Constraint-based Causal Discovery by using Path Constraints gained from Signal Injection and Recovery

Josephine Rehak

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
Josephine.Rehak@kit.edu
ORCiD: 0000-0001-6139-9703

Abstract

The discovery of causal relations via interventions has proven to be simple when only one observed variable is affected or unaffected. However, in a multivariate setting, it is likely that more than one variable is affected by the intervention. Thus drawing conclusions about the true causal graph becomes far more difficult as we can not retrieve information of any obvious causal relationship or causal order. We demonstrate, that causal discovery with multiple affected variables is possible by introducing a novel definition of path constraints for constraint-based causal discovery. We exercise our novel technique on a combustion engine simulation, were we inject wavelets of our choice in a variable of investigation and try to rediscover this wavelet in the other, observed variables to gain such path constraints and thus to restraint the causal graph search.

1 Introduction

Causal discovery deals with finding cause and effect relationships in data. Its subdomain of interventional causal discovery tries to achieve this knowledge gain by performing experiments [5].

Current methods for interventional causal discovery either inspect the interventional effect in upmost one variable and thus recover few causal information or cannot handle interventional effects in multiple variables. We want to investigate a specific approach that is low-intrusive and injects a signal into a running process and gathers information about its occurrence in the causal graph to deduce information about causal relations. We will demonstrate an example experiment on a combustion engine simulation. The paper is structured as follows: In Section **??**, we shed some light on existing approaches for interventional structure learning. In Section 3, we introduce our fundamentals for the novel low-invasive technique. In Section 4, the signal injections are demonstrated on a combustion engine example step by step. In Section 4, we draw the conclusion.

2 Causal Graphs

Causal graphs consist of a set of nodes representing variables V and a set of edges E representing causal relations. If a directed edge points from $A \in V$ to $B \in V$, then variable B is caused by A. A path from A to B is a chain of consistently directed edges $C(A, B) = \{A \rightarrow X_1, \ldots, X_i \rightarrow B\}$, $X_i \in V$, $i \in \mathbb{N}$ directed from A to B with a number of edges being equal or greater than one $|C(A, B)| \geq 1$. A direct causal relation between variables indicates $|C(A, B)| = 1$, but an indirect causal relation indicates a path $|C(A, B)| > 1$.

The goal of causal discovery is to gain knowledge over the true causal graph for the inspected environment. According to the theory of constraint-based learning [6], all the potential graphs form an equivalence class in respect to our knowledge about the edges and variables of the graph itself. If the number of graphs in the equivalence class equals one, we assume to have found the true causal graph. But if no knowledge of the causal graph is given and the larger the amount of inspected variables, the more edges have to be inspected. Table 2.1

shows, how the number of graphs in an equivalence class grows exponentially with the number of variables and edges. One can gain these constrains, by either inspecting data, or as in our case performing experiments.

2.0.1 Interventional Causal Discovery

Using interventions for causal discovery is one of the oldest and most popular approaches in science. Even despite their costliness, their potential of being unethical or by being simply not feasible. It is assumed that a variable is a cause of another variable B if an intervention on A also affects the associated variable B [8]. [2] distributed the existing approaches in two major categories called structural interventions and parametric interventions.

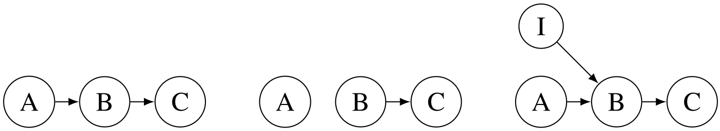
2.0.2 Structural Interventions

As shown in Figure 2.1, structural interventions (also called hard interventions) cut off all causal influences to the variable under intervention and determines its probability distribution completely. For example in randomized controlled drug trails [3], the treatment drug a patient receives is determined randomly but always one of several options. Here, as stated by the potential outcomes framework, the causal effect is identified by structurally intervening on the one variable while observing the effect in the other.

Parametric Interventions Parametric interventions (also called soft intervention) intervene on the probability distribution of a variable by adding another cause to it or its causes. They do not disturb the original causal structure, but

Count of Variables	1	2	3	4	5	6
Count of Potential Edges	0	1	3	6	10	15
Count of Potential Graphs	0	4	64	4,096	1,048,576	1,073,741,824

Table 2.1: Overview of the exponential increase in graphs with growing number of edges and variables



(a) Original Graph (b) Structural Intervention (c) Parametric Intervention

Figure 2.1: Causal graphs after a structural and a parametric intervention on variable B

no foreign influences of other variables can be prevented with certainty. In comparison, parametric interventions are a rather new technique.

3 Essentials of Wavelet Injections

The new intervention method can be classified according to Section **??** as soft intervention-based, since the original causal network is not perturbed but an additional variable is added as a cause to the inspected variable. The particularity of these interventions lies in the fact that a wavelet in the form of a wavelet is added to the variable and is tried to be rediscovered to gain causal information.

When injecting a wavelet into a variable A, the injected wavelet and the timeseries coming from the causing variable are added up. We assume the wavelet to spread in the graph in direction of the causal relations. If we find the wavelet in one variable, we assume a direct causal relation to be present, but in case of a discovery in several other variables, including B, we may not. Instead, we gain knowledge about an existing path between the variables with $|C(A, B)| \geq 1$, since the wavelet must have traveled somehow from A to B. We will refer to such path information as path constraints from here on. With each gained path constraint, we can remove all graphs from the equivalence class of potential graphs that have no path between the investigated variables present and hence do not support it and thus the number of potential graphs is decreased.

For signal recovery, we normalized the measured values. Otherwise, the different scales of the variables would make a comparison difficult. Then we applied on each measured variable the fast pattern matching algorithm called Mueens ultra-fast Algorithm for Similarity Search (MASS) [9]. It stepwise matches a desired pattern to a subsequence of the inspected timeseries and calculates the z-normalized distance. The aggregation of these distances results in an overall distance profile. If its minimal distance is below a chosen threshold, we assume the position to be our signal. Otherwise, we assume the signal to be absent in the observed variable and thus we gain no path constraint.

Note, that in general, we do not consider information about variables in which the injected signal could not be found, as the wavelet may be lost due to various reasons. For example, the signal may be of unfortunate form and hence be canceled out by the causal graph itself, it may be heavily deformed and thus be not recoverable or simply be too weak to be noticeable in other variables.

4 Applying Signal Injections

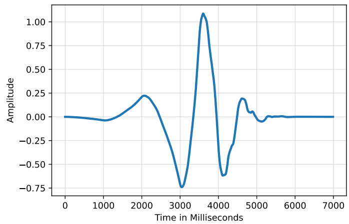
Here we demonstrate how we applied the signal injections on a combustion engine dataset and explain the experiment step by step.

Step 1: Wavelets for Injection

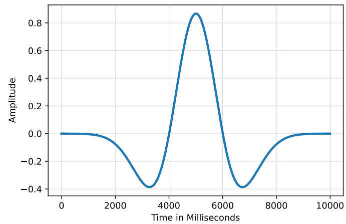
We decided to use three very distinct and well-defined wavelets for our signal injection in the combustion engine. These are a Daubechie 4 wavelet, a Mexican Hat wavelet and a Haar wavelet. They are depicted in Figure 4.1. We have chosen these wavelets because they contain amplitudes in the positive and negative value range and have a unique shape.

Step 2: Simulation Setup

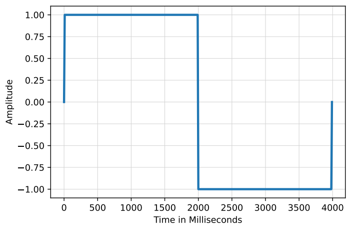
As a testing environment, we used a running combustion engine simulation [7, 4, 1]. To evaluate the performance of the novel discovery approach, we inferred the simulation's true causal graph as is shown in Figure 4.2. Here, we give a brief explanation of the causal relations: The *angle* of the throttle plate influences



(a) Daubechie 4 wavelet



(b) Mexican hat wavelet



(c) Haar wavelet

Figure 4.1: The wavelets we used for signal injection

how much *air intake* in the motor cylinder is possible. The *air intake* over time adds up to the *aircharge* in the cylinder before combustion. After combustion, depending on the *aircharge*, the *torque* increases and finally the overall *engine speed* rises. The increase in *engine speed* also depends on the *load* carried by the engine.

We added multiple sensors to the simulation to retrieve the timeseries required for signal recovery. In total we took measurements of six variables including the actuated variable. We injected the signals in the actuation of the angle only and inspected all the other measured timeseries for traces of the injected signal. According to Figure 4.2, we expect to find the wavelets in the air intake, aircharge, torque and engine speed variable, but not in the load variable, as it is

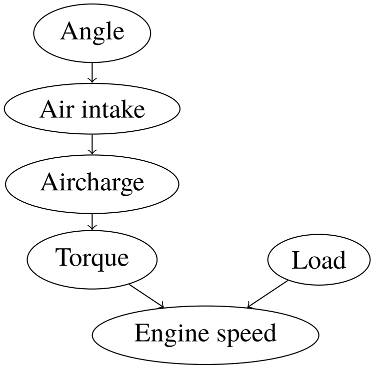


Figure 4.2: The true causal graph of the combustion engine simulation

independent of the angle variable. The wavelet rediscovery method are required to come to the same conclusion.

Step 3: Signal Discovery

As an implementation of the pattern matching algorithm, we used the python package stumpy [1]. With it, we found several wavelets in all variables depending on the influenced angle variable. Table 4.1 gives an overview of the wavelets we rediscovered. All in all, the Daubechie 4 wavelet and the Mexican hat wavelet performed best, as they were found in all variables depending on the angle variable in their actual positions. For evaluation purposes, we determined the actual position in advance, by comparing the measurements with wavelet injections with measurements without injections. Any divergence between those measurements must be caused by the wavelet. We decided to use this information only for evaluation, as we want to gain causal information with minimum number of measurements.

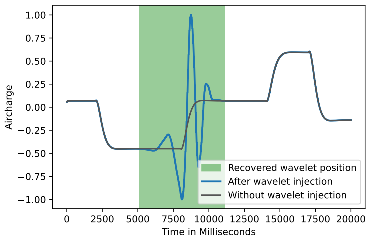
[1] `https://stumpy.readthedocs.io/`

	Air Intake	Aircharge	Torque	Engine Speed	Load
Daubechie 4 wavelet	✓	✓	✓	✓	×
Mexican hat wavelet	✓	✓	✓	✓	×
Haar wavelet	✓	×	×	×	×
Ground Truth	✓	✓	✓	✓	×

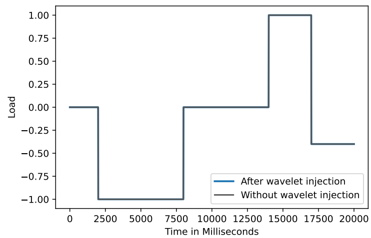
Table 4.1: An overview of the wavelets we injected into the angle variable and if they could be recovered in the other variables in their actual position.

Figure 4.3 presents an excerpt from our results for the aircharge and the load variable for each of the three wavelets. The colored area is where the signal was rediscovered by the pattern matching algorithm. It is colored green, when the wavelet is found in its actual position, but if it is red, it was found in a wrong position or in a variable, where no wavelet influence is present. In the aircharge variable, both the Daubechie 4 wavelet and the Mexican hat wavelet were rediscovered in their actual position. Only the Haar wavelet was found in the wrong position (Subfigure 4.3(e)).

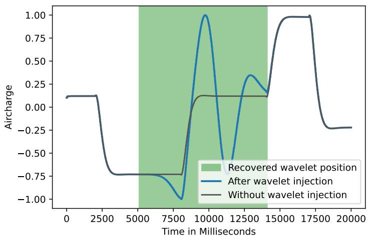
In the Load variable, no signal should be found at all, as the variable is not influenced by the angle variable where we injected the wavelet. But the discovery algorithm wrongly found the Haar wavelet in the Load variable (Subfigure 4.3(f)). We assume the mistaken discoveries of the Haar wavelet to be because of the simple wavelet form, it fits in several places of a time series, even if it is not present at all. Both Mexican hat wavelet and Daubechie 4 wavelet proved complex enough to allow a safe rediscovery.



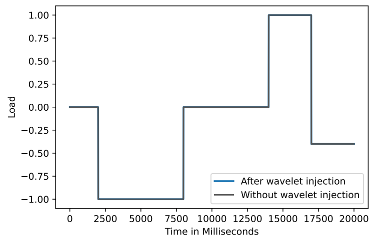
(a) Daubechie 4 Wavelet in Aircharge



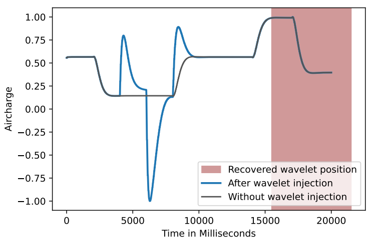
(b) Daubechie 4 Wavelet in Load



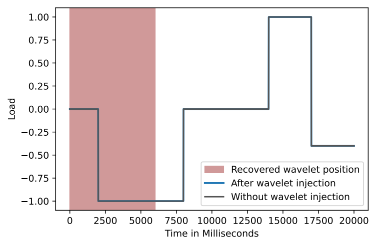
(c) Mexican hat Wavelet in Aircharge



(d) Mexican hat Wavelet in Load



(e) Haar Wavelet in Aircharge



(f) Haar Wavelet in Load

Figure 4.3: The wavelets as they were discovered in the exemplary aircharge and load variable. The area where the lines diverge indicates the presence of a signal and is highlighted green as a mark for successful recovery. Red highlights indicate a wrong recovery. If the lines do not diverge, no wavelet is present and nothing should be discovered.

Step 4: Causal Inference and Results

From the previous step, we were able to retrieve from the Mexican hat and the Daubechie 4 wavelet injection independently four path constraints $\{C(\text{Angle}, \text{Air intake}), C(\text{Angle}, \text{Aircharge}), C(\text{Angle}, \text{Torque}),$ $C(\text{Angle}, \text{Engine Speed})\}$. Next, we implemented a brute force algorithm, that generated all viable graphs for given variables and eliminated all graphs from the set not supporting the constraints. In total, we were able to reduce the number of graphs from 1,073,741,824 to 23,855,104 and eliminated with this method approximately 97.8% of potential causal graphs. The number of graphs may be reduced further by performing additional wavelet injections.

5 Conclusion

We explained the idea of discovering causal knowledge by injecting and retrieving wavelets in causal variables. For injection, we simply added a chosen wavelet to the incoming timeseries of a variable and tried to rediscover it in the depending variables via fast pattern matching. We gained causal information by defining path constraints to restrict the equivalence class for the true causal graph, as a path is assumed to be present between injected variable and the variable of rediscovery. We demonstrated the procedure on a running combustion engine simulation by adding three different wavelets (Haar, Daubechie 4 and Mexican hat) to an actuated variable. The procedure performed well for the Mexican hat wavelet and the Daubechie 4 wavelet. By using either of them, we were able to receive four path constraints and to reduce with them the number of graphs from 1,073,741,824 to 23,855,104.

References

[1] PR Crossley and JA Cook. “A nonlinear engine model for drivetrain system development”. In: *International Conference on Control 1991. Control’91*. IET. 1991, pp. 921–925.

[2] Frederick Eberhardt and Richard Scheines. “Interventions and causal inference”. In: *Philosophy of science* 74.5 (2007), pp. 981–995.

[3] Ronald Aylmer Fisher. “Design of experiments”. In: *British Medical Journal* 1.3923 (1936), p. 554.

[4] John J Moskwa and J Karl Hedrick. “Automotive engine modeling for real time control application”. In: *1987 American Control Conference*. IEEE. 1987, pp. 341–346.

[5] Josephine Rehak. “A Proposal on Discovering Causal Structures in Technical Systems by Means of Interventions”. In: *Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. 2021, p. 91.

[6] Josephine Rehak. “A Review on Approaches for Causal Structure Identification”. In: *Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. 2022, p. 105.

[7] Robert W Weeks and John J Moskwa. “Automotive engine modeling for real-time control using matlab/simulink”. In: *SAE transactions* (1995), pp. 295–309. URL: `https://www.mathworks.com/help/simulink/slref/modeling-engine-timing-using-triggered-subsystems.html`.

[8] James Woodward. *Making things happen: A theory of causal explanation*. Oxford university press, 2005.

[9] Chin-Chia Michael Yeh et al. “Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets”. In: *2016 IEEE 16th international conference on data mining (ICDM)*. Ieee. 2016, pp. 1317–1322.