-step pipeline approach (Fig. 1).
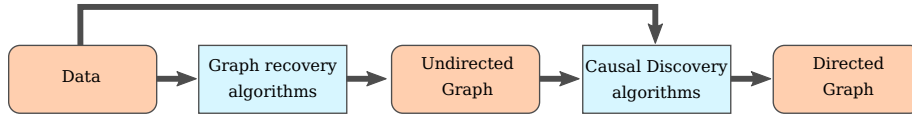

Figure 1: The CDT causal modeling package: General pipeline

CDT also provides an intuitive approach for including R based algorithms, facilitating the task of extending the toolkit with additional R packages. The package revolves around the usage of *networkx.Graph* classes, mainly for recovering (un)directed graphs from observational data.

2. Algorithms

CDT currently includes 17 algorithms for graph skeleton identification: 7 methods based on independence tests, and 10 methods aimed at directly recovering the skeleton graph. It furter includes 19 algorithms aimed at causal directed graph prediction, including 10 graphical and 9 pairwise approaches.

2.1 Recovering the graph skeleton

CDT includes two types of methods for recovering undirected dependence graphs from raw data: methods based on pairwise dependence statistics (also referred to as bivariate methods), and methods based on variable/feature selection.

Bivariate dependencies support variable/feature selection and are used to determine the (undirected) edges in the causal graph. They rely on statistical tests, e.g. Pearson's correlation or mutual information scores (Vinh et al., 2010). Bivariate dependencies are used in a first phase to establish the causal graph skeleton. In a further phase, heuristics aimed at builing a (causal) DAG from the causal graph skeleton are used. In particular, indirect edges (that is, edge $A \rightarrow C$, when edges $A \rightarrow B$ and $B \rightarrow C$ have been established) are removed using e.g., Network Deconvolution (Feizi et al., 2013). These graph pruning heuristics can be parameterized to control the sparsity of the graph.

Multivariate methods aims at recovering the full causal graph, that is, selecting parent, children and spouse nodes (parents of children) for all variables of the graph. This task has been thoroughly investigated through feature selection, graph heuristics (Friedman et al., 2008), and Markov blankets (Tsamardinos et al., 2003). All these methods output a `networkx.Graph` object.

2.2 Causal discovery

The main focus of the CDT package is causal discovery from observational data, ranging from the pairwise setting to the full graph modeling.

The pairwise setting considers a pair of variables and aims to determine the causal relationship between those variables. This setting implicitly assumes that both variables are already conditioned on other covariates, or readjusted with a propensity score (Rosenbaum and Rubin, 1983), and that remaining latent covariates have little or no influence and can be considered as "noise". The pairwise setting is also relevant to complete a partially directed graph resulting from other causal discovery methods. In the 2010s, the pairwise setting was investigated by Hoyer et al. (2009) among others, respectively proposing the Additive Noise Model (ANM). Later on, (Guyon, 2013) launched international challenges on Kaggle and Codalab on Cause-Effect pair (CEP) problems; CEP formulates bivariate causal identification as a machine learning task, where a classifier is trained from examples (A_i, B_i, ℓ_i), where variable pair (A_i, B_i) is represented by samples of their joint distribution and label ℓ_i indicates the type of causal relationship among both variables (independent, $A_i \rightarrow B_i$, $B_i \rightarrow A_i$). The CEP challenges spurred the development of state-of-the-art pairwise causal identification methods such as Jarfo, RCC (Fonollosa, 2016; Lopez-Paz et al., 2015).

The graph setting, extensively studied in the literature, involves Bayesian or score-based approaches. Bayesian approches rely either on conditional independence tests named **constraint-based methods**, such as PC or FCI (Spirtes et al., 2000; Strobl et al., 2017), or on **score-based methods**, finding the graph that maximizes a likelihood score through graph search heuristics, like GES or CAM (Chickering, 2002; Bühlmann et al., 2014). Other approaches leverage the celebrated Generative Adversarial Network setting **?**, such as CGNN or SAM (Goudet et al., 2017; Kalainathan et al., 2018). Graph setting methods output either a directed acyclic graph or a partially directed acyclic graph.

3. Implementation and utilities

R integration. As said, currently 10 algorithms are coded in R, and 17 in Python. The CDT package integrates all of them, using Wrapper functions in Python to enable the user to launch any R script and to control its arguments; this R script is executed in a temporary folder with a *subprocess* to avoid the limitations of the Python GIL. The results are retrieved through files back into the main Python process. The whole procedure is modular and allows contributors to easily add new R functions to the package.

Hardware configuration settings. At the package import, tests are realized to pinpoint the configuration of the user: Availability of GPUs and R packages and number of CPUs on the host machine. All settings are stored in a single object `cdt.SETTINGS`. For some algorithms, GPUs accelerations are available through the use of the PyTorch library.

Sustainability and deployment. In order for the package to be easily extended, integrating and encouraging the community contributions, special care was paid to the quality of tests. Specifically, a Continuous Integration tool (CIT), added to the git repository, allows one to sequentially execute tests. On new commits and pull request, the CIT automatically: i) Test all functionalities of the new version on the package using *pytest*[1] on toy datasets; ii) Build docker images and push them to `hub.docker.com` ; iii) Push the new version on *pypi*; iv) Update the documentation website. This procedure also allows to test the proper functioning of the package with its dependencies.

4. Conclusion and future developments

The Causal Discovery Toolbox (CDT) package (CDT) allows Python users to apply many causal discovery or graph modeling algorithms on observational data. It is already used by (Goudet et al., 2017; Kalainathan et al., 2018). As the output graphs are `networkx.Graph` classes, these are easily exportable into various formats for visualization such as Graphviz and Gephi.

The package promotes an end-to-end, step-by-step approach: the undirected graph (bivariate dependencies) is first identified, before applying causal discovery algorithms; the latter are constrained from the undirected graph, with significant computational gains.

Future extensions of the package include: i) developing GPU-compliant implementation of new algorithms; ii) handling interventional data and time-series data (e.g. for neuroimaging and weather forecast); iii) evaluating the direct and total effect in a graph, given a cause variable and a target variable. Finally, we plan to develop facilities to test whether common assumptions (e.g. causal sufficiency assumption) hold and reduce the risk of applying methods out of their intended scope.

References

Peter Bühlmann, Jonas Peters, Jan Ernest, et al. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 2014.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Soheil Feizi, Daniel Marbach, Muriel Médard, and Manolis Kellis. Network deconvolution as a general method to distinguish direct dependencies in networks. *Nature biotechnology*, 31(8):726, 2013.

José AR Fonollosa. Conditional distribution variability measures for causality detection. 2016.

---

1. Holger Krekel et al., `https://github.com/pytest-dev/pytest/`

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Olivier Goudet, Diviyan Kalainathan, et al. Learning functional causal models with generative neural networks. 2017.

Isabelle Guyon. Chalearn cause effect pairs challenge, 2013. URL `http://www.causality.inf.ethz.ch/cause-effect.php`.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Neural Information Processing Systems (NIPS)*, pages 689–696, 2009.

Diviyan Kalainathan, Olivier Goudet, et al. Sam: Structural agnostic model, causal discovery and penalized adversarial learning. 2018.

Markus Kalisch, Alain Hauser, et al. Package 'pcalg'. 2018.

David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Ilya O Tolstikhin. Towards a learning theory of cause-effect inference. In *ICML*, pages 1452–1461, 2015.

Adam Paszke, Sam Gross, Soumith Chintala, et al. Automatic differentiation in pytorch. 2017.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Marco Scutari. Package 'bnlearn', 2018.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Eric V Strobl, Kun Zhang, and Shyam Visweswaran. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. 2017.

Ioannis Tsamardinos, Constantin F Aliferis, and Alexander R Statnikov. Algorithms for large scale markov blanket discovery. 2003.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.