Python package for causal discovery based on LiNGAM

Takashi Ikeuchi
IKEUCHI@SCREEN.CO.JP
Mayumi Ide
IDE@SCREEN.CO.JP
SCREEN Advanced System Solutions Co., Ltd., Japan
Yan Zeng
YANAZENG013@GMAIL.COM
Department of Computer Science and Technology, Tsinghua University, China
Takashi Nicholas Maeda
TN.MAEDA@MAIL.DENDAI.AC.JP
School of System Design and Technology, Tokyo Denki University, Japan
Center for Advanced Intelligence Project, RIKEN, Japan
Shohei Shimizu
SHOHEI-SHIMIZU@BIWAKO.SHIGA-U.AC.JP
Faculty of Data Science, Shiga University, Japan
Center for Advanced Intelligence Project, RIKEN, Japan

Editor: Andreas Mueller

Abstract
Causal discovery is a methodology for learning causal graphs from data, and LiNGAM is a well-known model for causal discovery. This paper describes an open-source Python package for causal discovery based on LiNGAM. The package implements various LiNGAM methods under different settings like time series cases, multiple-group cases, mixed data cases, and hidden common cause cases, in addition to evaluation of statistical reliability and model assumptions. The source code is freely available under the MIT license at https://github.com/cdt15/lingam

Keywords: Causal structure learning, statistical reliability, model evaluation

1. Introduction
Statistical causal inference learns causal quantities from data (Imbens and Rubin, 2015; Pearl, 2000). A common procedure for this is as follows: Users first specify the causal quantity to be estimated, e.g., the intervention effect of a variable on another variable. Second, they draw the causal graph based on background knowledge. Then, they derive variables (if any) that should be used to identify the quantity of interest based on graphical criteria, such as back-door and front-door criterion, and their generalizations (Pearl, 1995; Shpitser and Pearl, 2008; Bhattacharya et al., 2020; Jung et al., 2020).

A fundamental step of the aforementioned procedure is to draw the causal graph based on background knowledge. However, it is often the case that background knowledge is not enough to draw the causal graph. Causal discovery (Spirtes et al., 1993; Pearl, 2019) is a methodology for inferring causal graphs in data-driven ways; it aims to help users draw causal graphs by combining data with prior knowledge.

A classic approach for causal discovery is to use conditional independence of variables for inferring the underlying causal graph (Spirtes et al., 1993; Pearl, 2000). This approach, in
principle, does not make specific assumptions on the functional forms of the causal relations of variables or distributions of variables; it only infers a set of equivalent models and is not able to estimate causal directions for most cases.

In contrast, a recent approach (Shimizu, 2014; Zhang and Hyvärinen, 2016; Shimizu, 2022) makes some assumptions on the functional forms or/and distributions of variables to address this limitation. The linear non-Gaussian acyclic model (Shimizu et al., 2006), abbreviated as LiNGAM, is the most well-known example, where the error variables assumedly follow non-Gaussian continuous distributions, but at most one error variable may be Gaussian. The assumption of non-Gaussian errors enables examining the independence of error variables, unlike that of Gaussian errors. This LiNGAM approach achieves better identification results and is capable of uniquely estimating causal directions in much more cases than the classic approach based on conditional independence. This feature of identifiability has attracted much attention of the research community (Drton and Maathuis, 2017; Glymour et al., 2019; Peters et al., 2017; Shimizu, 2014, 2022) and has led to numerous applications of the methodology, for example, in epidemiology (Rosenström et al., 2012), economics (Moneta et al., 2013), neuroscience (Mills-Finnerty et al., 2014), and materials science (Campomanes et al., 2014; Liu et al., 2021). See https://www.shimizulab.org/lingam/lingampapers for a list of papers on the methodology and its applications.

Representative causal discovery packages are TETRAD (Scheines et al., 1998; Ramsey et al., 2020), pcalg (Kalisch et al., 2012) and bnlearn (Scutari and Denis, 2021). These packages are rich in classic methods based on conditional independence including constraint-based methods such as PC and FCI (Spirtes and Glymour, 1991; Spirtes et al., 1995) and greedy score-based methods such as GES (Chickering, 2002) and NOTEARS (Zheng et al., 2018), whereas TETRAD and pcalg only provide a basic method for the LiNGAM approach (Shimizu et al., 2006) based on independent component analysis (ICA) (Hyvärinen et al., 2001). Causal Discovery Toolbox (Kalainathan et al., 2020) gives a Python front end to perform methods of pcalg and bnlearn written in R, but only offers the basic ICA-based method for LiNGAM (Shimizu et al., 2006). Further, their implementation of a nonlinear causal discovery method based on a similar idea of LiNGAM (Hoyer et al., 2009) is limited to two variable cases. Tigramite (https://github.com/jakobrunge/tigramite) offers time series causal discovery methods based on conditional independence (Gerhardus and Runge, 2020), but does not exploit additional information on the functional forms of the causal relations of variables or distributions of variables, unlike LiNGAM-type methods.

Thus, in this paper, we present a Python package for performing various LiNGAM-type methods including time series cases (Hyvärinen et al., 2010; Kawahara et al., 2011), multiple-group cases (Shimizu, 2012; Kadowaki et al., 2013), mixed data cases (Zeng et al., 2022), hidden common cause cases (Maeda and Shimizu, 2020; Zeng et al., 2021), and (multivariate) nonlinear cases (Peters et al., 2014). The package covers most of the major LiNGAM-type methods already used in application papers and relevant extensions. Users can choose suitable methods depending on what they assume based on their background knowledge. We plan to extend it further and encourage others to join the development. Moreover, the package offers additional functionalities including evaluation of statistical reliability based on bootstrapping (Komatsu et al., 2010) and model evaluation based on the magnitude of error independence (Entner and Hoyer, 2011; Tashiro et al., 2014).
2. Available models and estimation algorithms

This section gives a brief description of each of the LiNGAM methods available in this package. See the online documentation (https://lingam.readthedocs.io/en/latest/) for more details.

The most basic LiNGAM model (Shimizu et al., 2006) assumes that their causal relations are acyclic, with no hidden common causes, linearity, and non-Gaussian errors. The basic LiNGAM model can be estimated in different ways. This package implements two major algorithms: the original LiNGAM discovery algorithm based on ICA (Shimizu et al., 2006) and a direct method called DirectLiNGAM (Shimizu et al., 2011). The package further offers utilities to compute total effects between observed variables and their direct effects, drawing causal graphs using Graphviz, computing bootstrap probabilities of directed paths and edges, and incorporating prior knowledge on topological causal orders in the estimation by DirectLiNGAM. It further enables model evaluation by examining the independence of errors. Two extensions of the basic LiNGAM models are available in our package for multigroup analysis. First, Shimizu (2012) jointly estimates multiple LiNGAMs using multiple datasets from multiple sources by constraining their topological causal orders to be identical. This would enable a more accurate estimation of the LiNGAMs than estimating them separately, given the prior knowledge that they share a topological causal order. Second, Kadowaki et al. (2013) consider performing causal discovery on paired samples and propose an estimation method for learning causal structures in longitudinal data that collects samples over time. Their algorithm can analyze causal structures, including topological causal orders, that may change over time.

We compared the accuracy and runtime of our implementation of the ICA-based LiNGAM algorithm with those of an existing package, pcalg, for different numbers of variables. We also tested our implementation of DirectLiNGAM for comparison. The python code used to generate artificial data in our experiments is available at https://github.com/cdt15/lingam/blob/master/examples/data/GenerateDatasets.ipynb Fig. 1 shows that our implementation of DirectLiNGAM was more accurate than our and pcalg implementations of ICA-based LiNGAM. Our implementation of ICA-based LiNGAM was faster than its
pcalg version, whereas DirectLiNGAM was slower than our and pcalg implementations of ICA-based LiNGAM.

The package further offers two time series extensions of the basic LiNGAM: VAR-LiNGAM (Hyvärinen et al., 2010) combines it with vector autoregressive models (VAR), and VARMA-LiNGAM (Kawahara et al., 2011) does the same with vector autoregressive moving average models (VARMA). The package also provides a mixed data extension of the basic LiNGAM: Linear Mixed (LiM) causal discovery algorithm extends LiNGAM to handle the mixed data that consists of both continuous and discrete variables (Zeng et al., 2022). Further, our package can perform a nonlinear causal discovery RESIT (Peters et al., 2014) assuming a nonlinear additive noise model with acyclicity and no hidden common causes (Hoyer et al., 2009). Users can use a nonlinear regression from those implemented in scikit-learn (Pedregosa et al., 2011).

Another important extension is LiNGAM with hidden common causes or latent factors (Hoyer et al., 2008; Zeng et al., 2021). We implemented the RCD algorithm (Maeda and Shimizu, 2020) and CAM-UV algorithm (Maeda and Shimizu, 2021). The RCD algorithm allows the existence of hidden common causes and outputs a causal graph, where a bidirected arc indicates the pair of variables that have the same hidden common causes and a directed arrow indicates the causal direction of a pair of observed variables that are not affected by the same hidden common causes. CAM-UV is its nonlinear variant and assumes the structural equations additive in the observed variables and errors. We also implemented the Multi-Domain LiNGAM algorithm for latent factors (MD-LiNA) (Zeng et al., 2021). Given the observed measurement data, MD-LiNA allows to locate the latent factors in addition to uncovering the causal structure between such latent factors of interests.

3. Design, API and future development

To facilitate application by machine learning users, we designed the model with the fit function, similar to scikit-learn. The standard flow is to call the fit method shown below to build the model after the model instance is created.

```python
model = lingam.DirectLiNGAM()
model.fit(X)
```

After model building, the graph is returned as an adjacency matrix.

We plan to continue further development of the package. We also encourage others to join the project. It is easy to extend the model by following the Contribution Guide in the online documentation and further referring to other models. The Contribution Guide includes the code style and the ways to check the format, write the documentation, perform unit tests, and create a pull request. The package would benefit applied researchers and practitioners in enjoying the results of recent developments in causal discovery and deriving better causal conclusions based on domain knowledge and data.

Acknowledgments

We thank support from JSPS Grant-in-Aid for Scientific Research (C) #20K11708, ONR N00014-20-1-2501, and China Postdoctoral Science Foundation (2022M711812).
References


Python package for causal discovery based on LiNGAM


