Finding the causes of failure: Neuralwave 2024

Mark Sobolev
Faculty of Informatics
Università della Svizzera Italiana
Lugano, Switzerland

Marco Gabriel
Faculty of Informatics
Università della Svizzera Italiana
Lugano, Switzerland

Laurynas Varnas
Faculty of Informatics
Università della Svizzera Italiana
Lugano, Switzerland

Jonatan Bella
Faculty of Informatics
Università della Svizzera Italiana
Lugano, Switzerland

Yassine Oueslati
Faculty of Informatics
Università della Svizzera Italiana
Lugano, Switzerland

Abstract

We proposed a solution to a causal discovery in a multiple dataset context setting with a prior structural knowledge. Given the different measurements set, we produced a directed graph of causal influences and a ranking of influencing variables. In addition, we implemented a way to precisely constraint our causal model to reflect the changed mode. We developed a complex evaluation metric set for the unsupervised settings. Finally we designed a way to combine different approaches to causal discovery to produce the most reliable approach to generalised causal discovery setting.

1 Objective

We've been tackling a problem, where we were given several datasets with different latent context, for those we needed to provide the directed acyclic graph, which could be interpreted casually. Namely the datasets were structured as a set of columns, which were grouped in a several blocks, with an assumption of locality between elements of a block. We needed to discover various causal related structures about the given data:

1. The underlying causal graph
2. The root causes of the observed change
3. Visualisation of the graph, the root causes and their influence

1.1 Causal discovery

Causal discovery is a classic task, which also is extremely challenging. The causality discovery is used mainly in biology (Sachs, 2005) The task is to infer a Directed Acyclic Graph which minimizes a specific metric (which is trying to model causality) by observing the given data. The classic algorithms included using constrained-based approach to construct a graph (Glymour,2001: 5.4.2); the more novel ones usually vary in their assumptions (e.g. linearity and non-Gaussianity for DirectLiNGAM (Shimizu et al 2011)), but propose also more flexible ways to learn and evaluate the results (Zheng et al 2018). The complexity of the causal discovery relates to several facts: searching over the space of DAG is known to be NP-hard (Chickering 1996).

A recent major breakthrough in the field can be characterised by the introduction of a reformulation of a task to a continuous optimisation problem (Zheng, 2018); this introduced a family of new gradient-optimisational algorithms (Wang et al, 2021; Yu et al. 2019), however some skepticism is also present (Kaiser, 2021).

1.2 Evaluation of causal graph in unknown setting

As the task in a general setting should be solved without any ground truth given (except structural prior knowledge), one should come up with an algorithm for this. The task is not well researched, however there are some general strategies for doing such. First, you can divide your data to several sets, and try to evaluate your DAGs on the test data using Markov blanket (Pearl, 1986) , following (Wang, 2025). Second, a set of formal criterions for the graph evaluation can be suggested, such as generalizability (Yang, 2019), and graph falsifying metrics (Eulig, 2023). Both ways were computationally infeasible for us, so we've developed 4 intermediate strategies for the evaluation:

1.2.1 Peer dataset

The first way to evaluate the algorithms was by providing a helper dataset (i.e. the most popular nowadays IHDP (Shalit, 2017)), and evaluating datasets on that. As usually the causal discovery is done by using synthetic datasets (Cheng, 2022), it should be a good estimate for an implementation.

1.2.2 Distribution change evaluation

The second way was done by incorporating the fact that we had two different distributions for two datasets, thus the algorithm was to train the model on the first, prior dataset, and try to minimize the drop of causal estimation (with a helper regression algorithm e.g. XGBoost (Chen & Guestrin, 2016)).

1.2.3 Likelihood of the graph by the inner metrics

The third way is the comparison of the inner statistical metrics of different algorithms, e.g. p-values of the DirectLinGam [2]. By this way the algorithms themselves report the quality of the output.

1.2.4 Predicting the data with the causal graph

This approach is based on training the regression and just comparing the output by the usual regression metrics, namely MSE or MAPE.

2 Approach

2.1 Methodology

The process is divided into three stages. First, we do not have access to any ground-truth. In the second stage, we get access to the first 15 rows of the target adjacency-matrix. In the last stage, we get additional access to the lower diagonal adjacency-matrix. We use this ground-truth as prior knowledge in the following algorithms as well as to evaluate our results. To evaluate the performance, we calculate the Structural Hamming Distance (SHD) between the resulting adjacency matrix and the full ground truth.

In every stage, we fit various algorithms to the given data and compared their results. We outline these algorithms and the used frameworks in the following sections.

2.2 Algorithms

2.2.1 Causal discovery with Ordering-based Reinforcement Learning (CORL)

CORL[10] uses a Reinforcement Learning (RL) approach and is implemented in the gCastle Framework.

2.3 DAG_GNN

In DAG-GNN [11], the problem is formulated as a continuous optimization problem and solved using a generative model. It is impllemented in gCastle.

2.3.1 LiNGAM

LiNGAM estimates structural equation models or linear causal Bayesian networks, assuming the data is non-Gaussian. First, to apply DirectLiNGAM[2], we merge the datasets while introducing a new feature to differentiate the origins. Secondly, we use MultiGroupDirectLiNGAM: [6] on the separate datasets. Additionally, we try Independent component analysis (ICA-LiNGAM)[7] which assumes the observed variables to be linear functions of mutually independent and non-Gaussian disturbance variables.

2.3.2 Fast Causal Inference (FCI)

The FCI algorithm [9] works under both latent variables and selection bias. We use its implementation in causal-learn [13].

2.3.3 VAR-LINGRAM

Var-Lingram (Shimizu, 2006) is a time series extension for a Lingram model. The linearity of a data-generating process was an important contribution, due to the similarity with our structural constraints.

2.3.4 Peter-Clark (PC)

PC [8] is a general purpose algorithms for causal discovery which works by first identifying the undirected causal graph, and then directing the edges. It is also implemented in causal-learn.

2.3.5 Structural Agnostic Model (SAM)

We use the SAM[4] implementation of the CausalDiscoveryToolbox (CDT). SAM implements an adversarial game in which separate models generate each variable based on the real observations from all other variables, on the hypothesis, that the target variable depends on those. A discriminator then attempts to distinguish between real and generated samples. Finally, a sparsity penalty forces each generator to consider only a small subset of the variables, yielding a sparse causal graph. SAM allows to alleviate the commonly made condition, that there is no hidden common cause to any pair of observed variables. It is shown to be orders of magnitudes slower than LiNGAM for example, but also significantly better.

2.3.6 LLM: o1-preview from OpenAI

We prompt the o1-preview LLM from OpenAI to respond with the nodes causing the increased value, given the datasets.

2.4 Frameworks

2.4.1 DoWhy

The key feature of DoWhy [1] [5] is that it provides algorithms for the refutation and falsification of causal graphs.

Table 1: Comparison of different Algorithms

	Part
Name	SHD Distance to ground truth (lower is better)
CORL	493
DAG_GNN	543
PC	523
FCI	523

2.4.2 CausalDiscoveryToolbox (CDT)

The CausalDiscoveryToolbox supports causal inference in graphs and in pairwise settings. It implements several algorithms, including the PC, LiNGAM, and SAM algorithms. https://github.com/FenTechSolutions/CausalDiscoveryToolbox [3]

2.4.3 gCastle

GCastle is a causal structure learning toolchain containing various functionalities related to causal learning such as the PC and LiNGAM algorithms. [12]

3 Results

3.1 Causal Graph

Comparing the various algorithms to discover causal graphs, we found LiNGAM to produce good quality graphs while being significantly faster than many others. DUe to these findings, we chose it for our final implementation. An excerpt of the comparisons we did can be seen in Table 1.

The final resulting graph is shown in Fig 1.

3.2 Root Cause

3.2.1 graph pipeline

This approach uses causal inference to uncover and visualize the main drivers affecting a target variable. We model the causal relationships between variables as a directed graph, where each node represents a variable, and edges indicate directional influence. Using algorithms that calculate causal impact, we identify the path with the highest total effect leading to the target. This allows for pinpointing critical variables influencing outcomes.

3.2.2 Example

In this example, the target variable is Station5_mp_85, shown in red. The most influential path (highlighted in bold red) passes through two intermediate variables, Station2_mp_28 and Station2_mp_13, with a total effect of 1.288. Here, Station2_mp_28 acts as a primary influencing node, connecting other variables to the target. This analysis helps focus on key variables that drive the outcome, aiding in targeted intervention.

3.2.3 risk evaluation method

The methodology involves utilizing the LiNGAM framework to estimate causal relationships between measurement points in high and low scrap datasets from a manufacturing process, incorporating a prior knowledge matrix to enhance interpretability, and employing backtracking to identify influential paths leading to specific target variables; the expected result is a causal graph that reveals key influences on scrap rates, providing actionable insights for quality improvement interventions. This is computationally infeasible.

The sub-graph showing the root cause is visualised in Fig. 2.

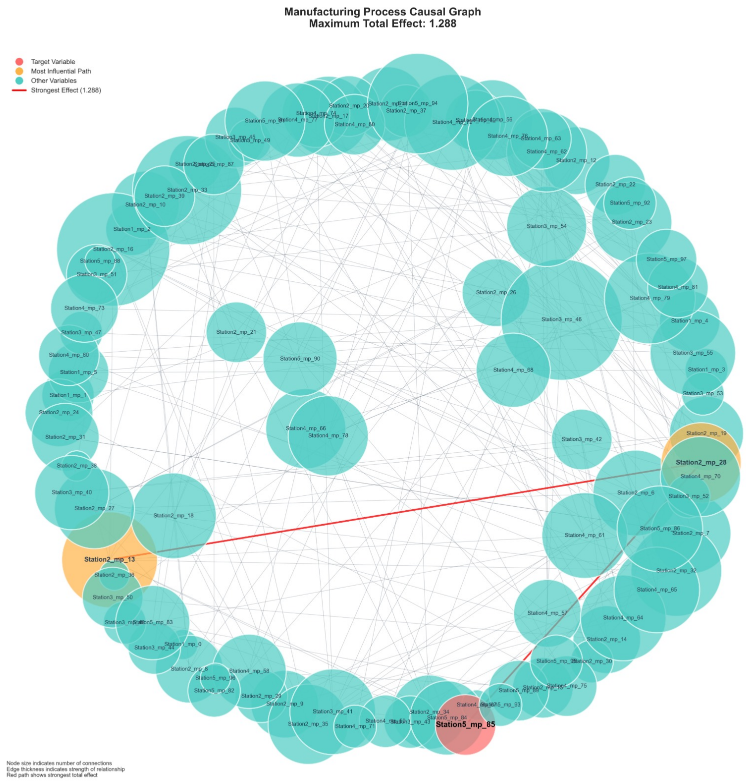

Figure 1: The entire causal graph

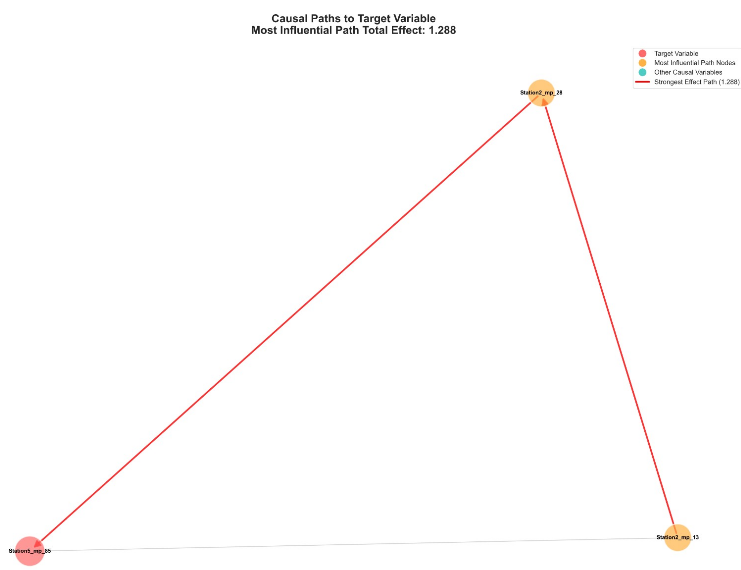

Figure 2: The sub-graph showing the root cause, from Station2_mp_13 to Station2_mp_28 to Station5_mp_85.

3.3 Visualisation

To visualize the root cause, we mainly show all nodes that feed into the target node. This is achieved by inverting the found adjacency matrix and starting from the target node, visit all its descendants recursively. Thus, we get the sub-graph of all nodes influencing the target node.

Further, we identify the most influential path according to its total effect on the target node, as seen in Fig. 2.

4 Challenges

The most challenging issue throughout the process was to evaluate possible solutions without knowing the ground truth or just having partial ground truth.

5 Video

A video demonstration can be seen here: `https://www.loom.com/share/515b376a54ef4ea9b03461e467991c65?sid=a8092050-37d9-430e-bfaf-90c8cbf3ed57`

References

[1] Patrick Blöbaum et al. "DoWhy-GCM: An Extension of DoWhy for Causal Inference in Graphical Causal Models". In: *Journal of Machine Learning Research* 25.147 (2024), pp. 1–7. URL: `http://jmlr.org/papers/v25/22-1258.html`.

[2] Takashi Ikeuchi et al. "Python package for causal discovery based on LiNGAM". In: *Journal of Machine Learning Research* 24.14 (2023), pp. 1–8. URL: `http://jmlr.org/papers/v24/21-0321.html`.

[3] Diviyan Kalainathan and Olivier Goudet. *Causal Discovery Toolbox: Uncover causal relationships in Python*. 2019. arXiv: `1903.02278 [stat.CO]`. URL: `https://arxiv.org/abs/1903.02278`.

[4] Diviyan Kalainathan et al. *Structural Agnostic Modeling: Adversarial Learning of Causal Graphs*. 2022. arXiv: `1803.04929 [stat.ML]`. URL: `https://arxiv.org/abs/1803.04929`.

[5] Amit Sharma and Emre Kiciman. *DoWhy: An End-to-End Library for Causal Inference*. 2020. arXiv: `2011.04216 [stat.ME]`. URL: `https://arxiv.org/abs/2011.04216`.

[6] Shohei Shimizu. "Joint estimation of linear non-Gaussian acyclic models". In: *Neurocomputing* 81 (2012), pp. 104–107. ISSN: 0925-2312. DOI: `https://doi.org/10.1016/j.neucom.2011.11.005`. URL: `https://www.sciencedirect.com/science/article/pii/S0925231211006813`.

[7] Shohei Shimizu et al. "A Linear Non-Gaussian Acyclic Model for Causal Discovery". In: *Journal of Machine Learning Research* 7.72 (2006), pp. 2003–2030. URL: `http://jmlr.org/papers/v7/shimizu06a.html`.

[8] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Vol. 81. Jan. 1993. ISBN: 978-1-4612-7650-0. DOI: `10.1007/978-1-4612-2748-9`.

[9] Peter Spirtes, Christopher Meek, and Thomas S. Richardson. "Causal Inference in the Presence of Latent Variables and Selection Bias". In: *CoRR* abs/1302.4983 (2013). arXiv: `1302.4983`. URL: `http://arxiv.org/abs/1302.4983`.

[10] Xiaoqiang Wang et al. *Ordering-Based Causal Discovery with Reinforcement Learning*. 2021. arXiv: `2105.06631 [cs.LG]`. URL: `https://arxiv.org/abs/2105.06631`.

[11] Yue Yu et al. *DAG-GNN: DAG Structure Learning with Graph Neural Networks*. 2019. arXiv: `1904.10098 [cs.LG]`. URL: `https://arxiv.org/abs/1904.10098`.

[12] Keli Zhang et al. *gCastle: A Python Toolbox for Causal Discovery*. 2021. arXiv: `2111.15155 [cs.LG]`. URL: `https://arxiv.org/abs/2111.15155`.

[13] Yujia Zheng et al. "Causal-learn: Causal discovery in python". In: *Journal of Machine Learning Research* 25.60 (2024), pp. 1–8.