ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA
Universidad de Córdoba

BACHELOR THESIS

Bachelor's Degree in Computer Science Engineering

Causal Inference for Time Series Analysis: Theory and Application

Inferencia causal para el análisis de series temporales: teoría y aplicación

Author
Joaquín Mateos Barroso

Supervisors
José María Luna Ariza
Sebastián Ventura Soto

Junio, 2025

Statement of Authorship

Joaquín Mateos Barroso declares that he is the author and assumes the originality, understood in the sense that he has not used sources without properly citing them, of the Bachelor Thesis:

Causal Inference for Time Series Analysis: Theory and Application

Signed:

Abstract

Understanding causal relationships in time series data is crucial across domains such as economics, engineering, and neuroscience. While existing methods have shown success in discovering causality among individual time series, real-world problems often involve systems with multiple, interacting sequences. This project addresses the challenge of time series group causal discovery — inferring causal relations between clusters of time series. We propose a novel computational framework that extends traditional causal discovery methods to handle grouped time series data. Building upon recent theoretical work, we implement and empirically evaluate various previously exposed algorithms, together with new proposals for group-level strategies. These proposals are designed to overcome the low recall and spurious relations observed in previous methods, by introducing subgroups or group embeddings that allow to leverage the amount of information used by auxiliary algorithms. We introduce a new library, *group-causation*, to study and benchmark time series group causal discovery. Experiments conducted on synthetic and real datasets demonstrate the scalability and accuracy of our proposed methods, particularly under high-dimensional and complex dependencies. Our results highlight the promise of combining different group reduction strategies to improve causal understanding in multivariate time series data.

Key words: Artificial Intelligence, Machine Learning, Causal Inference, Time Series Analysis.

Resumen

Comprender las relaciones causales en series temporales es crucial en ámbitos como la economía, la ingeniería, o la neurociencia. Aunque los métodos existentes han demostrado su eficacia para descubrir causalidad entre series temporales individuales, los problemas del mundo real suelen involucrar sistemas con múltiples secuencias que interactúan entre sí. Este proyecto aborda el reto del descubrimiento causal de grupos de series temporales. Proponemos un nuevo marco computacional que amplía los métodos tradicionales de descubrimiento causal para manejar datos de series temporales agrupadas. Basándonos en trabajos teóricos recientes, implementamos y evaluamos empíricamente varios algoritmos previamente expuestos y nuevas propuestas de métodos para el descubrimiento causal a nivel de grupos. Estas propuestas están diseñadas para superar la baja sensibilidad y las relaciones espurias observadas en los métodos anteriores, introduciendo subgrupos o embeddings de grupos que permiten equilibrar y aprovechar la cantidad de información utilizada por los algoritmos auxiliares. Presentamos una nueva librería, *group-causation*, para estudiar y evaluar comparativamente el descubrimiento causal de grupos de series temporales. Los experimentos llevados a cabo en datos sintéticos y reales demuestran la escalabilidad y precisión de los métodos propuestos, sobre todo en el caso de dependencias complejas y alta dimensionalidad. Nuestros resultados ponen de relieve la promesa de combinar diferentes estrategias de reducción de grupos para mejorar la comprensión causal en series temporales multivariantes.

Palabras clave: inteligencia artificial, aprendizaje automático, inferencia causal, análisis de series temporales.

"You cannot answer a question that you cannot ask, and you cannot ask a question that you have no words for."

Judea Pearl

"All we have to decide is what to do with the time that is given us."

J.R.R. Tolkien

Contents

List of Figures

List of Tables

List of Algorithms

List of Definitions

1. Introduction

In numerous scientific and industrial fields, understanding the underlying causal mechanisms is essential for accurate modeling, prediction, and decision-making. Traditional statistical methods often focus on identifying correlations, which, while informative, do not necessarily imply causation. Causal inference, however, seeks to uncover the true cause-and-effect relationships, providing deeper insights and more reliable foundations for action [Pearl, 2009].

Time series data, which consist of sequential observations over time, are prevalent in disciplines such as economics, neuroscience, climatology, and engineering. Analyzing these data to discern causal relationships poses unique challenges due to temporal dependencies and potential confounding factors. Causal discovery in time series aims to identify how variables influence each other over time, offering valuable information for forecasting, intervention strategies, and system optimization [Runge et al., 2023].

While significant progress has been made in causal discovery for individual time series, many real-world applications involve systems with multiple, possibly interrelated, time series. For instance, in financial markets, the price movements of different assets are interconnected; in predictive maintenance, various engine components interact as a whole within complex mechanisms. Therefore, developing methodologies capable of handling groups of time series is crucial for capturing the intricate web of causal relationships in such systems.

In this project, we present a computational framework tailored for causal discovery across groups of time series, mainly following the mathematical foundations detailed in [Wahl et al., 2024], which we implement in a new library named after *group-causation*. We also propose new approaches for this task, using advanced statistical techniques to manage the complexities arising from multiple interacting time series, ensuring scalability and robustness. We conduct comprehensive experiments using synthetic and

real data to evaluate the performance of our framework, comparing it with other algorithms through the use of state-of-the-art algorithms including PCMCI [Runge et al., 2019] and DYNOTEARS [Pamfil et al., 2020]. The results demonstrate the effectiveness of our approach in accurately identifying causal structures, even in challenging scenarios with high-dimensional data and complex dependencies.

2. Causal Inference Preliminaries

Classical Statistical Inference [Vélez Ibarrola and Pérez, 2013] studies a probable theoretical distribution, using as basis a sample of this distribution, estimating parameters and testing hypotheses. Its techniques do not necessarily rely on causal relations; the covariance, Definition A.4, between random variables, for example, is a symmetrical statistic.

On the other hand, **Causal inference** aims to understand and answer questions like "Does treatment A cause outcome B?" (causal discovery), or "What would have happened to the variable A if B had happened?" (counterfactuals) [Pearl and Mackenzie, 2018].

It is important to note that these questions are significantly different from traditional questions such as "Is B statistically dependent (non-independent) of A?" (what would just indicate that, in presence of certain value of A, B has a different distribution; $f_{A,B} \neq f_A \cdot f_B \Leftrightarrow f_B \neq f_{B|A}$) or "What is the best estimator of A knowing that B has happened?" (what is obtained with Bayes estimators via a conditioned distribution [Vélez Ibarrola and Pérez, 2013], $\pi(a|B=b) = \frac{\pi(a)f_a(b)}{\int_{W_A} f_t(b)\pi(t)dt}$, and inside the integral considers different events that are more probable due to the fact $B = b$).

These tools are very useful for prediction, and modern machine learning uses them constantly, but causal inference allows extracting causal implications that might be useful for both a better comprehension of the data and relationships between variables, and obtaining, in some cases, even better estimators than the ones obtained via traditional methods, such as Bayes or minimal risk.

A classical illustration of the saying "correlation does not imply causality" is the **Simpson's paradox** [Pearl and Mackenzie, 2018], which states that, for example, we could obtain a positive Pearson correlation between smoking and getting good grades in a difficult math test. Though, this correlation might be caused by people with higher

age smoking more and also getting better grades. In Figure 2.1 a representation of this happening can be seen, where the x axis could be the quantity of cigarettes taken per month, and y axis the score, over 100 in the math test. If the groups are ordered according to their age (Group 1 has lowest age and 3 the highest), then we can see that the colored regressions make much more sense than the black line.

While statistical inference or classical machine learning could use the black line, and it would be very useful for the task at hand, causal inference tries to find these causalities rather than correlations.

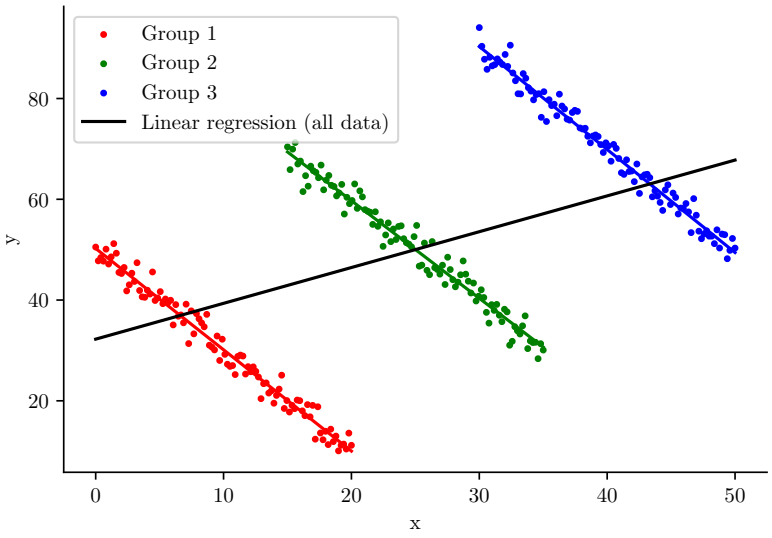


Figure 2.1: Example of Simpsons' paradox

2.1. Structural Causal Models

In order to assert implications between variables it is necessary to know what these implications mean, and Structural Causal Models (SCMs), as explained in [Pearl, 2009, Section 1.4] are the most direct and human-like way of directly defining these implications.

Definition 2.1 (Structural Causal Model)**.** *A **SCM** is defined by three finite sets:*

- *Exogenous random variables, $\boldsymbol{U}$, that are obtained from external methods.*
- *Endogenous random variables, $\boldsymbol{V}$, that are obtained as a function of variables from both $\boldsymbol{U}$ and $\boldsymbol{V}$.*

- *Causal relations,* $\boldsymbol{f}$*, that is, a set of* $|\boldsymbol{V}|$ *functions*[1]*,* $f_i : \mathcal{P}(\boldsymbol{U} \cup \boldsymbol{V} \backslash \{V_i\}) \to \boldsymbol{V}$*, one for each variable,* $V_i \in \boldsymbol{V}$*, in such a way that it can be assigned* $V_i = f_i(Pa(V_i))$*, being* $Pa(V) \subseteq \boldsymbol{U} \cup \boldsymbol{V} \backslash \{V_i\}$ *the set of parents of* V_i*. Certain properties needed for these equations to be useful will be developed in Section 2.3.*

An example of a basic Structural Causal Model would be having sets with one single variable, $\mathbf{U} = \{U\}$, $\mathbf{V} = \{V\}$, where the exogenous variable could follow a normal distribution $U = \mu_U \sim \mathcal{N}(0, 1)$, and the endogenous variable could be dependent in a linear way; $V = U + \mu_V$, being $\mu_V \sim \mathcal{N}(0, 1)$ the random noise of V. Until this moment, the model could be simply seen as a vector of random variables, but [Pearl, 2009, Section 3.2] interventions provide the first important causal tool.

Definition 2.2 (Intervention)**.** *An* ***intervention*** *in a SCM* $\mathbb{M}$ *is the modification of one or various of the definition assignments.*

A ***hard intervention*** *is the replacement of causal relation so that an endogenous variable becomes an exogenous one. Following the previous example, an intervention on* V*, would be* $do(V := k)$*, for* $k \in \mathbb{R}$*, and the new* U *distribution (that in this case would remain unchanged) would be* $P_U^{do(E:=k)} = \mathcal{N}(0, 1)$*.*

A ***soft intervention*** *is an intervention on a causal relation that doesn't modify the endogenous variables set e.g., in the same example as before,* $do(V := 3U + \mu_V)$*.*

Figure 2.2: Example: Cause-effect intervention.

Given the SCM $\mathbb{M} \equiv \{U := \mu_U, V := 4 \cdot U + \mu_V\}$, being $\mu_U, \mu_V \sim \mathcal{N}(0, 1)$, we have:

$$P_V^{\mathbb{M}} \sim 4\mu_U + \mu_V \sim \mathcal{N}(0, 5)$$

$$P_V^{do(U:=2)} \sim 4 \cdot 2 + \mu_V \sim \mathcal{N}(8, 1) = P_{V|U=2}$$

$$P_U^{do(V:=4)} \sim \mu_U \sim P_U^{\mathbb{M}} \neq P_{U|V=4}^{\mathbb{M}}$$

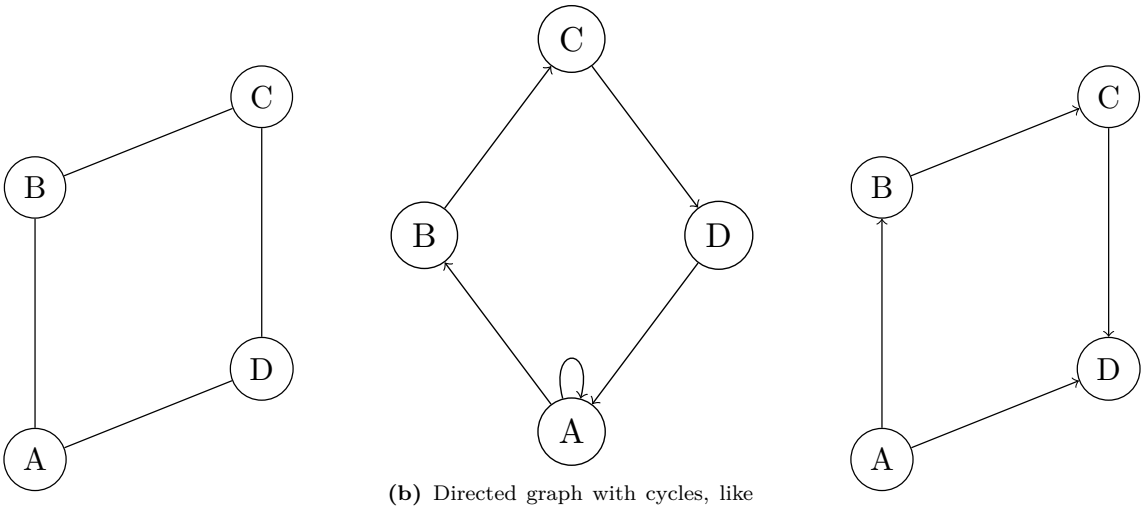
Interventions on U change the distribution of V, but interventions on V do not have effect on U, despite U and V may be dependent.
This asymmetry can also be formulated from the independence of U and V when intervening with $do(V := \mu'_V)$, but remaining dependent when intervening with $do(U := \mu'_U)$.

Example in Figure 2.2 helps clarifying the difference between this model and the use of the conditional probability.

[1]Given a set X, $\mathcal{P}(X) = 2^X$ denotes the power set of X, i.e., the set of all subsets of X.

2.2. Bayesian Networks

When representing conditional information, using nodes and arrows seems to be an intuitive approach. The basic structure that is employed for this purpose, with great historical relevance in many fields of mathematics, is the **Directed Acyclic Graph (DAG)**[2]:



(a) Undirected graph

(b) Directed graph with cycles, like $A \to A$ or $B \to C \to D \to A \to B$.

(c) DAG

Figure 2.3: Representations of basic types of graphs.

- In the *graph*, nodes are an abstract representation of a measure or a group of measures. This will involve the use of either a random variable or stochastic process in the theoretical case, or a time series when data is collected. Some representations of graphs can be seen in Figure 2.3.

- The condition of *acyclity* (no directional path will start and end in the same edge) is, on the one hand, a quite intuitive assumption, as stating that a process has implications on itself is trivial, and on the other hand allows to develop a consistent theoretical formulation of causal problems. Note however that this condition does not mean that a variable will not be able to have implications on the future of itself, what will be represented not through a cycle, but through an edge in the way $X^i_{t-1} \to X^i_t$.

Using this structure, [Pearl, 1988, Chapter 3] developed a formal model that aimed to provide machines with a way of reason probabilistically about the world:

[2]A formal definition of these objects can be consulted in Appendix A.2.

Definition 2.3 (Bayesian Network)**.** *A **Bayesian Network** is a set of random variables $\boldsymbol{X}$, represented as nodes of a DAG $\mathcal{G}$, that due to being nodes of a DAG are associated with a topological sorting, $\mathcal{V} := \boldsymbol{X} = \{X^1, ..., X^n\}$, and their conditional probability distributions satisfy*[3]

$$P(x^i|x^1, ..., x^{i-1}) = P\left(x^i|Pa(x^i)\right),$$

where $Pa(X^i)$ is the set of parents of X^i in $\mathcal{G}$, and no other subset of $\{X^1, ..., X^{i-1}\}$ satisfies the same property.

This network defines a unique joint probability (or density) distribution,

$$P(x^1, ..., x^n) = \prod_i P\big(x^i|Pa(x^i)\big)$$

for which computation it suffices to know the unconditional distributions of the exogenous variables $\boldsymbol{U}$, that is, those st. $Pa(X^i) = \emptyset$, and the conditional distributions of the endogenous variables $\boldsymbol{V} = \boldsymbol{U}^c$ given their parents.

These networks have been proven to be very useful in a wide variety of applications in medicine [Lewis and Groth, 2020], predictable maintenance, [Partovi et al., 2022] machine learning [Galvani et al., 2021] and many more fields [Iqbal et al., 2015]. Nevertheless, they are not able to represent causal information, as the Bayesian Networks purely use conditional probability, which is able to pass information both from cause to effect and from effect to cause. That way, even using the simplest nontrivial model, $\mathcal{G} = (\mathcal{V}, \mathcal{E}) = \big(\{U, V\}, \{U \to V\}\big)$, one can check that observing V influences the belief in U, as studied in example of Figure 2.2.

2.2.1. d-separation

Given a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set of known variables $\boldsymbol{Z} \subsetneq \mathcal{V}$ it is important to know whether two nodes $U, V \in \mathcal{V}$ are connected, in the sense that U has information about V (or vice versa) that is not in $\boldsymbol{Z}$. This is exactly what conditional independence tests, Appendix A.4, do. [Pearl, 2009, Theorem 1.2.4] demonstrated that this probabilistic

[3]Note that, for the continuous random variables case, the standard, followed here, does the notation abuse of specifying the density function through a probability distribution function. Also, to simplify formulation, we are denoting $P(x^i) := P\{X^i = x^i\} = P\{\omega \in \Omega : X^i(\omega) = x^i\}$, where Ω is the sample space.

property of conditional independence is equivalent to the graphical property of d-separation:

Definition 2.4 (d-separation)**.** *Given a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, two nodes $U, V \in \mathcal{V}$ are said to be **d-separated** (directionally separated) by $\boldsymbol{Z} \subsetneq \mathcal{V}$ if and only if every path from U to V is blocked. A path $\pi \equiv (U \to X^1, ..., X^l \to V)$ is said to be blocked if and only if any of the following cases occurs in π:*

- ***Causal chain**: $X \to Z \to Y$, being $Z \in \boldsymbol{Z}$. Z is a mediator.*
- ***Confounding**: $X \leftarrow Z \to Y$, being $Z \in \boldsymbol{Z}$. Z is a common cause of X and Y.*
- ***Collider**: $X \to W \leftarrow Y$, being $W \notin \boldsymbol{Z}$ and no descendant of W is in $\boldsymbol{Z}$.*

*If U, V are not d-separated, there exists a path that is not blocked, and they are said to be **d-connected**.*

2.3. Causal DAGs

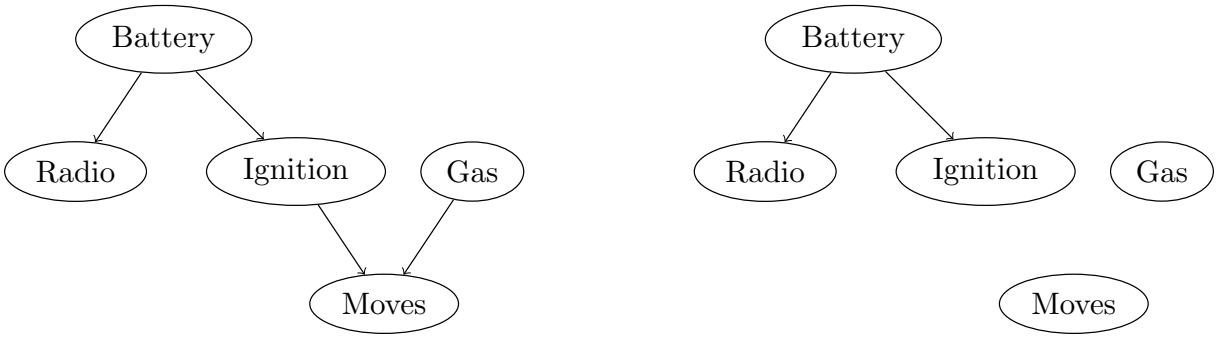
It would be interesting to combine the graphical representation of Bayesian Networks with the causal structure of SCMs, and with this purpose [Pearl, 2009, Section 1.3] designed Causal DAGs.

Definition 2.5 (Causal DAG)**.** *Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a Bayesian Network, with $\mathcal{V} = \{X^1, ..., X^n\}$. Let $\mathcal{G}^{do(X^i:=x)}$ and $P^{do(X^i:=x)}$ denote the graph and distribution result, respectively, from the **intervention** $do(X^i = x)$, i.e., extrapolating Definition 2.2 to graphs by deleting all edges $X^j \to X^i$, for $j \in \{1, ..., n\}$, and just considering as possible those samples in the space $\{X^i = x\}$. Then $\mathcal{G}$ is said to be a **Causal DAG** (or Causal Bayesian Network), and its probability distribution is*

$$P^{do(X^i=x)}(x^1, ..., x^n) = \begin{cases} \prod_{j \neq i} P\big(x^j | Pa(x^j)\big) & \text{if } x^i = x \\ 0 & \text{otherwise} \end{cases}$$

These networks do have causal information, and allow to graphically and efficiently represent causal information about real systems. For example, if one wanted to represent the behavior of a vehicle, he could create the Causal DAG in Figure 2.4a, which represents some information like the fact that having battery allows starting the car; *Battery* $\to$

Ignition and it is necessary to both have started the car and to have gas in it in order to allow it to move by itself; $Ignition \rightarrow Moves \leftarrow Gas$. There are 2 ways to see how information goes the other way around:



(a) Base Causal DAG. **(b)** Causal DAG intervened on $do(Moves = True)$.

Figure 2.4: Example of a Causal DAG representing a vehicle state.

- If we saw that the car was moving, then we could suspect that its motor is working, and we would modify our belief about the gas:

$$P\{Gas = True | Moves = True\} > P\{Gas = True\}.$$

- On the other hand, if we found a car and we forced the car to move by putting the car on a conveyor belt, $do(Moves = True)$, then no information could be extracted about whether the found car has gas or not, and this is graphically represented as deleting the edges $Ignition \not\rightarrow Moves \not\leftarrow Gas$, Figure 2.4b:

$$P^{do(Move=True)}\{Gas = True\} = P\{Gas = True\}.$$

From this kind of models it is easy to generate a dataset, as seen in Algorithm 1. Due to the capacity to find a total order in $\mathcal{V}$ based on the implications, this algorithm is guaranteed to iterate over all variables $X^i \in \mathcal{V}$.

2.4. Assumptions

There are some assumptions [Hasan et al., 2024] that are necessary to work with the model, like the ones we've made during previous definitions, and other assumptions about the way in which models are created and data is sampled that allow turning the process around and extracting the Causal DAG once a dataset is given. Some of the

Algorithm 1 Algorithm to obtain a data sample from a (potentially Causal) Bayesian Network

Input: Bayesian Network with a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, being $\mathcal{V} = \{X^1, ..., X^n\}$ and a set of equations $\{X^i = f_i(Pa(X^i), \mu_i\}$.
Output: Sampled data $\mathbf{x} = \{x^1, ..., x^n\} \in \mathbb{R}^n$
procedure SAMPLING ALGORITHM
    $\mathcal{U} \leftarrow$ exogenous variables $= \{X^i \in \mathcal{V} : \nexists X^j \in \mathcal{V}$ s.t. $X^j \rightarrow X^i\}$
    for $X^i \in \mathcal{U}$ **do**
        $x^i \leftarrow$sample(μ_i)
    while $\exists X^i \in \mathcal{V}$ s.t. all variables in $Pa(X^i)$ are known **do**
        noise $\leftarrow \mu_i$
        $x^i \leftarrow f_i(Pa(x^i)$,noise$)$
    return $\mathbf{x} = \{x^1, ..., x^n\}$

most important are listed here, and, in general, the first 3 will be assumed without any need to mention them. We will assume that there is a theoretical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $X^i \in \mathcal{V}$ is a random variable, from which a dataset $\boldsymbol{\mathcal{D}}$ has been sampled.

Definition 2.6 (Causal Markov Condition)**.** ***Causal Markov Condition*** *is said to be fulfilled iff every variable $X^i \in \mathcal{V}$ is conditionally independent of its non-descendants, given its direct causes (parents) in the causal graph. I.e., $\forall X^j \in \mathcal{V}$, it is true that $X^i \perp\!\!\!\perp X^j | Pa(X^i)$.*

Intuitively, this means that the parents of the variable encode all the relevant information about it. This assumption is equivalent to the one formulated during the definition of base Bayesian Networks, Definition 2.3. Next definitions are more related with the distribution of the data.

Definition 2.7 (Faithfulness Assumption)**.** ***Faithfulness Assumption*** *is said to be fulfilled iff all observed conditional independencies in the data are reflected in the graph, in the way of d-separation. I.e., $\forall X^i, X^j \in \mathcal{V}$, $S \subseteq \mathcal{V}$, if $X^i \perp\!\!\!\perp X^j | S \Rightarrow X^i$ is d-separated of X^j in $\mathcal{G}|S$.*

Definition 2.8 (Causal Sufficiency)**.** ***Causal Sufficiency Assumption*** *is said to be fulfilled iff there are no latent/hidden/unobserved confounders, i.e., all the common causes are measured.*

This one is a strong assumption that may not hold in real-world models, and users of Causal models need to be aware about it.

Definition 2.9 (Further Data Assumptions)**.** *Some other typical assumptions are:*

- *Existence of* ***linear relationships*** *in the way of* $X^i = aX^j + \mu_i$*, having* ***continuously valued*** *or* ***discretely valued*** *data.*
- *Assumptions are about the distribution of data, like assuming that variables are i.i.d.;* ***independent and identically distributed****, or that they follow a distribution that moves in time, like a* ***time series****, Appendix A.3.*
- ***Stationarity***[4]*; data preserves statistical properties along time, e.g., moments such as mean, variance, ...*
- *Sometimes, it is also assumed that noise* μ_i *follows a specific type of* ***distribution****, such as Gaussian, Exponential, Weibull, etc.*

2.5. Time Series DAGs

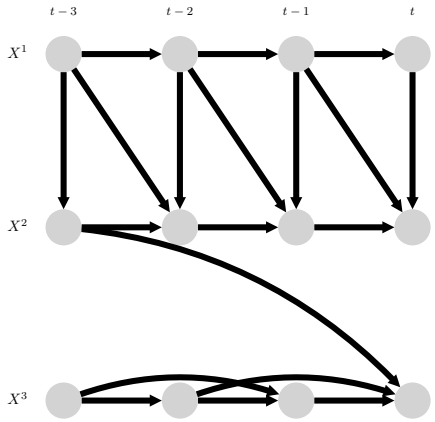
Up to this point, variables have been assumed to be distributed i.i.d.. On the other hand, many real world problems follow a time series structure, which is commonly modeled through stochastic processes, Section A.3. In order to apply causality in this context a new framework was developed an [Runge et al., 2023]. In this section the previous graph definitions are extended to time series and in the next one these algorithms are presented. We will assume a given, countable, set of moments for observation T.

Definition 2.10 (Full-Time Causal DAG)**.** *A* ***Full-Time Causal DAG*** *is* $\mathcal{G} = (\mathcal{V}^F, \mathcal{E}^F)$*, where* $\mathcal{V}^F$ *is a family of sets of nodes* $(\mathcal{V}_{-\infty}, ..., \mathcal{V}_t, ..., \mathcal{V}_\infty)$*, where each* $\mathcal{V}_t$ *is a set of momentary nodes* $X_t^1, ..., X_t^n$*, and* $\mathcal{E}^F$ *is a set of edges between these nodes;* $\{X_t^i \to X_{t'}^j\}_{i,j,t,t'}$*, restricted to* $t \leq t'$ *(implications don't go back in time) and to* $i \neq j$ *always that* $t = t'$*, so that there is no auto-causation in the same instant.*

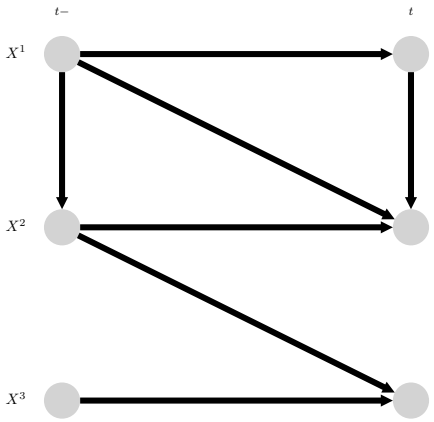
These graphs are computationally intractable, so they are compiled by checking the property of **Consistency Throughout Time**; there exists $\gamma \in \mathbb{N} - \{0\}$ s.t. causal structure of nodes in $\{\mathcal{V}_{t-\gamma}, ..., \mathcal{V}_t\}$ is the same for any $t \in T$. The maximum possible γ is named after maximal temporal lag.

[4]The fact that a stochastic process is stationary is defined has having $F_X(X_{t_1+\tau}, ..., X_{t_m+\tau}) = F_X(X_{t_1}, ..., X_{t_m})\ \forall \tau \in T$.

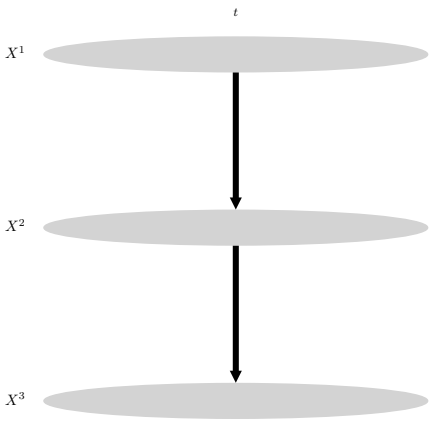
Definition 2.11 (Window Causal DAG)**.** *A* ***Window Causal DAG*** *of a full time causal graph $\mathcal{G}^F$ with a finite maximal temporal lag, γ, is the subgraph $\mathcal{G}^W = (\mathcal{V}^W, \mathcal{E}^W)$ s.t. $\mathcal{E}^W = \mathcal{E}^F$ and $\mathcal{V}^W = (\mathcal{V}_{t-\alpha}, ..., \mathcal{V}_t)$, where γ is the maximal temporal lag of $\mathcal{G}^W$. Example in Figure 2.5a.*



(a) Window Causal DAG.



(b) Extended Summary causal DAG.



(c) Summary Causal DAG.

Figure 2.5: Example of basic time series DAGs.

These are the typical time series DAGs that models try to predict from data, but there are more graphs that do also represent important information about the causal relations in a more concise way.

Definition 2.12 (Summary Causal graph)**.** *The* ***Extended Summary Causal graph*** *of a Window Causal DAG $\mathcal{G}^W$ is $\mathcal{G}^E = (\mathcal{V}^E, \mathcal{E}^E)$ s.t. $\mathcal{V}^E = (\mathcal{V}_{t^-}, \mathcal{V}_t)$, edges in $\mathcal{E}^E$ are*

initially the same as in $\mathcal{E}^W$*, and edges* $X^i_{t^-} \to X^j$ *are added to* $\mathcal{E}^E$ *iff* $\exists l \in \{1, ..., \gamma\}$ *s.t.* $X^i_{t-l} \to X^j_t$ *in* $\mathcal{G}^W$*. Example in Figure 2.5b.*

The ***Summary Causal graph*** *of a Window Causal DAG* $\mathcal{G}^W$ *is* $\mathcal{G}^S = (\mathcal{V}^S, \mathcal{E}^S)$ *s.t.* $\mathcal{V}^S = \mathcal{V}_t$ *and* $X^i_t \to X^j_t$ *is in* $\mathcal{E}^S$ *iff* $\exists l \in \{0, ..., \gamma\}$ *s.t.* $X^i_{t-l} \to X^j_t$ *in* $\mathcal{G}^W$*. Example in Figure 2.5c.*

3. State of the Art

3.1. Causal Discovery

Causal Discovery can be defined as the process of, given a dataset and a set of assumptions, extracting a Causal DAG that represents the theoretical causal relations between variables as "faithfully" as possible, where the meaning of faithfully depends on the context and objective. In Figure 3.1 there is a taxonomy with the main causal discovery methods, and in this section there are basic descriptions and examples of each of these categories.

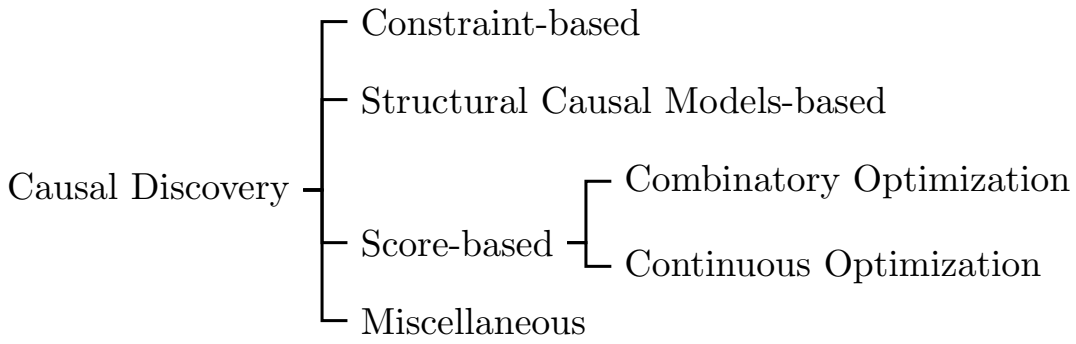


Figure 3.1: Taxonomy of Causal Discovery algorithms for tabular (i.i.d.) variables.

3.1.1. Constraint-based

These algorithms construct the skeleton by initializing all possible edges, as existing, like in Figure 3.2a, and deleting those that do not meet certain restrictions. The assumptions of Markov, Faithfulness and Causal Sufficiency imply that

$$X^i \to X^j \Leftrightarrow X^i \not\perp\!\!\!\perp X^j \mid S, \;\; \forall S \subseteq \mathcal{V} - \{X^i, X^j\}.$$

To check this condition, different Conditional Independency Tests, Section A.4, can be used. Another big problem is checking all possible values of S. As explained in

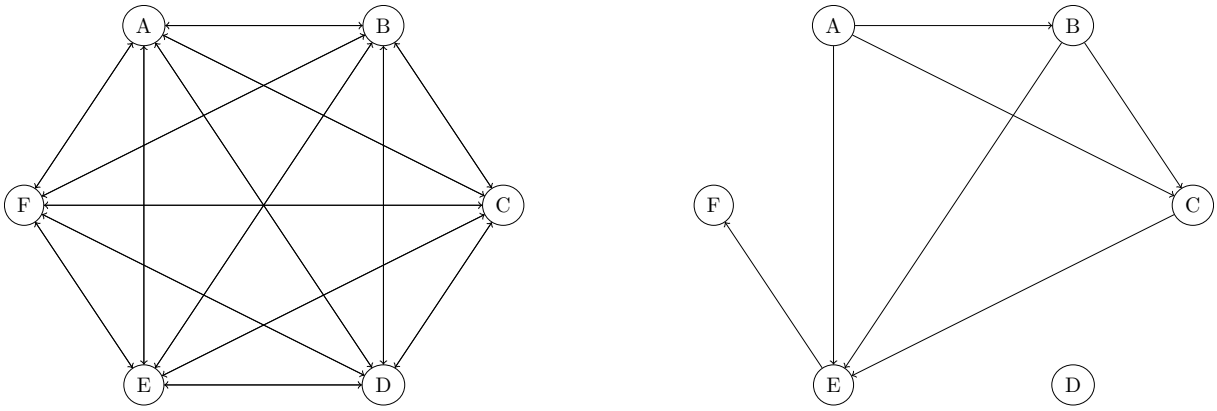
[Runge et al., 2019, Runge, 2020, Faller and Janzing, 2025], there are several drawbacks of testing conditional independence with many conditioning sets:

- Big conditioning sets force a low predicting power, leading to Type I errors (false positives).
- Small or improperly chosen conditioning sets give low accuracy, what results in more Type II errors (false negatives).
- A single misorientation can propagate through false positives cascades, resulting in a graph that does not accurately represent the true causal structure.

Several algorithms manage this problem by testing first over small conditioning sets, and stop testing edges when we are confident enough that at least one conditioning set makes the variables conditionally independent.

Algorithm 2 PC Algorithm for Causal Discovery

Input:
Set of variables $\mathcal{V}$,
dataset $\mathcal{D}$,
Significance level α
Output: A partially directed acyclic graph (PDAG) representing the equivalence class of DAGs.
Initialize the complete undirected graph $\mathcal{G}$ over $\mathcal{V}$.
for each pair (X, Y) in $\mathcal{V}$ **do**
    Set the separation set $S_{XY} = \emptyset$.
Set the conditioning set size $l \leftarrow 0$.
while there exists an edge (X, Y) in $\mathcal{G}$ with $\left|adj(X) \setminus \{Y\}\right| \geq l$ **do**
    for each edge (X, Y) in $\mathcal{G}$ such that $\left|adj(X) \setminus \{Y\}\right| \geq l$ **do**
        for each subset $S \subseteq adj(X) \setminus \{Y\}$ with $|S| = l$ **do**
            if $X \perp\!\!\!\perp Y \mid S$ (as determined by a statistical test on $\mathcal{D}$ at level α) **then**
                Remove edge (X, Y) from $\mathcal{G}$.
                Set $S_{XY} \leftarrow S$ and $S_{YX} \leftarrow S$.
                break out of the innermost loop.
    $l \leftarrow l + 1$.
Orient edges:
for each ordered triple (X, Y, Z) such that X and Z are not adjacent in $\mathcal{G}$ and $Y \in adj(X) \cap adj(Z)$ **do**
    if $Y \notin S_{XZ}$ **then**
        Orient edge $X - Y$ as $X \to Y$.
        Orient edge $Z - Y$ as $Z \to Y$.
Apply additional orientation rules until no more edges can be oriented.
Return $\mathcal{G}$.



(a) Initial fully connected skeleton. **(b)** Possible output of algorithm.

Figure 3.2: Example of structure learning with PC algorithm.

The most famous between these algorithms is the PC (Peter-Clark) Algorithm, designed by Peter Spirtes and Clark Glymour [Spirtes and Glymour, 1991]. It iterates in increasing order over all possible dimensions for the conditioning sets, starting from 0, and in each iteration tests all remaining edges conditioning over all possible conditioning sets of that dimension, always that the variables are not d-separated. An example of a possible output of this algorithm is shown in Figure 3.2b.

3.1.2. Structural Causal Models-based

SCMs provide a natural framework for representing cause-effect relationships as systems of equations. In the classical setting, SCMs often assume Gaussian noise, which can lead to identifiability issues. However, when the error terms are non-Gaussian, the model becomes identifiable even without prior knowledge of the causal ordering.

The most widespread method of this kind is the Linear Non-Gaussian Acyclic Model (LiNGAM) [Shimizu et al., 2006]. It is a particular SCM that leverages the non-Gaussianity of the error terms to discover causal directions. Under the LiNGAM assumptions, each observed variable X^i is modeled as

$$X^i = \sum_{X^j \in \mathrm{Pa}(X^i)} a_{ij} X^j + \mu^i = A_j X^{Pa(X^j)} + \mu^i.$$

In Gaussian SCMs, the covariance structure of the data is insufficient to determine the unique causal ordering. In contrast, the LiNGAM approach exploits higher-order statistical information (non-Gaussianity) to uniquely identify both the causal order and the connection strengths. This is achieved by applying independent component analysis

(ICA) to the residuals of the observed data, which decomposes the data into independent components corresponding to the external influences [Shimizu et al., 2012].

3.1.3. Score-based

Certain scores can be assigned to graphs depending on how well they are able to describe the data, and by trying to either maximize or minimize this score over the set of all graphs some algorithms are able to predict the generating bayesian network, which is, at data distribution level, equivalent to the searched causal DAG.

The most widely used fundamental score is the Bayesian Score [Koller and Friedman, 2009, Section 18.3], or its derivation **Bayesian Information Criterion (BIC)**. To calculate this score, first a Bayesian Network structure is assumed, so that, given a DAG $\mathcal{G}$, and applying Bayes Theorem A.1 one can obtain the probability or likelihood of $\mathcal{G}$ the generating structure of the given the dataset $\mathcal{D}$:

$$P(\mathcal{G}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{G})P(\mathcal{G})}{P(\mathcal{D})},$$

and since $P(\mathcal{D})$ is constant, we can obviate this factor.

A very interesting fact about the marginal likelihood $P(\mathcal{D}|\mathcal{G})$ can be observed by applying the total probability Theorem:

$$P(\mathcal{D}|\mathcal{G}) = \int \cdots \int_{\Theta} P(\mathcal{D}|\boldsymbol{\theta}, \mathcal{G}) P(\boldsymbol{\theta}|\mathcal{G}) d\theta_1 \cdots d\theta_r,$$

being $\boldsymbol{\theta}$ a specific vector of parameters for the graph and Θ the whole parameters space. This relation means that we are averaging between all possible parameters of the model, what implies that this score is conservative and doesn't assume that the best parameters can be always found, as maximum likelihood does.

After applying a logarithmic transformation to ease the computational calculations and some scaling one arrives at the Bayesian Information Criterion score:

$$\text{BIC} = \log \mathcal{L}(\hat{\boldsymbol{\theta}} : \mathcal{D}) - \frac{\log M}{2} \text{Dim}(\mathcal{G}),$$

where $\mathcal{L}$ is the likelihood operator, M is the number of instances and $\text{Dim}(\mathcal{G})$ is the number of independent parameters in $\mathcal{G}$, what allows benefiting those networks that explain well

the data with a low quantity of edges, thus generating a simple model that predicts the data with a high probability.

In addition to the BIC, several other scores are commonly used to assess model fit while balancing complexity. One popular measure is the **Akaike Information Criterion (AIC)**, defined as

$$\mathrm{AIC} = -2\log \mathcal{L}(\hat{\boldsymbol{\theta}} : \mathcal{D}) + 2\mathrm{Dim}(\mathcal{G}).$$

The AIC tends to favor models with a slightly better fit, although it may select more complex models than the BIC.

Another criterion is the **HannanQuinn Information Criterion (HQIC)**, which imposes a penalty of the form

$$\mathrm{HQIC} = -2\log \mathcal{L}(\hat{\boldsymbol{\theta}} : \mathcal{D}) + 2\mathrm{Dim}(\mathcal{G})\log(\log M),$$

where n is the number of observations. HQIC lies between AIC and BIC in terms of the penalty strength, often providing a compromise when the sample size is large.

Two main approaches have been developed to optimize these scores: Combinatory Optimization and Continuous Optimization.

Combinatory Optimization

Combinatorial optimization methods, directly explore the discrete space of DAGs. These approaches evaluate candidate graphs by computing their score and then search for the graph with the best score using techniques such as greedy search, dynamic programming, or other discrete search strategies. Although this approach can be computationally demanding due to the super-exponential number of possible graphs, many algorithms incorporate heuristics or restrictions (e.g., sparsity assumptions) to make the search tractable for moderate numbers of variables. Some metaheuristics approaches have used Ant Colony Optimization [de Campos et al., 2002] or Hill Climbing [Beretta et al., 2017].

Continuous Optimization

In continuous optimization methods, the discrete space of DAGs is embedded into a continuous space. Instead of searching over a combinatorial set of graphs, these

approaches relax the acyclicity constraint so that it can be expressed as a smooth function. The resulting optimization problem is then solved using gradient-based techniques. This framework has the advantage of leveraging modern continuous optimization tools and is particularly useful when the number of variables is large. Recent work in this direction includes differentiable acyclicity constraints that allow the use of standard gradient descent methods. The main algorithm that has developed this kind of optimization is NOTEARS (Non-combinatorial Optimization via Trace Exponential and Augmented lagRangian for Structure learning) [Zheng et al., 2018].

3.1.4. Hybrids

Hybrid algorithms aim to combine the strengths of both constraint- and score-based methods. A prominent example is the Max-Min Hill Climbing (MMHC) algorithm [Tsamardinos et al., 2006], which first employs constraint-based tests to narrow the search space and then applies a score-based search to refine the graph structure. Such methods often provide a good balance between computational tractability and robustness.

3.2. Causal Discovery on Time Series

Once the fundaments about causal discovery and time series DAGs have been presented, in this section there will be discussed the most important and representative time series causal discovery algorithms for time series and groups of time series that currently make up the state of the art of this technique.

Let $\boldsymbol{\mathcal{D}} = \{\mathcal{D}_1, ..., \mathcal{D}_k\}$ be a single time series dataset, where each $\mathcal{D}_t = \{x_t^1, ..., x_t^n\}$ contains the data points at a particular timestamp. The objective of Time Series Causal Discovery is to obtain a Window Causal DAG $\mathcal{G}$, with nodes $\{X_i^{t-\tau}\}_{i,\tau}$, that is best described by data $\boldsymbol{\mathcal{D}}$. Again, with this objective, different techniques have been developed, very similar to the ones studied for tabular data. Their taxonomy can be consulted in Figure 3.3.

Most of seen algorithms can manage time series DAGs directly. However, time series data has some particularities that they aren't always able to consider correctly:

- **Curse of dimensionality**: When dealing with a large number of time series variables and considering multiple lags, the number of potential conditioning sets

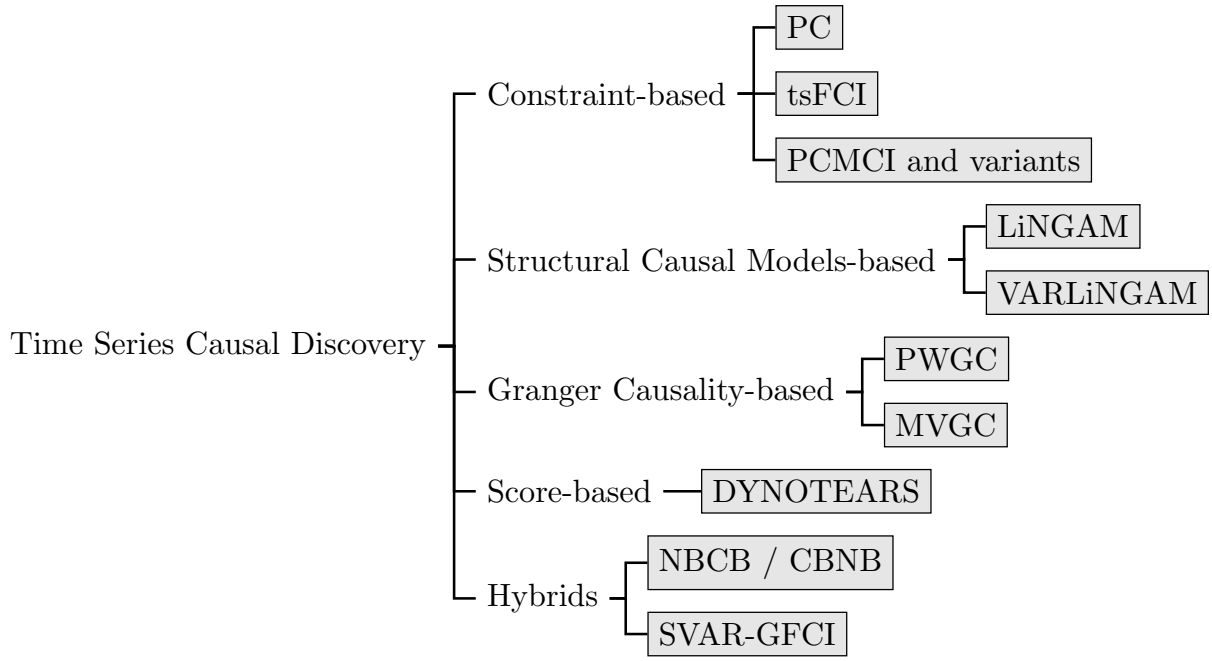


Figure 3.3: Taxonomy of Causal Discovery algorithms for time series variables. Gray boxes represent examples of types of algorithms.

can become very large, leading to computational challenges and reduced statistical power due to the increased dimensionality of the CI tests. In particular, the number of edges of a time series fully connected DAG with n variables and max lag τ is[1] $n(n-1)/2 + \sum_{i=1}^{\tau_{max}} n^2 = (\tau_{max} + 1/2)n^2 - n/2 \in \mathcal{O}(n^2\tau_{max})$. A graphical representation of this point is shown in Figure 3.4.

- **Sensitivity to Autocorrelation**: Strong autocorrelation, a common feature of time series data, can significantly affect the reliability of conditional independence tests used by the PC algorithm, potentially leading to both false positives (inferring spurious causal links) and false negatives (failing to detect true links)[Runge, 2020].

- **Assumption of Causal Sufficiency**: The basic PC algorithm assumes causal sufficiency, Section 2.4, so the presence of unobserved latent confounders can lead to incorrect causal inferences. Knowing that data follows a time series distribution may help to relax this assumption with specific algorithms.

[1]We are not counting repeating edges like $X^1_{t-1} \to X^2_{t-1} \equiv X^1_t \to X^2_t$.

Table 3.1: Comparison table between state-of-the-art algorithms for causal discovery over time series data. Number of cites were consulted on 23 May, 2025.

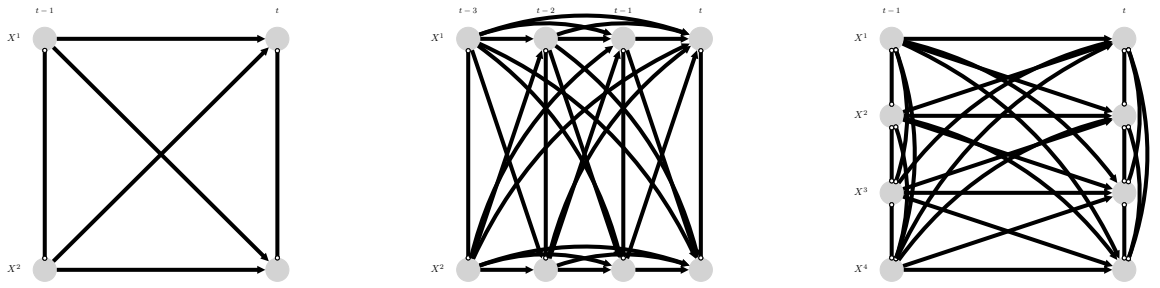
Algorithm	Description	Assumptions	Benefits	Drawbacks	Cited by
PC-Stable [Spirtes and Glymour, 1991]	Test Conditional Independencies	Causal Markov Condition, Causal Sufficiency, Faithfulness, no unobserved confounders	Good theoretical guarantees, manages non-linear relationships	Sensitive to autocorrelation, computationally expensive	1343
PCMCIplus [Runge et al., 2019] and [Runge, 2020]	Apply PC_1 and test Momentary Conditional Independence	Causal Markov Condition, Causal Sufficiency	Good theoretical guarantees, handles non-linear relationships, handles autocorrelation, handles hidden confounders, handles high-dimensionality	Computationally expensive	971 and 275
VARLiNGAM [Hyvärinen et al., 2010]	Suppose SCM and test residuals	Linear SCM, no unobserved confounders, non-Gaussian noises	No hyperparameters	Sensitive to hidden confounders, inapplicable to non-linear data	509
Granger Causality [Granger, 1969]	Test predicting power	Predicting power is able to predict Causal DAG	Fast to compute, simple model, extensible to non-linearity through more complex regressions	Specifically designed for economics, with no theoretical guarantees and creating spurious associations	37 537
DYNOTEARS [Pamfil et al., 2020]	Continuous Score Optimization	Score is able to predict Causal DAG	Fast to compute, handles high-dimensionality	Sensitive to parameters, inapplicable to non-linear data, good score doesn't guarantee faithful graph	250

3.2.1. Constraint-based

The same idea described in Section 3.1.1 can be applied to time series Causal Discovery with few modifications. Despite the classical PC algorithm can be extended to time series, other important methods based on the same principle have also become popular in the last years.

3.2.1.1. PC in time series

When the PC algorithm is extended to time series, it aims to discover causal relationships between variables not only at the same time point but also across different time lags. This involves considering lagged versions of the variables as potential causes of other variables at the current time point. Basic extensions of PC to time series have been implemented in the Tigramite package [Runge et al., 2025], as well as in Salesforce CausalAI Library [Ashimine, 2023].



(a) Fully connected graph with 2 variables and a max lag of 1. There are 5 edges[1].

(b) Fully connected graph with 2 variables and a max lag of 3. There are 13 edges[1].

(c) Fully connected graph with 4 variables and a max lag of 1. There are 22 edges[1].

Figure 3.4: Representation of the curse of dimensionality on time series causal DAGs. The number of edges is $\mathcal{O}(n^2\tau_{max})$.

To address these problems, other constraint-based algorithms have been developed, such as time series Fast Causal Inference [Entner and Hoyer, 2010] or PCMCI [Runge et al., 2019].

3.2.1.2. PCMCI and variants

PCMCI (PC_1 followed by Momentary Conditional Independence) is a two-stage causal discovery method specifically developed for large-scale, potentially nonlinear, time series

[1]We are not counting repeating edges like $X^1_{t-1} \to X^2_{t-1} \equiv X^1_t \to X^2_t$.

datasets. It aims to improve upon direct application of the PC algorithm to time series, which can be challenged by previously explained particularities. PCMCI assumes that the data is stationary and has time-lagged dependencies, and it also assumes causal sufficiency, Section 2.4.

The two stages of PCMCI are [Runge et al., 2019]:

1. **PC_1 Condition Selection:** This stage employs a variant of the skeleton discovery part of the PC algorithm, called PC_1. For each time series variable $X_t^j \in \{X_t^1, \ldots, X_t^n\}$, PC_1 aims to identify a preliminary set of potentially relevant causal parents $Pa(X_t^j)$ from the lagged versions of all variables in the system $(X_{t-1}, X_{t-2}, \ldots, X_{t-\tau_{max}})$. This is done through iterative unconditional and conditional independence testing with a liberal significance level $\alpha_{PC} \in [0, 1]$. The goal is to efficiently reduce the number of variables needed for conditioning in the subsequent stage by converging to a typically small set of relevant conditions that likely includes the true causal parents. After each iteration, the preliminary parents are sorted by their **test statistic value**.

2. **Momentary Conditional Independence (MCI) Test:** This stage takes the parent sets $Pa(X_t^j)$ identified by PC_1 and performs MCI tests to determine the presence of a causal link $X_{t-\tau}^i \to X_t^j$ for time delays $\tau \in \{1, \ldots, \tau_{max}\}$. The MCI test checks the **conditional independence** of $X_{t-\tau}^i$ and X_t^j given a conditioning set that includes the parents of the effect variable X_t^j (excluding the potential cause $X_{t-\tau}^i$) and optionally the time-shifted parents of the potential cause $X_{t-\tau}^i$:

$$MCI : X_{t-\tau}^i \perp\!\!\!\perp X_t^j | Pa(X_t^j) \setminus \{X_{t-\tau}^i\}, Pa_p^X(X_{t-\tau}^i) \quad \text{[Runge et al., 2019]}$$

   The MCI test aims to control false positives, especially in the presence of highly interdependent time series. The conditional independence tests used can accommodate nonlinear functional dependencies and both discrete and continuous variables. PCMCI efficiently exploits sparsity in the causal network, leading to polynomial computational complexity. Empirical results show that PCMCI often achieves higher detection power than other methods. PCMCI is implemented in the Tigramite open-source software package [Runge et al., 2025].

PCMCIplus is an extension of the PCMCI algorithm that aims to discover both

lagged and contemporaneous (instantaneous) causal links in autocorrelated nonlinear time series data. Traditional CI-based methods often struggle with strong autocorrelation, exhibiting low recall and inflated false positives. PCMCIplus also assumes **causal sufficiency**.

PCMCIplus builds upon PCMCI with two central modifications [Runge, 2020]:

1. **Separate Edge Removal for Lagged and Contemporaneous Conditioning:** Unlike standard PC-like algorithms, PCMCIplus separates the skeleton edge removal phase into two parts: one focusing on **lagged conditioning sets** and the other on contemporaneous conditioning sets. The lagged phase in PCMCIplus uses significantly fewer CI tests.

2. **Optimized Choice of Conditioning Sets using MCI Idea:** For the contemporaneous conditioning phase, PCMCIplus optimizes the choice of conditioning sets for individual CI tests by leveraging the momentary conditional independence (MCI) concept from PCMCI . This optimization helps to make the CI tests better calibrated under autocorrelation and increases detection power.

By optimizing the conditioning sets and separating the treatment of lagged and contemporaneous links, PCMCIplus aims to increase the effect size in individual CI tests, leading to higher detection power while maintaining well-controlled false positives, even with strong autocorrelation. Correct adjacency information obtained in the skeleton phase then leads to better orientation recall in the subsequent collider and rule-based orientation phases, which are similar to those used in the PC algorithm but adapted for the output of the modified skeleton phase. PCMCIplus is also implemented in Tigramite package [Runge et al., 2025]. Some variants of PCMCI have been developed, like L(atent)-PCMCI to deal with latent confounders, [Reiser, 2022], or like J(oint)-PCMCI to extract information from multiple datasets with different contexts [Günther et al., 2023]. Since these datasets manage situations that we are not going to consider explicitly, they will not be deeply explained here.

3.2.2. Structural Causal Models-based

The same structural causal models used for i.i.d. data can be used in time series, assuming linear dependencies, by using the standard Vector Autoregressive (VAR) model:

$$X_t^i = \sum_{\tau=1}^{\tau_{max}} \sum_{j=1}^{n} a_\tau^i X_{t-\tau}^j + \boldsymbol{\mu}_t = \sum_{\tau=1}^{\tau_{max}} A_\tau X_{t-\tau} + \boldsymbol{\mu}_t, \tag{3.1}$$

where A_τ are coefficient matrices capturing lagged temporal effects.

3.2.2.1. VARLiNGAM

The LiNGAM algorithm seen in Section 3.1.2 can be easily extended to time series applying Vector Autoregression, generating what is known as VARLiNGAM (Vector AutoRegression LiNGAM) [Hyvärinen et al., 2010]. This method, again, finds a causal ordering among variables by applying ICA to residuals, and is able to reconstruct both the temporal structure and the instantaneous causal structure, applying the base LiNGAM.

3.2.3. Granger Causality-based

A classical approach to a (non-causal) graph identification, originally applied in economics, is the Granger Causality [Granger, 1969]. Even though it recovers statistical relations between variables, and not causal ones, its extension to graphs has been shown to have a high graph prediction power [Nouri, 2023].

Let X_t^i and X_t^j be two stationary time series. In **Pairwise Granger Causality** (PWGC) we will take as restricted variable just the potential effect, $Y_t = X^j$, and in **Multivariate Granger Causality**(MVGC) we will take all the variables except for the potential cause, $Y_t = \{X_t^l | l \neq i\}$. To determine whether X^i Granger-causes X^j, we compare the predictive performance of two models:

$$\textbf{Restricted model:} \quad X_t^j = \alpha_0^j + \sum_{\tau=1}^{\tau_{max}} \alpha_\tau^j Y_{t-\tau} + \varepsilon_t^j, \tag{3.2}$$

$$\textbf{Unrestricted model:} \quad X_t^j = \beta_0^j + \sum_{\tau=1}^{\tau_{max}} \beta_\tau^j Y_{t-\tau} + \sum_{\tau=1}^{\tau_{max}} \gamma_\tau^j X_{t-\tau}^i + \eta_t^j, \tag{3.3}$$

where p is the number of lags, and ε_t and η_t are white noise error terms. The null

hypothesis is that X^i does not help predict X^j, i.e.,

$$H_0 : \gamma_1 = \gamma_2 = \cdots = \gamma_p = 0.$$

To compare the two models, one typically computes the residual sum of squares (RSS) for each model and uses an F-test. The test statistic is given by:

$$F = \frac{\left(RSS_{\text{restricted}} - RSS_{\text{unrestricted}}\right)/p}{RSS_{\text{unrestricted}}/(n - 2p - 1)},$$

which under H_0 follows an F-distribution with p and $n - 2p - 1$ degrees of freedom.

3.2.4. Score-based

Extending score-based causal discovery algorithms seen in Section 3.1.3 to the time series case involves adapting them to some particularities. Likelihood function is decomposed over time steps rather than individual independent samples. For instance, the joint likelihood of a time series can be factorized as a product of conditional likelihoods given past observations. The score (e.g., BIC) now reflects both the model fit at each time point and the models ability to capture the temporal dependencies.

Given the structural differences between DAGs and time series DAGs, and the increase in the size of the search space, algorithms based on discrete graph optimization haven't shown to have a good performance for time series causal discovery, contrary to those based in continous optimization.

3.2.4.1. DYNOTEARS

DYNOTEARS (DYnamic NOTEARS) is an extension of the NOTEARS framework that adapts the continuous score optimization specifically for time series data. It uses a linear model specified as

$$X_t = X_t \cdot \mathbf{W} + \sum_{\tau=1}^{\tau_{max}} X_{t-\tau} \cdot \mathbf{A}_\tau,$$

naming $\mathbf{W}$ as the weighted adjacency matrix of the intra-slice edges and $\mathbf{A}_\tau$ as the adjacency matrix of the inter-slice edges. That way, as explained in [Pamfil et al., 2020], the score optimization problem of a loss function $L(\mathbf{W}, \mathbf{A} : \mathcal{D})$ can be reduced to the

continuous minimization

$$\min_{\mathbf{W},\,\mathbf{A}} L(\mathbf{W}, \mathbf{A} : \mathcal{D}) \quad \text{s.t.} \quad h(\mathbf{W}) = 0, \tag{3.4}$$

where $h(\mathbf{W}) := \text{tr } e^{\mathbf{W} \circ \mathbf{W}} - N$, being $\circ$ is the Hadamard product. This is because [Zheng et al., 2018] demonstrated that the condition $h(\mathbf{W}) = 0$ is equivalent to the acyclity of the associated DAG.

3.2.5. Hybrids

Hybrid causal discovery methods combine ideas from different methodological families to harness complementary strengths.

NBCB (Neighborhood-Based then Constraint-Based) and CBNB (Constraint-Based then Neighborhood-Based) represent two alternative ordering of hybrid strategies in causal discovery mixing constraint-based methods and structural models [Bystrova et al., 2024]:

- NBCB: The method begins with a neighbourhood selection phase. This stage typically employs relatively simple structural criteria, such as correlation thresholds or sparsity-inducing regressions to select a candidate set of potential parents or neighbours for each variable. Then, in the constraint-based phase, detailed conditional independence tests refine the causal graph by pruning spurious edges.

- CBNB: In contrast, CBNB first performs a constraint-based analysis that tests for conditional independencies directly. After constructing an initial causal DAG, a subsequent neighbourhood-based refinement is conducted to add or confirm links that might have been missed or to tighten the estimated structure.

Other methods like SVAR-GFCI (Structural Structural Vector Autoregression - Greedy Fast Causal Inference) [Malinsky and Spirtes, 2019] use specific economic conditions identify relationships among economic variables by imposing certain structures on a VAR framework, over which a constraint-based algorithm is used to extract the causal DAG.

However, all of these methods rely on a strong set of assumptions or domain-specific conditions and work poorly in general scenarios. Even in original papers, there are not

significant differences in the general case between these and the original methods. For these reasons, general comparisons and surveys like [Assaad et al., 2022, Hasan et al., 2024] don't usually include them in the testing phase.

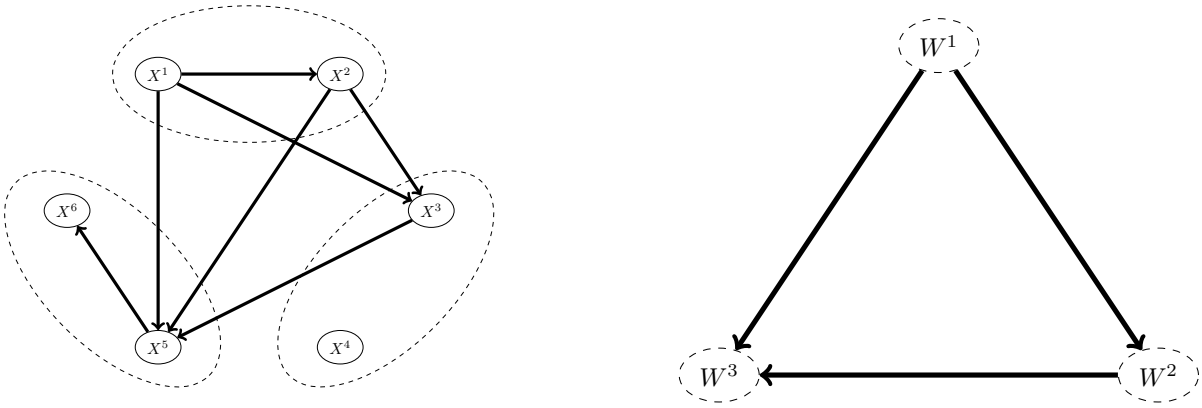
3.3. Causal Discovery on Groups of Time Series

When instead of variables there are groups of variables, the standard [Wahl et al., 2024] is to define that if any variable from a group has effects in another, then the first group is a cause of the second.

Definition 3.1 (Group Causal Graph)**.** *Let $\mathcal{G}^{micro} = (\mathcal{V}^{micro}, \mathcal{E}^{micro})$ be a causal DAG, that we will name micro-level causal DAG, and let $\mathcal{P}$ be a partition of $\mathcal{V}^{micro}$ with m subsets, or groups of variables $W^i = \{X^{i_1}, ..., X^{i_r}\} \subseteq \mathcal{V}^{micro}$. The associated **Group Causal graph** is the directed mixed graph (note that this graph isn't necessarily directed nor acyclic) $\mathcal{G}^{group}$ whose set of nodes is the chosen partition, $\mathcal{V}^{group} := \mathcal{P} = \{W^1, ..., W^m\}$, and whose set of edges is defined by the rule*

$$W^i \to W^j \Leftrightarrow \exists X^{i'} \in W^i, \exists X^{j'} \in W^j \ s.t. \ X^{i'} \to X^{j'}.$$

A graphical representation of an example of this definition is shown in Figure 3.5.



(a) Example graph where micro-level nodes are the circles $A, B, ..., F$ and the groups are the dashed ovals.

(b) Conversion of micro-level graph to group-level graph through rules in Definition 3.1.

Figure 3.5: Example of conversion from a micro-level causal DAG to the associated group-level graph.

In [Wahl et al., 2024] there are proposed 3 main methods to perform causal discovery for variable groups. However, [Wahl et al., 2024] does not implement them and thus they can only compare them theoretically. In the following chapters it is explained how we have implemented them, what allowed performing an empirical comparison between them

Table 3.2: Comparison table between theoretical properties of proposed algorithms for causal discovery over groups of time series. The obvious assumption that is obviated are the assumptions of the auxiliar CD (Causal Discovery) algorithm. Algorithms are taken from [Wahl et al., 2024].

Algorithm	Description	Assumptions	Benefits	Drawbacks
Micro-level CD	Find node-level causal DAG and extrapolate to group-level	Group-level causal relations are faithfully represented in micro-level variables	Straightforward application of definition, shouldn't have many false negatives	Computationally expensive, may lead to false positives in high-dimensional groups
Dimension Reduction + CD	Reduce groups to single-instance time series and apply causal discovery on these time series	Causal information of variables isn't lost during reduction	Fast to compute, shouldn't find as many false positives	Relies on a dimensionality reduction, may lead to false positives in high-dimensional groups
Group-Level CD	Apply causal discovery directly on groups of time series	Node-level independencies are represented in groups	Straightforward method	Relies on multivariate conditional independencies, which isn't well settled

and contrast theoretical properties with the ones obtained during experimentation. Their theoretical properties are left in this section.

3.3.1. Micro-level Causal Discovery

Perhaps the most direct way to address causal discovery on groups of variables is to obtain the group-level causal graph from the node-level graph. This is done in 2 steps:

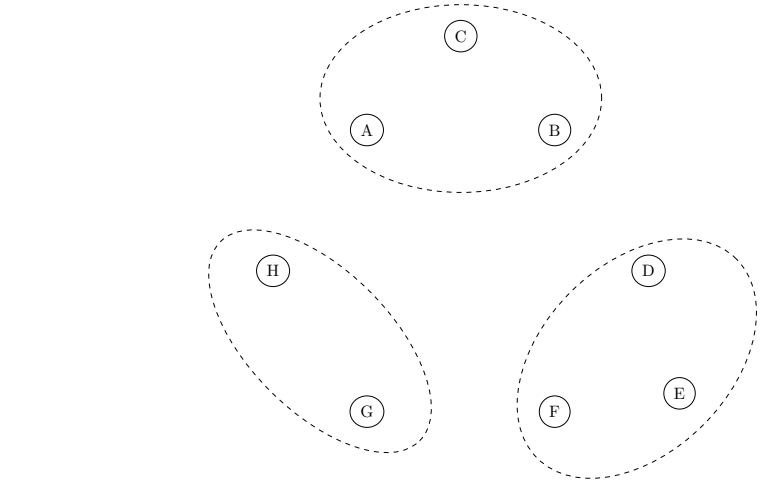
1. Apply a node-level causal discovery algorithm to find the inner causal graph, $\mathcal{G}^{micro}$, with nodes $X_t^1, ...X_t^n, X_{t-1}^1, ..., X_{t-\tau_{max}}^n$. A graphical representation of nodes considered during this step is shown in Figure 3.6b

2. Convert the micro-level causal graph in the group causal graph, with nodes $W_{t-\tau}^i \subseteq \mathcal{V}_{t-\tau}$ by applying the rule

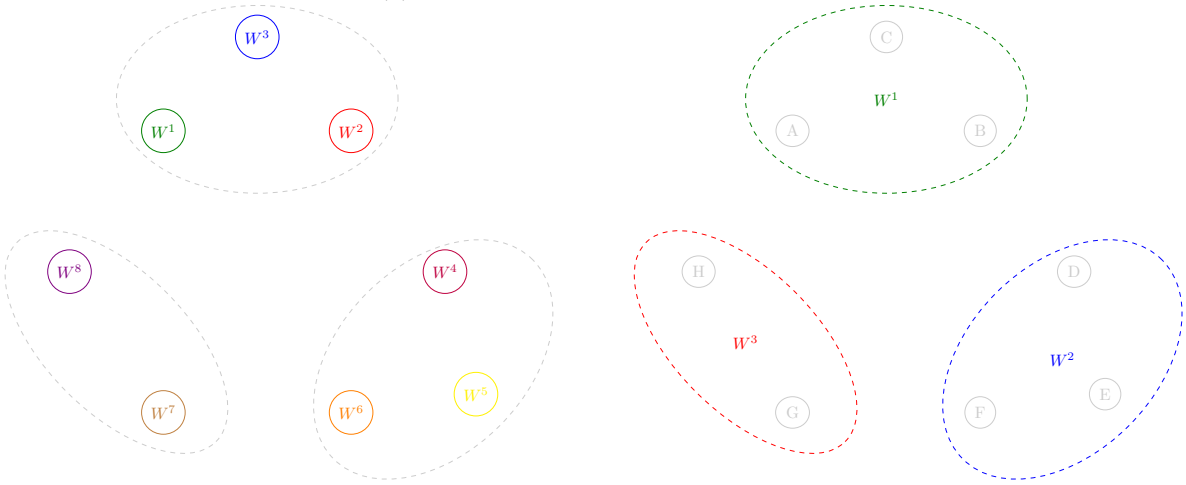
$$W_\tau^i \to W_{\tilde{\tau}}^j \Leftrightarrow \exists X_\tau^{i'} \in W_\tau^i, \exists X_{\tilde{\tau}}^{j'} \in W_{\tilde{\tau}}^j \text{ s.t. } X_\tau^{i'} \to X_{\tilde{\tau}}^{j'}.$$

The main drawbacks of this algorithm are the high computation time it needs and the fact that, when there are many variables in each group W_τ^i, many spurious group causalities may be found, since one single micro-level false positive leads to a group-level

false positive.



(a) Base graph with 8 nodes and 3 groups.



(b) Nodes considered when applying micro-level causal discovery.

(c) Nodes considered when applying dimension reduction + causal discovery.

Figure 3.6: Graphical representation of different group causal discovery algorithms. A dashed ellipse represent the joint of various variables in a group.

3.3.2. Dimension reduction + Causal Discovery

Another quite straightforward approach is to reduce each time series group to a single time series through a dimensionality reduction technique, such as PCA, Appendix A.5, and later apply a node-level causal discovery algorithm on these reduced time series. A graphical representation of this node separation is shown in Figure 3.6c. We name this method after DRCD(Dimension Reduction + Causal Discovery)

This method relieves computational time from the micro-level, and solves the false positive problem, but there appears another problem related with high dimensional groups. The more variables there are in each group, the more diluted the information particular

to each micro-level time series will be, and thus the algorithm will have false negatives.

3.3.3. Group-level Causal Discovery

This method consists on applying classical causal discovery algorithms such as PC, but considering each node as a group of time series. The main problem of this method is the calculation of properties like conditional independencies over groups of variables, since that field hasn't been as researched as much as the single-instance one.

3.4. Direction Extraction or "Vector Causal Inference"

Some studies have carried out experimental comparisons between algorithms for causal inference on groups of time series, but just in pairs, i.e. to check, given 2 sets of variables W^1, W^2, whether $W^1 \to W^2$, $W^1 \leftarrow W^2$, $W^1 \leftrightarrow W^2$, or unknown relation, like [Wahl et al., 2023, Ahmad et al., 2024]. While these algorithms could be used to calculate a group causal graph between various groups of time series, methods they use rely on considering these particular graphs, and use information that creates many spurious relations even in the case that the algorithm works perfectly.

For instance, the algorithm 2G-vecci (two group vector causal inference), [Wahl et al., 2023], with a representation shown in Figure studies whether conditioning on one vector of variables, $\boldsymbol{X}$ deletes connections between the nodes in another set of variables, $\boldsymbol{Y}$ (what would mean that $\boldsymbol{X}$ explains out relations between nodes in $\boldsymbol{Y}$, meaning that $X \to Y$) or creates connections (what would mean that nodes in $\boldsymbol{X}$ retain information that has potentially arrived from different nodes of $\boldsymbol{Y}$, meaning that $X \leftarrow Y$).

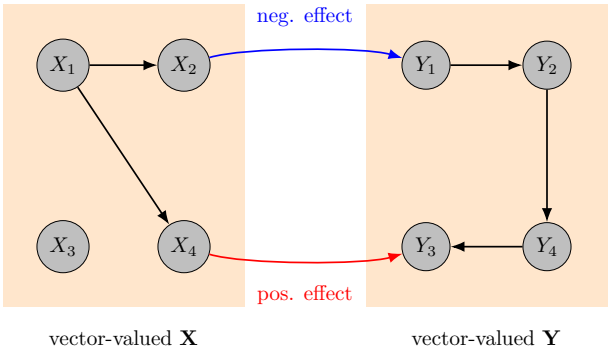


Figure 3.7: Representation of the Vector Causal Inference algorithm 2G-vecci.

This idea clearly fails when there are more than 2 sets of variables, since if we had a

relation $\boldsymbol{X} \to \boldsymbol{Y} \to \boldsymbol{Z}$, even if the algorithm predicted edges perfectly — what it does not — then it would create the spurious edge $\boldsymbol{X} \to \boldsymbol{Z}$. It is also empirically demonstrable that these algorithms extract graphs much less faithful[2] than the ones extracted with methods that we will see in following sections.

[2]With "faithful" we are referring to the metrics that will be explained in Section 5.2. An implementation of this algorithm has been performed to make these empirical comparisons, and can be consulted in our library *group-causation*, Chapter 6, in the module *group_causal_discovery_.direction_extraction*.

4. Problem Statement and Objectives

4.1. Lack of tested algorithms in Group Causal Discovery for Time Series

There are many experimental studies of time series causal discovery algorithms, like [Nouri, 2023, Hasan et al., 2024, Runge, 2020], most of them by generating random time series DAGs, namely *ground truth DAGs*, $\mathcal{G}^{GT}$, sampling data from those DAGs and some assumed causal models and later using these algorithms to infer the causal structure given the data. The output of the algorithm is later compared with the ground truth graph with different metrics, like those we will study in Section 5.2.

However, none of these studies have carried out a complete study that infers a complete causal DAG over groups of time series, as explained in Section 3.3, [Wahl et al., 2024]. Note that this problem is significantly different from the 2 previous ones:

- Causal discovery algorithms for time series assume univariate time series, and their properties as so are used for the construction of the DAGs. Most of these properties, such as conditional independency in constraint-based algorithms or residuals tests in VARLinGAM, are not directly extrapolable to multivariate time series, which are needed in group causal discovery.

- Algorithms that infer a single causal direction in groups of time series can not extract a whole causal graph, since the algorithm is not able to infer an independency between variables $W^1 \perp\!\!\!\perp W^2$, but just the absence of enough information to conclude what's the actual causal direction. Also, these algorithms, because they are focused on the relationships between variables one by one, do not use additional information that may come from relationships between different groups of variables, which has been shown to be crucial in causal inference

[Pearl and Mackenzie, 2018].

For that reason, it is important to contrast different proposals of causal discovery for groups of time series, like those theoretically explained in [Wahl et al., 2024], and to propose new methods particularly designed for this task.

4.2. Proposal of new Approaches for Group Causal Discovery in Time Series

Note that previously explained methods for group causal discovery perform the micro causal discovery method directly on either the micro-level causal DAG, $\mathcal{G}^{micro}$, or on the group causal graph, $\mathcal{G}^{group}$. Our approach consists on mixing the qualities of dimension micro-level and dimension reduction algorithms in such a way that disadvantages in Table 3.2 are leveled out, and the resulting graphs leverage its false positives and negatives.

On the one hand, such an algorithm should not leave as much free space for false positives as micro-level does by allowing to take all relationships between nodes of different groups. On the other hand, it shouldn't compress as much information as dimension reduction algorithm does neither, as it leads to release information that might be useful to find causal structures. In order to balance these problems, we have developed 2 different approaches. Both of them are hybrid approaches, using ideas from *DRCD* and *micro-level* causal discovery, and follow the general idea of, given a set of groups $\{W^1, ..., W^m\} \subseteq \mathcal{P}(\mathcal{V}^{micro})$, extract a new set of groups, $\{\tilde{W}^1, ..., \tilde{W}^r\}$[1], each of them associated with a single time series, in order to perform micro-level causal discovery on this set of nodes to extract $\tilde{\mathcal{G}}^{groups}$, and get G^{group} from it. The pseudocode of this idea is in Algorithm 3, and its variants are implemented in our *group-causation* library, `https://joaquinmateosbarroso.github.io/group-causation/`, in the module *group_causal_discovery.hybrid*.

Subgroups Causal Discovery

A first solution to this problem might be to consider smaller groups, and perform causal discovery from single time series extracted from each of the smaller groups. To do so, given a set of disjunct groups $\{W^i\}_i \subseteq 2^{\mathcal{V}^{\text{micro}}}$, $W^i = \{X^{i_1}, ..., X^{i_r}\} \subseteq \mathcal{V}^{micro}$, that covers $\mathcal{V}^{\text{micro}}$, $\cup_i W^i = \mathcal{V}^{\text{micro}}$, for each $W \in \{W^i\}_i$, this algorithm divides recursively each group

[1] Generally, with $r > m$.

Algorithm 3 General Structure of our Hybrid Causal Discovery methods

Input: Set of groups $\mathcal{V}^{group} = \{W^1, ..., W^m\} \subseteq \mathcal{P}(\mathcal{V}^{micro})$, new groups extraction method $GroupsExtraction$ and Dimensionality Reduction + Causal Discovery Algorithm, $DRCD$.
Output: Predicted group causal graph $\mathcal{G}^{group}$.
function HYBRID CAUSAL DISCOVERY
    $NewGroups \leftarrow \emptyset$
    for all $W^i \in \mathcal{V}^{group}$ **do**
        $ExtractedGroups^i \leftarrow GroupsExtraction(W^i)$
        $NewGroups \leftarrow NewGroups \cup ExtractedGroups^i$
    $\{\tilde{\mathcal{V}}^{group}, \tilde{\mathcal{E}}^{group}\} =: \tilde{\mathcal{G}}^{group} \leftarrow DRCD(NewGroups)$
    $\mathcal{E}^{group} \leftarrow \{W^i \to W^j | \ (\tilde{W} \to \tilde{W}') \in \mathcal{E}^{group}, \ \tilde{W} \in ExtractedGroups^i, \ \tilde{W}' \in ExtractedGroups^j\}$
    return $\mathcal{G}^{group} := \{\mathcal{V}^{group}, \mathcal{E}^{group}\}$

in 2 disjunct subgroups through a *division* function

$$D : \mathcal{P}(\mathcal{V}^{micro}) \ni W \mapsto (\tilde{W}, \tilde{W}') \in \mathcal{P}(\mathcal{V}^{micro}) \times \mathcal{P}(\mathcal{V}^{micro}),$$

such that $W = \tilde{W} \cup \tilde{W}'$, $\tilde{W} \cap \tilde{W}' = \emptyset$. It later applies the same division in $\tilde{W}$ and $\tilde{W}'$, separately, and the algorithm takes as a parameter a fixed threshold, $\tau \in [0, 1]$, of how much *information* of a subgroup should be contained in the single time series in order to be valid. Given a function

$$I : \mathcal{P}(\mathcal{V}^{micro}) \ni W \mapsto I(W) \in [0, 1] \subset \mathbb{R},$$

that indicates how much *information* from the variables $X^{\tilde{i}} \in \tilde{W}$ can be represented by a single time series representing the whole subgroup, each group will be recursively divided until each unique subgroup complies with $I(\tilde{W}) \geq \tau$. The pseudocode of the method is in Algorithm 4, and a graphical representation, for a simpler understanding, is shown in Figure 4.1a.

As a particular case using PCA[2], in order to be relatively certain that we are not losing too much information about the group, during the division, one could create one subgroup with the half of group variables with the highest weight in the first eigenvector,

[2]In the rest of the section we will suppose that variables in W are standardized, in order to simplify calculations.

Algorithm 4 Subgroups Extraction

Input: Group $W \in \mathcal{P}(\mathcal{V}^{micro})$, division function D, information function I, information threshold $\tau \in [0, 1]$.
Output: Set of disjunct subgroups $\{\tilde{W}^1, ..., \tilde{W}^r\} \subseteq \mathcal{P}(\mathcal{V}^{micro})$, with $\cup\{\tilde{W}^1, ..., \tilde{W}^r\} = W$.
function SUBGROUPS EXTRACTION(W)
    if $I(W) \geq \tau$ or $|W| = 1$ **then**
        return $\{W\}$
    else
        $(\tilde{W}, \tilde{W}') \leftarrow D(W)$
        $S_1 \leftarrow$ SUBGROUPS EXTRACTION($\tilde{W}$)
        $S_2 \leftarrow$ SUBGROUPS EXTRACTION($\tilde{W}'$)
        return $S_1 \cup S_2$

$\boldsymbol{v}_1 = \sum_i \boldsymbol{w}_i X^i$, by sorting the weights in descending order $w^{\tilde{i}_1}, ..., \tilde{w}^{\tilde{i}_r}$, and taking

$$D(W) = D\left(\{X^{i_1}, ..., X^{i_{\tilde{r}}}\}\right) := \left(\{X^{\tilde{i}_1}, ..., X^{\tilde{i}_{\lceil r/2 \rceil}}\}, \{X^{\tilde{i}_{\lceil r/2 \rceil + 1}}, ..., X^{\tilde{i}_r}\}\right).$$

Also, a simple and effective information function is the proportion of variance explained by the first eigenvector, which, as explained in Appendix A.5, can be easily calculated as

$$I(W) = I\left(\{X^{i_1}, ..., X^{i_{\tilde{r}}}\}\right) := \frac{\lambda_1}{\sum_j \lambda_{i_j}}.$$

Group Embedding Causal Discovery

A dimensionality reduction algorithm can be applied to extract, from each group with $k_i \in \mathbb{N}$ time series, a set of $k_i' \leq k_i$ time series that try to summarize the data through a *dimensionality reduction* function

$$f_{\text{dim. red.}} : \mathbb{R}^{d \times k_i} \to \mathbb{R}^{d \times k_i'},$$

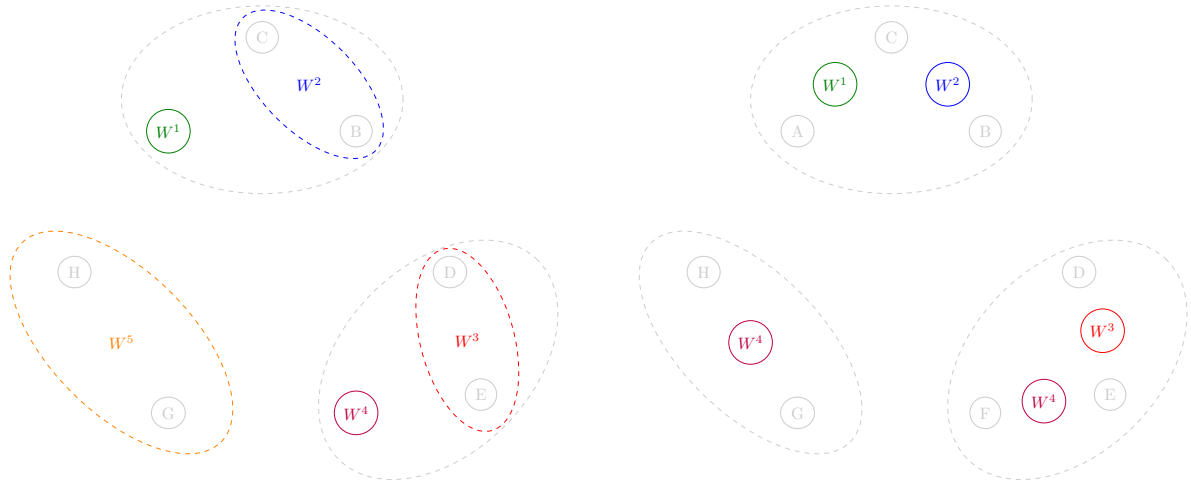
if there are d data samples. If the dimensionality reduction algorithm is consistent with the dependency functions, this should extract important information for causal relations, and thus applying micro-level causal discovery on this dimensionality-reduced set of time series should extract just the significant cause-effect relations. Just as in the previous case, we will use a threshold, $\tau \in [0, 1]$, this time for the amount of information that will be retained from each group. The dimensionality reduction algorithm will manage this

threshold. This algorithm does not need pseudocode, as it is just the application of a dimensionality reduction algorithm on the variables of the group W, with an information threshold of τ. A graphical representation of this approach is shown in Figure 4.1b.

As a particular case, using PCA, one could extract the minimum amount of principal components that represent at least a τ proportion of the total variance, i.e.,

$$W' := \arg\min_{\boldsymbol{V} \in C} |\boldsymbol{V}|, \qquad \text{being } C = \left\{ \{\boldsymbol{v}_1, ..., \boldsymbol{v}_{k'}\} |\ 1 \leq k' \leq k, \frac{\sum_{i=1}^{k'} \lambda_i}{\sum_{i=1}^{k} \lambda_i} \geq \tau \right\},$$

where, with the notation used in Appendix A.5, $\boldsymbol{v}_i$ represents the ith principal component, and λ_i its associated eigenvalue.



(a) Nodes considered when applying subgroups causal discovery.

(b) Nodes considered when applying group embedding causal discovery.

Figure 4.1: Graphical representation of proposed group causal discovery algorithms, taking as basis the graph in Figure 3.6a. A dashed ellipse represents the joint of various variables in a group.

4.3. Need to prefix the groups in Group Causal Discovery

For standard group causal discovery, groups of variables are supposed to be known beforehand. Sometimes this is a plausible assumption; a neurologist studying different parts of the brain may want to know causal relations between different parts of the brain, where each part is compound by a group of time series. Unfortunately, this isn't always the case; if a quantitative analyst is studying the causal relations between different stocks, it would be really interesting if the algorithm was able to extract the division in causal groups that best describes divisions in behavior of stocks. Note that this score must, in general, depend on the context, and that usually, a division created

by experts will be preferred to one optimized by computational methods.

A new score for this purpose will be defined and an empirical study will be performed to check what's the best way to optimize this score inside the search space, between a exhaustive search, a genetic algorithm or a random search.

5. Work Methodology

Except for very particular physical systems, it is not trivial to create a causal model that describes a set of real world data, and relations between them. Many projects have used expert knowledge to extract what are considered to be the ground truth graphs in certain systems, to later use real data from these systems to test causal discovery algorithms [Bojinov et al., 2019, Bernasconi et al., 2024].

However, and specially when treating with time series and high dimensional data, many complex causal relations between variables could appear, and a scientific analysis of the experiment may not be able to recover the whole causal graph. For that reason, it is standard in bibliography to either create synthetic datasets, as in [Pamfil et al., 2020, Assaad et al., 2022], or to use synthetic datasets created by others, with known causal relations, like [Hasan et al., 2024, Moraffah et al., 2021], in order to apply algorithms on this data and compare obtained graphs with the ground truth ones.

In any case, there are no standard datasets obtained from modeling causal relations between groups of time series, so to compare this kind of algorithms it has been necessary to model and implement personalized synthetic data generation methods.

5.1. Synthetic Data Generation

In order to generate feasible structures and data to test causal discovery algorithms 3 steps have to be followed:

1. **Generate a graph**: It is not trivial to generate a uniformly random graph from the set of all Directed Acyclic Graphs, as explained in [Melançon et al., 2001]. However, in general, the followed approach is to generate a DAG with certain randomness (not uniformly), and we have followed the approach in [Runge et al., 2025]. This approach consists on, if we know that the graph has n variables, first sampling a set of lags $\{\tau_1, ..., \tau_L\}$, each of them from $\{0, ..., \tau_{max}\}$, and later taking a uniformly

random subset from the set of all combinations (in case of contemporaneous links) or permutations (for lagged links) of n elements taken in pairs; $\{(i_1, j_1), ..., (i_L, j_L)\}$, being $i_k < i_l$ (or $i_k \leq i_l$ for lagged links) $\forall l, k$. This last condition, as explained in Appendix A.2, forces the resulting graph, with edges $\{X^{i_1}_{\tau_1} \to X^{j_1}_t, ..., X^{i_L}_{t-\tau_L} \to X^{j_L}_t\}$, to be acyclic, so it is a DAG.

In case of generating group causal DAGs, this same algorithms is applied to the group variables $W^1, ..., W^m$, what gives a group DAG $\mathcal{G}^{group}$, where each group arrow $W^i_{t-\tau} \to W^j_t$ is converted in a random set of edges $\mathcal{E}^{i \to j;\tau} \subseteq \{X^{i'}_{t-\tau} \to X^{j'}_t |\ X^{i'} \in W^i,\ X^{j'} \in W^j\}$ (the size of this set is a parameter). Also, inside each group a new DAG $\mathcal{G}^{micro}_i$ is created following the same method. Finally, G^{micro} is created with the following nodes and edges, respectively:

$$\mathcal{V}^{micro} := \left\{X^i_{t-\tau} |\ i \in \{1, ..., n\}, \tau \in \{0, ..., \tau_{max}\}\right\}, \quad \mathcal{E}^{micro} := \bigcup_{i,j,\tau} \mathcal{E}^{i \to j;\tau} \bigcup_i \mathcal{E}^{micro}_i.$$

2. **Extract a causal model from the graph**: The obtained DAG can be directly converted into a Causal Dynamic Bayesian Network, where relations between variables may be defined through structural equations in such a way that variables without parents [1] are random following a certain distribution μ^i_t (for our implementation, it may be gaussian, Weibull or uniform), and the rest of nodes X^j_t will have an additive relation with its parents $Pa(X^j_t)$:

$$X^j_t = \sum_{X^j_{t-\tau} \in Pa(X^i_t)} \alpha^{i \to j;\tau} f^{i \to j;\tau}(X^i_{t-\tau}) \ + \ \mu^i_t,$$

being $f^{i \to j;\tau} : \mathbb{R} \to \mathbb{R}$ the dependency functions; linear, negative exponential, tanh,... And being $\alpha^{i \to j;\tau} \in (0, 1]$ the weights of variables. Weights of autodependency relations $\alpha^{i \to j;1}$ should be higher, as it is a natural conception that a time series should follow a continuous distribution with respect to time.

3. **Sample data from the causal model**: Iterating through timestamps, starting from the beginning, once all variables in time $t-1$ have known values, values for t can be directly sampled using the structural equations.

[1] In time series case, the only variables that may not have parents are the first variables X^i_t, being $t \leq \tau_{max}$, because we will in general assume autocorrelation $X^i_{t-1} \to X^i_t$.

5.2. Metrics to measure fitness of a causal DAG

Let $\mathcal{G}^{gt}$ be the ground truth graph and $\mathcal{G}$ the predicted graph from an algorithm, and $\bar{\mathcal{E}}$ the complementary of $\mathcal{E}$, i.e., all the possible edges that are not in $\mathcal{E}$. We define true positive edges $TP = |\mathcal{E}^{gt} \cap \mathcal{E}|$, false positive edges $FP = |\mathcal{E} - \mathcal{E}^{gt}|$, true negative edges $TN = |\mathcal{E}^{gt} \cap (\mathcal{E}^{full} - \mathcal{E})|$, and false negative edges $FN = |(\mathcal{E}^{full} - \mathcal{E}) - \mathcal{E}|$. Following metrics measure how "well" the algorithm has predicted the structure, or, in other words, how similar is $\mathcal{G}$ to $\mathcal{G}^{gt}$:

- **Precision** $= \frac{TP}{TP+FP}$, measures the confidence one can have in an edge predicted by the algorithm.
- **Recall** $= \frac{TP}{TP+FN}$, measures the ability of the algorithm to discover all edges.
- **F1-score** $= harmonic\ mean(\text{Precision, Recall}) = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision+Recall}}$, allows leveraging both cases by symmetrically representing Precision and Recall in one metric.
- **Structural Hamming Distance (SHD)**, measures the Hamming distance between graphs, defined by the sum of the number of edges that are in $\mathcal{E}$ and not in $\mathcal{E}^{gt}$, and vice versa. In the practice it is implemented by pairwise summing the absolute difference between their adjacency matrices.

5.3. New score to extract a set of groups

For those cases in which there are no predefined groups and no extern information is known, we propose a new score that is later optimized through different methods. This score should comply with 2 main characteristics:

- It should increase with the ability of the first principal component of each group to represent the information about the group. The proportion of variance explained by the first eigenvector, which as explained in Appendix A.5, is easily calculated from the eigenvectors of the correlation matrix, $\xi^j = \frac{\lambda_1^j}{\sum_i \lambda_i^j}$, is a good measure of this, and given that we are interested in not having any particular group with a too low ξ^j, the harmonic mean to obtain what we name after Harmonic Explained Variance (HEV),

$$\bar{\xi} := \frac{\text{number of groups}}{\sum_j 1/\xi^j}.$$

- Given a fixed HEV $\bar{\xi}$, it would be interesting to obtain the lowest number of groups possible, as that would mean that there are more variables in each group that are faithfully represented by a single principal component. On the contrary, a high number of groups will make it easier for the first principal component to represent a high amount of information of these more little groups; in fact, in the upper bound, when the number of groups is equal to the number of variables, $\bar{\xi}$ will trivially be 1. The value $1 - \frac{\text{number of groups}}{\text{number of variables}}$ fulfills this requirement; being higher with less groups, and tending towards 0 when there are too many groups.

Note how a small value of any of these metrics would mean that the partition is deficient, either because of the lack of ability to represent group information, or due to the triviality of the group. The geometric mean allows considering this fact. Such a score should not be considered to be a perfect metric, as can be noted for example by the fact that a partition in the style $\mathcal{P} = \{\{X^i | i \neq i_0\}, \{X^{i_0}\}\}$ would usually be considered better than another $\mathcal{P} = \{\{X^i | i \notin \{i_0, i_1\}\}, \{X^{i_0}, X^{i_1}\}\}$, because the set $\{X^{i_0}\}$ has an HEV of 1, which will be inevitably reduced considerably when another variable is included in the group. Anyways, we consider that in the general case, and when no expert partition is available, the score defined next could determine a relatively faithful partition.

Definition 5.1 (First Component Explainability Score)**.** *Given a set of variables* $\{X^1, ..., X^n\}$ *and partition of it,* $W^j = \{X^{j_1}, ..., X^{j_{n_j}}\}$ *for* $j \in \{1, ..., k\}$*, with* $W^j \cap W^{j'} = \emptyset \; \forall j \neq j'$*, let* ξ^j *be the proportion of variance explained by the first principal component of each group, and* $\bar{\xi} = \frac{k}{\sum_{j=1}^{k} 1/\xi^j}$ *the HEV by the first principal component of each group. Then, the* ***First Component Explainability Score*** *is defined by the geometric mean of* $\bar{\xi}$ *and* $1 - \frac{k}{n}$*.*

$$\textit{First Component Explainability Score} := \sqrt{\bar{\xi} \cdot \left(1 - \frac{k}{n}\right)} = \sqrt{\frac{\bar{\xi}(n-k)}{n}}$$

6. Development and Experimentation

Methods explained in previous section for synthetic data generation, causal discovery in time series and testing with metrics have been implemented in the python library *group-causation*, the code of which can be consulted in
`https://github.com/JoaquinMateosBarroso/group-causation`.
Detailed documentation has been created with the tool *sphinx* and is online in `https://joaquinmateosbarroso.github.io/group-causation`.

6.1. Framework for Time Series Causal Discovery

In Figure 6.1 the different causal discovery algorithms explained in Chapter 3 are shown in a class diagram, including our two new approaches explained in Chapter 4; *GroupEmbeddingGroupCausalDiscovery* and *SubgroupsGroupCausalDiscovery*.

Also, the synthetic data generation process is implemented in the module *create_toy_datasets*, where the main functions are *generate_toy_data* and *generate_group_toy_data*. Implementation details are explained in the documentation.

Finally, the implementation of different metrics and scores explained in Section 5.2 is in the module *utils*.

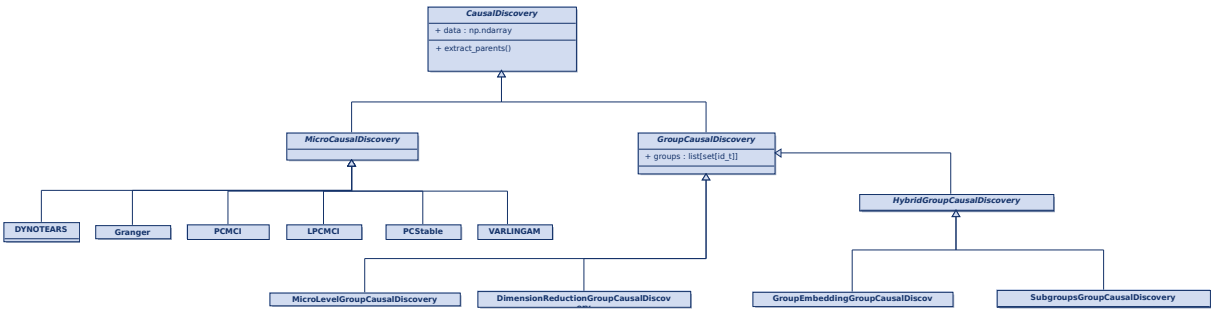


Figure 6.1: Class Diagram of the Causal Discovery structure implemented in the *group-causation* library. Many attributes and methods have been deleted with expository purposes.

6.1.1. Example

Here is a detailed example of the data generation and benchmarking process:

- First, a random DAG is generated, $\mathcal{G}^{gt}$, with edges $\{X^0_{t-1} \to X^0_t, X^1_{t-1} \to X^1_t, X^2_{t-1} \to X^2_t, X^0_{t-1} \to X^1_t, X^1_{t-3} \to X^0_t, X^2_{t-3} \to X^0_t\}$, which is graphically shown in Figure 6.2a. This DAG is associated with a structural causal process that needed some weights parameters. In this case, the process is:

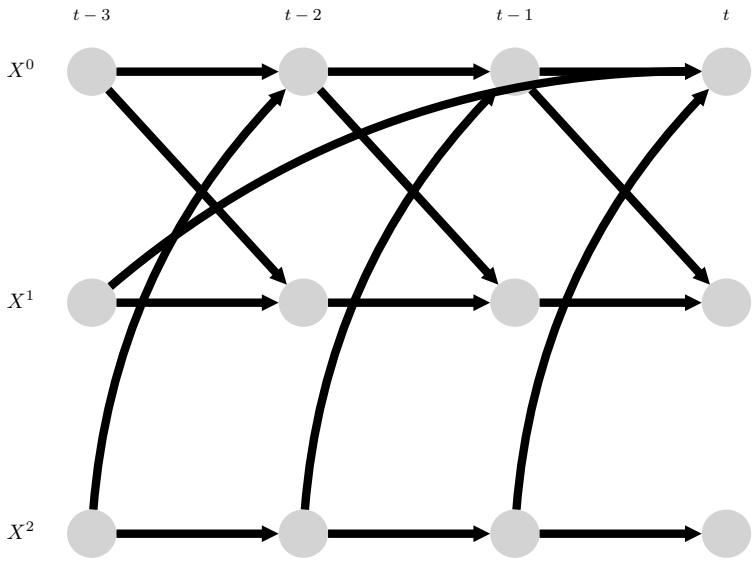
$$
\begin{aligned}
X^0_t &= 0.8\ X^0_{t-1} + 0.7\ X^1_{t-3} + 0.7\ X^2_{t-1} + \mu^0_t \\
X^1_t &= 0.8\ X^1_{t-1} - 0.7\ X^0_{t-1} + \mu^1_t \\
X^2_t &= 0.8\ X^2_{t-1} + \mu^2_t,
\end{aligned}
$$

  being $\mu^i_t \sim \mathcal{N}(0, 0.2)$.

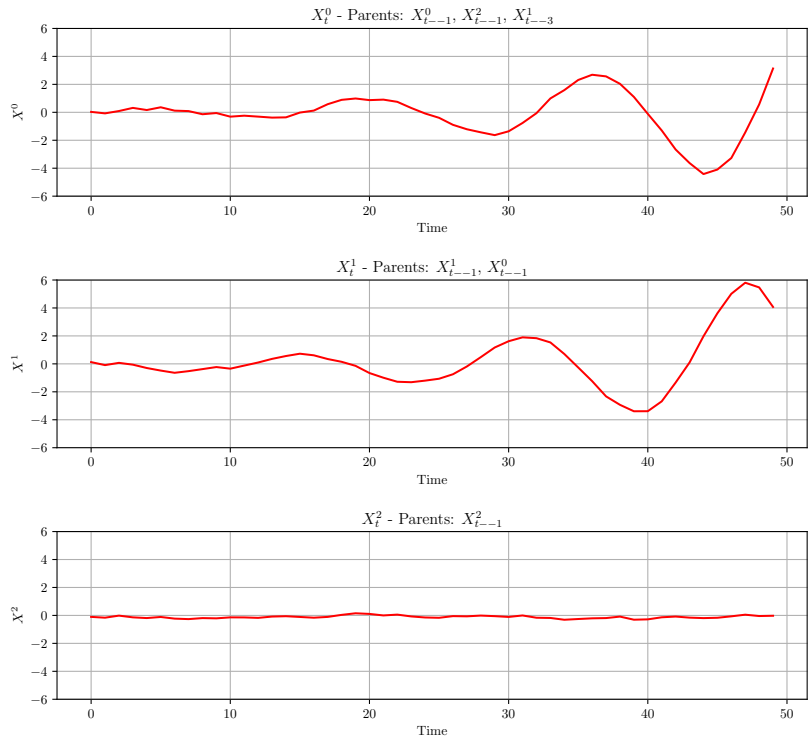
- Second, this DAG is instantiated in a dataset by generating random numbers for the first 2 timestamps, and later applying the rules of the process, obtaining a dataset like the one plotted in Figure 6.2b. Note that plots of X^0_t and X^1_t are very similar. This is because X^0_t "watches" the past of the other variable at X^1_{t-3} and tries to mimic it (due to the positive weight in the structural process), while X^1_t tries to go in the opposite direction of X^0_{t-1} (due to the negative weight), thus creating a periodic relation of ups and downs.

- Then, this dataset is given as input to different algorithms, like pcmci, which obtain a predicted graph, $\mathcal{G}^{pred}$, which, with exhibition purposes, we will suppose to have edges $(\mathcal{G}^{gt} \cup \{X^0_{t-1} \to X^2_t, X^1_{t-2} \to X^0_t\}) - \{X^0_{t-2} \to X^2_t\}$.

- Finally, both graphs are compared to compute false positives and false negatives, and the different studied metrics. In this case we would obtain that the only false positives are $X^0_{t-1} \to X^2_t, X^1_{t-2} \to X^0_t$, and the only false negative is $X^0_{t-2} \to X^2_t$. That way, the precision would be $\frac{TP}{TP+FP} = \frac{5}{5+2} = 0.71$ and the recall $\frac{TP}{TP+FN} = \frac{5}{5+1} = 0.83$, leading to a F1 score of $\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 0.77$. Also, the least steps procedure to recover ground truth graph from the obtained graph is removing the 2 false positives and adding the false negative, so the Structural Hamming Distance is 3.



(a) Randomly generated DAG.



(b) Plot of generated time series.

Figure 6.2: Example of random generation of a DAG and a time series dataset generated from it.

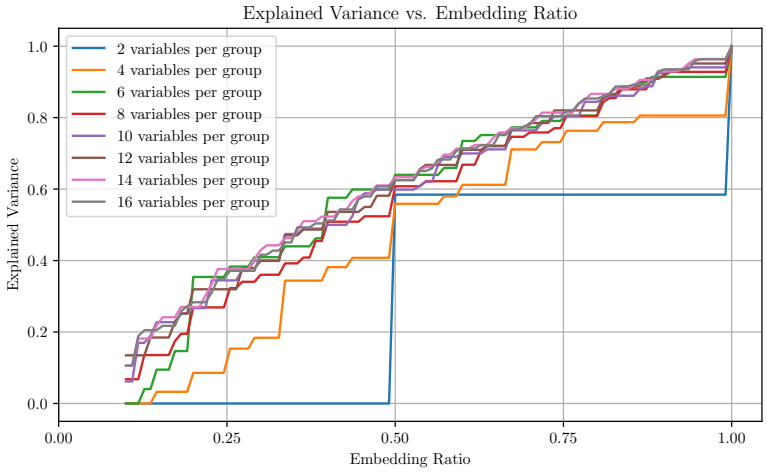
6.1.2. Choosing Variance Threshold

During the development of our group causal discovery methods we needed to specify how much information should be retained in the time series variables that described a whole group. The variance threshold, κ, was introduced as a hyperparameter that sets the amount of information that each new time series should retain, and now it should be studied in order to obtain relevant DAGs.

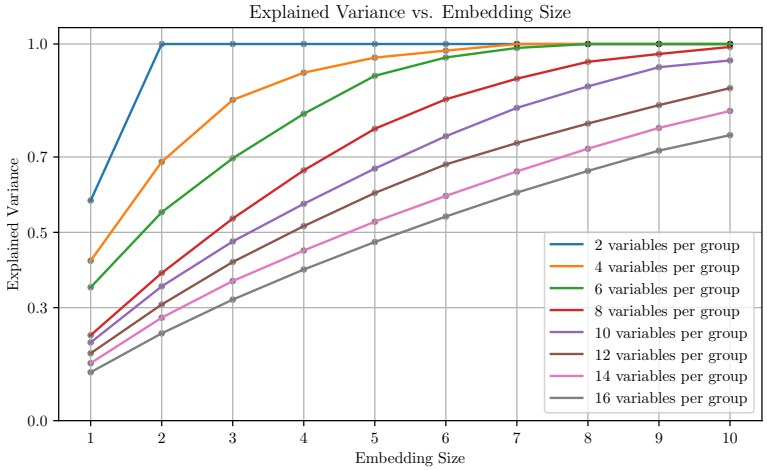
With this purpose, a study was performed using the functions to generate distributions from causal DAGs in the *create_toy_datasets* module of our library *group-causation*, and testing which should be the variance threshold in order to obtain specific embedding ratios (i.e., ratio between the number of variables in the embedding and the original number of variables) and embedding sizes (number of variables in the embedding). Data was created from a time series DAG with 6 groups, a varying number of average variables per group, $T = 2000$ timestamps and linear relations with gaussian noise. Results are shown in Figure 6.3. Some interesting insights can be extracted from them:

- Results of this experiment, in Figure 6.3a shows that, while when groups are little there is a step relation (derived from the fact that if in a group there are, for example, 2 variables, then increasing the embedding ratio from 0.49 to 0.5 allows including 1 variable, thus increasing a lot the variance threshold), for bigger groups there is a monotonous relation between the embedding ratio and the variance threshold, what gives even more sense to the use of this hyperparameter, as it does also preserve a monotonous relation with the fraction of dimensionality of each group that is preserved.

- Figure 6.3b shows another important fact that is very related with the previous observation. As the average number of variables per group grows, the increase of the variance threshold with respect to the embedding size gets smaller. This means that if groups are big, then the same variance threshold that could be used for another dataset with small groups produces embeddings with a higher number of variables. This is interesting, as this is the reason why, by setting a fixed variance threshold, our algorithm will automatically reduce the size of the embeddings and avoid the false positives that big groups impose in the micro-level algorithm.



(a) Variance thresholds obtained when applying different embedding ratios.



(b) Variance thresholds obtained when applying different embedding sizes.

Figure 6.3: Plots where Y axis represents average variance threshold that are needed to obtain the average embedding ratio/size represented in X axis. Each curve represents the results obtained with a specific average number of variables per group[1].

For these reasons, we recommend that for datasets with small groups (from 1 to 6 variables per group) a higher variance threshold is used (between 0.5 and 0.7), allowing to extract embeddings with sizes between 1 and 2.5, as checked by the horizontal grid line at the value 0.6 in 6.3b. It is also recommended that for bigger groups the variance gets reduced, so that groups sizes experience a bigger reduction. For groups with 7-12 variables, the threshold could be between 0.4 and 0.6, thus giving a embedding size of between 2 and 5, while for bigger groups a threshold of around 0.3 retains a high amount of information without increasing exceedingly the group size.

[1]Note that we are not forcing all groups to have this specific number of variables, but that the average group size is the one specified.

6.1.3. Choosing Auxiliar Micro Causal Discovery Algorithms

Once the abstract Group Causal Discovery methods like micro-level or DRCD are explained in Section 3.3 and our proposed approaches are understood in 4.2, in order to apply them to real datasets, particular algorithms need to be used to apply them on the problems once they are reduced to a traditional causal discovery problem.

Choosing these micro causal discovery algorithms is not a simple task, so in this section we will study the best algorithms for this purpose in each group causal discovery method.

- **Micro-level**: The micro causal discovery algorithm used for this method should be able to extract meaningful graphs even in the high dimensionality case, because it will be receive the whole input graph, which will generally be, in the group causal discovery case, quite big. Given the good theoretical properties of PCMCI, explained in [Runge et al., 2019, Runge, 2020]„ its ability to control false positives, to be extended to non-linear relations with simply changing the conditional independence test, to account for autocorrelation, and its particularly good results in the big group case, both in previous surveys, [Assaad et al., 2022, Hasan et al., 2024], and in our particular study of micro time series causal discovery, in Appendix D, we choose pcmci as the algorithm to be applied for micro-level causal discovery. PCMCI effectively controls false positives and accounts for autocorrelations, making it robust in noisy, high-frequency data.

- **DRCD**: This method reduces the number of time series to be explored to the number of groups, that could potentially be between 8 and 12, so an algorithm with this purpose should work correctly with datasets that have a low dimensionality, and it should also be able to correctly surpass hidden confounders, since after reducing the dimensionality so much, the risk of deleting a variable that has meaningful causal relations with variables in a different group arises. Now, for a smaller dimensionality, PCMCI has also shown to be a good and trustful algorithm, as well as DYNOTEARS, which combines temporal modeling with sparsity constraints. They have also shown to be some of the best algorithms for the low dimensionality case in surveys [Assaad et al., 2022] and in our benchmarks in Appendix 3.6b, so they are the algorithms chosen to be applied in DRCD.

- **Hybrids**: For both *group embedding* and *subgroups* methods, the underlying

algorithm should be able to work correctly in a varying dimensionality, since, depending on the distribution of the data in each group, the number of time series that it will receive could be any integer between the number of groups and the number of variables. For that reason, following previous results, and considering the very positive theoretical results of PCMCI, it will be used for the hybrid algorithms.

6.2. Groups Extraction

Even under the assumption that a score like the one explained in Definition 5.1 is able to represent the affinity of the division in groups, the optimum over the search space should be obtained, where the search space is the family of all partitions of the variables. This space has been studied in mathematics for more than a century, and researches like [Bell, 1934] extract meaningful results from them. One of the most important is the recursive formula to obtain the size of this space, which is

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k,$$

knowing that $B_0 = 1$. That way one can compute easily the first Bell numbers, which are shown in Table 6.1. The rapid increase of this space makes it unfeasible to explore it exhaustively even for a number of variables much smaller than the ones we will study. For example, $B_{20} = 5.172 \cdot 10^{13}$, and $B_{100} = 4.759 \cdot 10^{115}$; a quantity higher than the number of atoms in the observable universe.

B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}
1	1	2	5	15	52	203	877	4 140	2 1147	115 975	678 570	4 213 597	27 644 437

Table 6.1: First Bell numbers.

This is the reason why in our library *group-causation* the module *groups-extraction* has three different groups extractors in the form of classes:

- **Exhaustive Groups Extractor** performs an exhaustive search, iterating over the whole space of all partitions of n variables, and returns the partition with the optimum value.

- **Random Groups Extractor** generates randomly a number of partitions equal to the total number of partitions tested in the genetic algorithm[2] explained next.

- **Genetic Groups Extractor** uses a simple genetic algorithm to optimize the partition of the variables according to the specified function (in our case, the function will be our First Component Explainability Score). The initial population size is taken as $\min\{100, B_{\lfloor n/2 \rfloor}\}$, in such a way that for a low number of variables the population is smaller than for the exhaustive approach, and for bigger datasets, the complexity of the algorithm doesn't get too big. Also, typical values of mutation probability = 0.2, mating probability = 0.5 and 50 generations are used. The crossover procedure is shown in Algorithm 5, and the simple mutation is in Algorithm 6. As selection algorithm there has been used the classical tournament selection algorithm, which randomly selects a subset of the population (in our case, with size 3), and the fittest individual from this group is selected as a parent.

Algorithm 5 Crossover Between Two Partitions

Input:
Two partitions $\mathcal{P}_1$ and $\mathcal{P}_2$ of the same variables $X^1, ..., X^n$
Output: Two new partitions $\mathcal{O}_1$ and $\mathcal{O}_2$ derived from crossover
Initialize empty element-to-group maps M_1 and M_2
for each subset S in $\mathcal{P}_1 \cup \mathcal{P}_2$ **do**
    if random coin flip is heads **then**
        for each element $e \in S$ **do**
            Assign e to S in M_1
    else
        for each element $e \in S$ **do**
            Assign e to S in M_2
for each element $e \in \mathcal{U}$ **do**
    if e not assigned in M_1 **then**
        Assign e to $\{e\}$ in M_1
    if e not assigned in M_2 **then**
        Assign e to $\{e\}$ in M_2
Reconstruct $\mathcal{O}_1$ from values in M_1
Reconstruct $\mathcal{O}_2$ from values in M_2
Return $\mathcal{O}_1$, $\mathcal{O}_2$

[2]Genetic algorithms are stochastic search metaheuristics inspired by natural selection, commonly used for solving combinatorial optimization problems by iteratively evolving a population of candidate solutions through selection, crossover, and mutation.

Algorithm 6 Mutation of a Partition

Input: A partition $\mathcal{P}$ of the variables $X^1, ..., X^n$
Output: A mutated partition $\mathcal{P}'$
if $|\mathcal{P}| > 1$ **then**
    Randomly select two distinct indices i and j
    if $\mathcal{P}_i \neq \emptyset$ **then**
        Randomly select an element $e \in \mathcal{P}_i$
        Remove e from $\mathcal{P}_i$ and add it to $\mathcal{P}_j$
Return $\mathcal{P}$

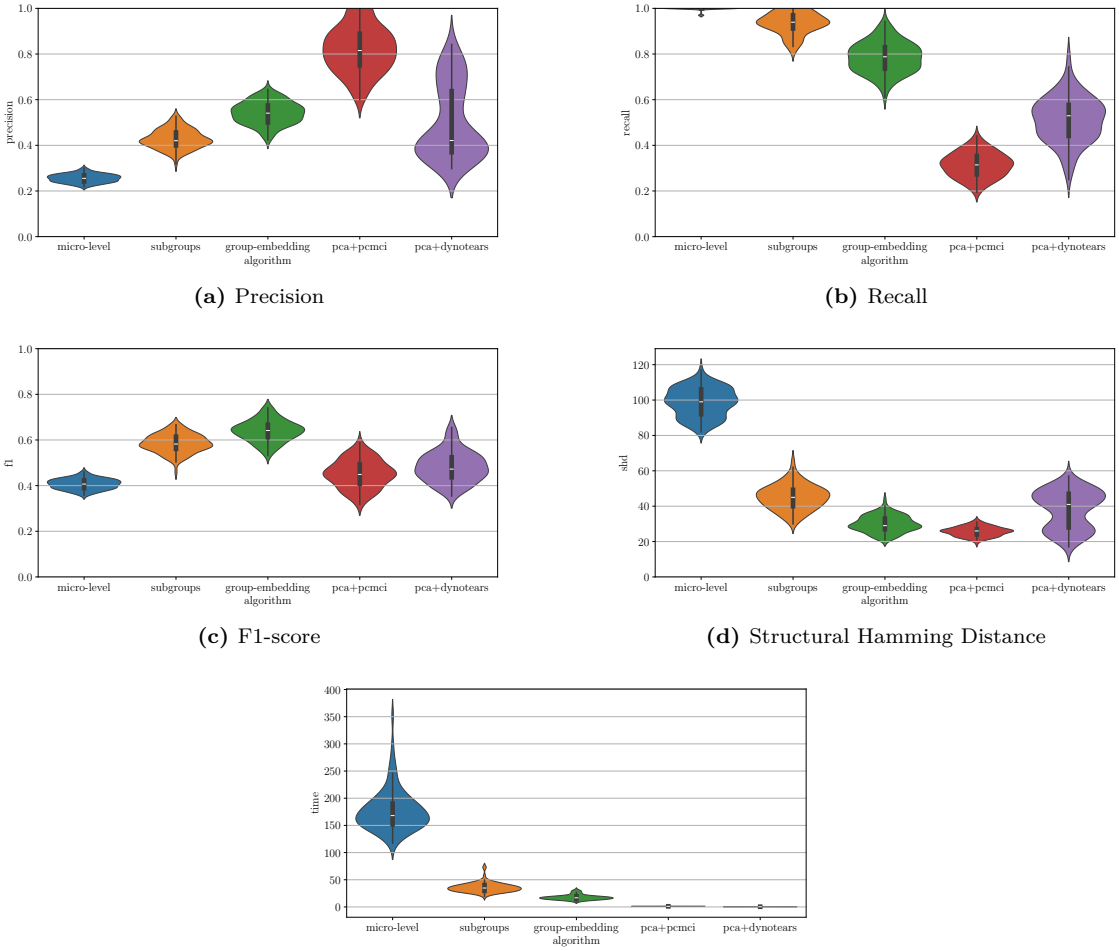
7. Results and Discussion

7.1. Group Time Series Causal Discovery Results

An experiment was performed to study metrics obtained from the execution of different causal discovery methods for groups of time series in 100 different synthetic datasets. Datasets have been generated synthetically through methods explained in Section 2.3, each of them with a different base graph structure. Variables have linear relations and noise is $\mu_i^t \sim \mathcal{N}(0, 0.2)$. The inner- and outer-group crosslinks density have been fixed to 0.5, autocorrelation coefficients are 0.4 and dependency coefficients are randomly chosen in $\{-0.3, 0.3\}$. Node-level CD algorithms are using the recommended hyperparameters in original papers. Hybrid approaches follow the parameters recommendations in Chapter 4. There have been used 6 groups and 60 variables, what leads to *N_vars_per_group*= 10. is shown in Figure 7.1. There, the group-embedding approach is near the best algorithms in both precision and recall, but reaches the best F1-score, while maintaining a conservative time consumption.

In order to contrast whether these affirmations are statistically significant we have applied a Friedman test to the different metrics, obtaining the ranks in Table 7.1 and the results of the test, using a significance level of $\alpha = 0.05$, are shown in 7.2. These results allow rejecting the null hypotheses of the test, which is that there is no difference in the distributions of the metrics across the algorithms. That way, it is appropriate to perform a post-hoc test to determine which pairs of algorithms differ in each metric. In particular, we have applied a Nemenyi test with significance level $\alpha = 0.05$, and its results are shown in Figure 7.2.

As expected, the DRCD algorithm pca+pcmci reaches the highest precision, with significant differences with the rest of algorithms. This happens because it does not include an edge unless its effects can be observed even after reducing a lot the dimensionality.



(a) Precision

(b) Recall

(c) F1-score

(d) Structural Hamming Distance

(e) Time (seconds)

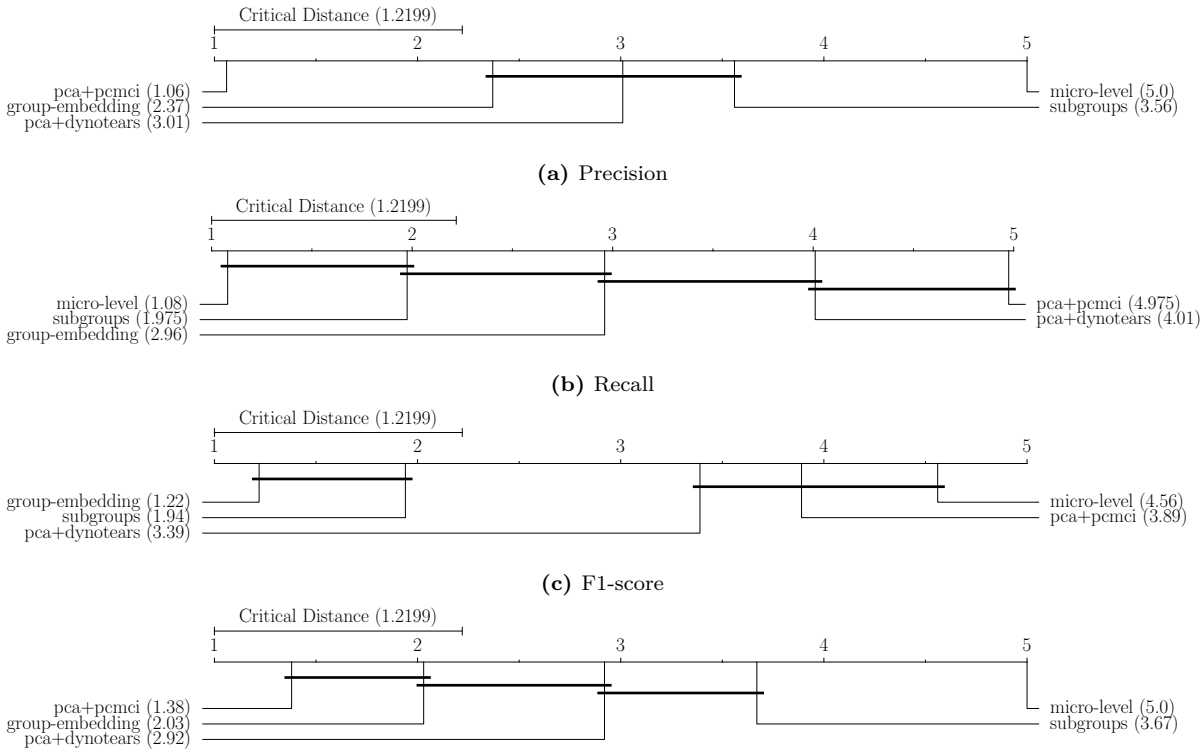
Figure 7.1: Violin plots with results of the static experiment in the high dimensionality case. Boxplots are shown in inner black boxes and in color there are shown the aproximated distributions of the metrics.

Algorithm	$\text{Rank}_{\text{Precision}}$	$\text{Rank}_{\text{Recall}}$	Rank_{F1}	Rank_{SHD}
Micro-level	5.00	1.08	4.56	5.00
Subgroups	2.40	1.98	1.94	3.67
Group-embedding	2.37	2.96	1.22	2.03
pca+pcmci	1.06	4.98	3.89	1.38
pca+dynotears	3.01	4.01	3.39	2.92

Table 7.1: Average ranks obtained in each algorithm for each metric of those tested in 7.1.

Metric	Statistic	Critical value	p-value
Precision	338.97	9.49	$1.62 \cdot 10^{-10}$
Recall	386.37	9.49	$1.80 \cdot 10^{-10}$
F1	306.80	9.49	$1.56 \cdot 10^{-10}$
SHD	320.82	9.49	$1.42 \cdot 10^{-10}$

Table 7.2: Graphs with results obtained after applying a post-hoc Nemenyi test to determine which pairs of algorithms differ in each metric between those studied in Figure 7.1.



(a) Precision

(b) Recall

(c) F1-score

(d) Structural Hamming Distance

Figure 7.2: Nemenyi test ranks values for the high dimensional experiment.

While this may be positive in some cases, it also forces to discover just a low number of associations, as can be extracted from the fact that pca+pcmci reaches a significantly lower recall than the one of all the algorithms using different techniques (just the other DRCD algorithm doesn't have significant differences with it).

Also, the micro-level algorithm reaches the best recall, showing significant difference with all the other algorithms except for the subgroups algorithm. This happens because, as explained in Section 3.3, it tends to extract a high number of edges, even when there's not much confidence on it because a single edge from one group to another is enough to extract a group link.

Finally, our group-embedding algorithm, while not being the best in precision nor recall, is able to reach a balance between them and reaches the best F1-score, showing significant differences with all the other methods except for our other hybrid algorithm, subgroups. It could be expected that the group-embedding approach reaches better results than the subgroups, as while the subgroups follows a more interpretable approach – by dividing the problem of estimating edges between groups to estimating them in subsets of that group – the group-embedding approach retains the highest possible amount of

variance for the used dimensionality, and it is precisely this variance what allows extracting dependencies.

Also, this best approach of group-embedding, while losing in SHD, does not reach significant differences with the best algorithms in this metric. This is a significantly more important observation noting that one could expect that an algorithm that tends to predict a low number of edges, like those of the type DRCD, reaches a small SHD, since this metric is upper bounded by $|\mathcal{G}^{gt}| + |\mathcal{G}^{pred}|$.

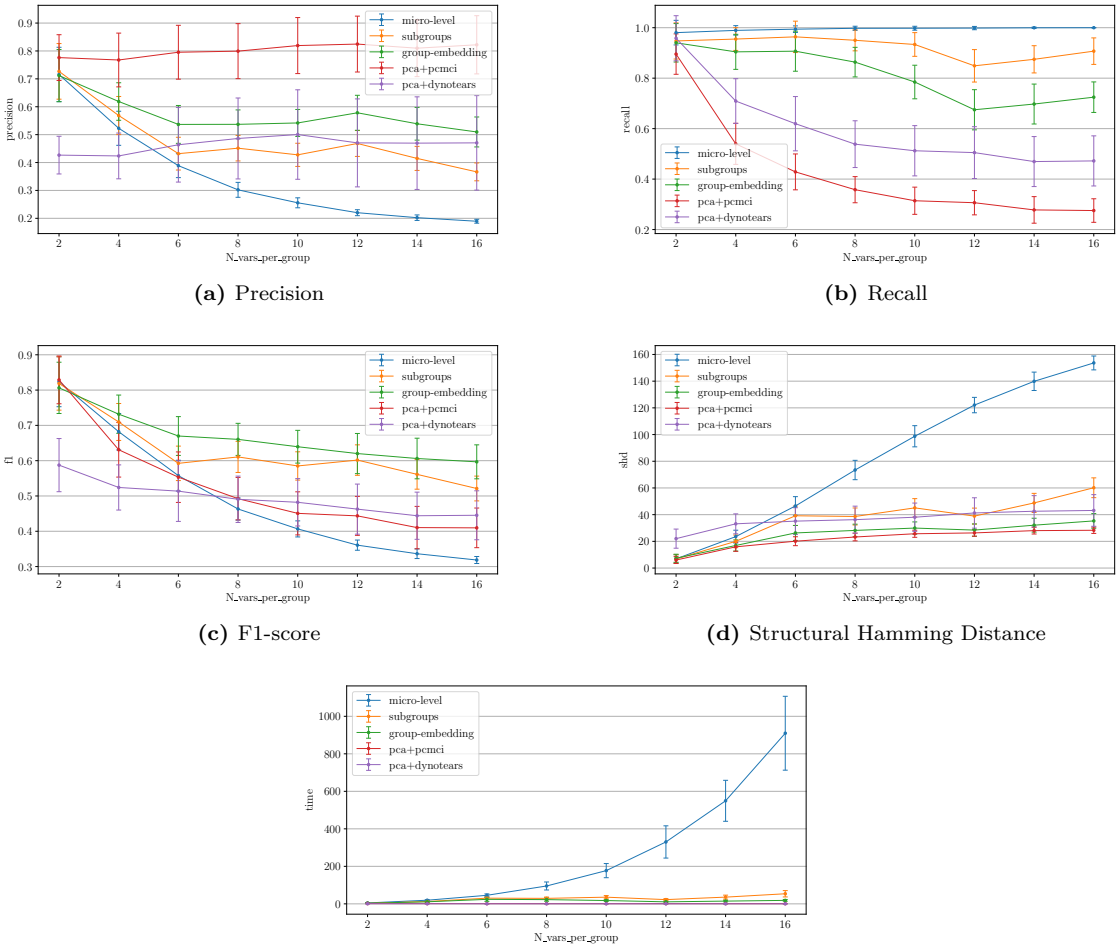
An experiment using the same variable relations and noise, with a fixed number of 6 groups and an increasing number of variables per group, from 2 to 16, and 100 different graph and dataset generations per point has its results plotted in Figure 7.3.

There, theoretical properties of micro-level CD and DRCD are experimentally contrasted, as it is demonstrated by the fact that *micro-level* finds more false positives (lower precision) when the groups dimensionality increases and the dimensionality reduction + CD algorithms, *pca+pcmci* and *pca+dynotears*, both lead to a high quantity of false negatives (low recall) in the high dimensionality case. Also, it is clear in the time graph that *micro-level* algorithms have a much higher computational cost.

On the other hand, the proposed *group-embedding* and *subgroups* approaches, while not winning in precision nor recall[1], maintain a high and stable trend in both metrics, and reaches the F1-score maximum in the high dimensionality case. Also, it is able to overcome computational complexity of micro-level.

Similar conclusions can be extracted from the graphs obtained after applying the same experiment with respect to an increasing number of groups and a fixed number of 5 variables per group (in this case with 25 executions per point, due to the increase of computational complexity with a high number of groups), shown in Figure 7.4. pca+pcmci is still the method with highest precision, followed by our hybrid approaches, and in the F1-score, all the other methods experiment a rapid decline in this metric. This makes sense, as having more groups will generate more spurious relations, to which micro-level is quite sensible, and the dynotears algorithm is less stable than pcmci. On the other hand, pca+pcmci doesn't experiment this decrease, probably due to an easy prediction

[1]This was obvious, since hybrid methods won't have the same predictive power as micro-level nor the same certainty as DRCD.



(a) Precision

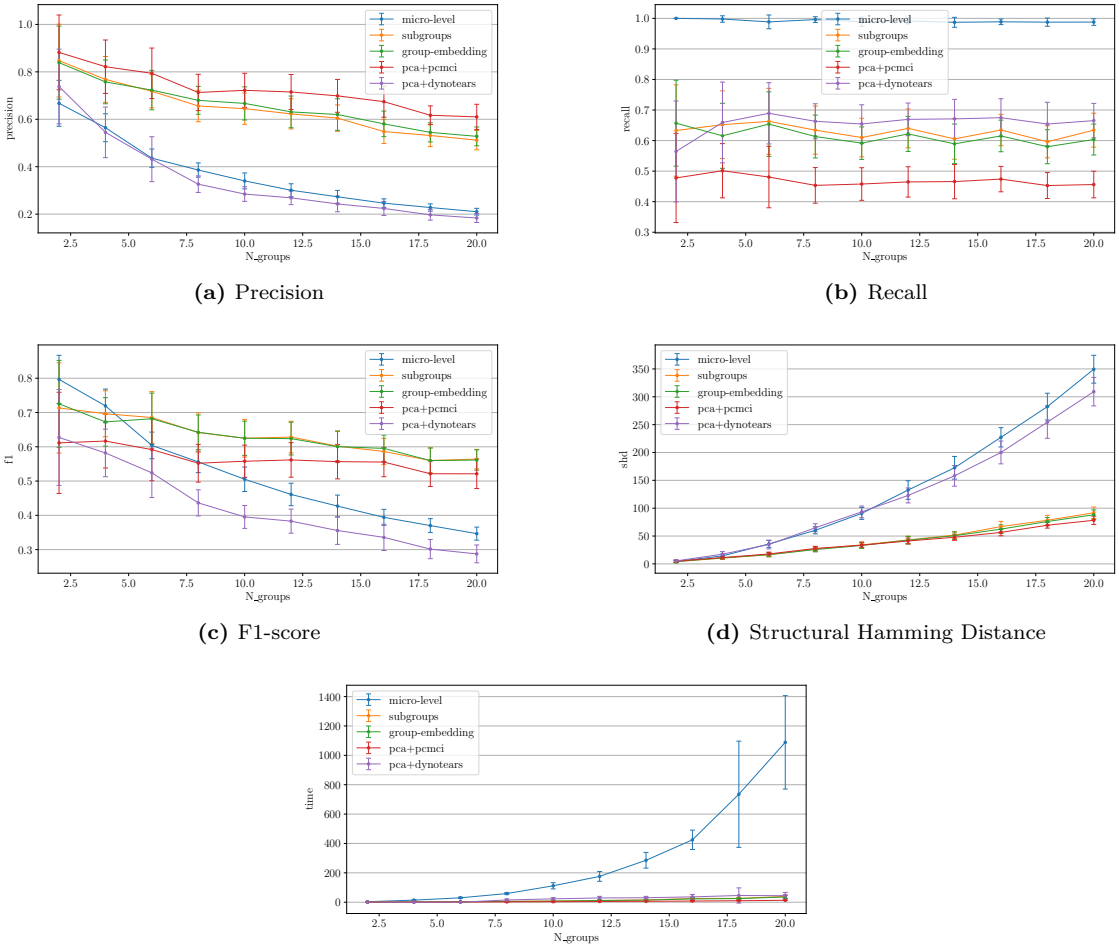
(b) Recall

(c) F1-score

(d) Structural Hamming Distance

(e) Time (seconds)

Figure 7.3: Plots evaluation metrics obtained from executions of different causal discovery methods for groups of time series as the average number of variables per group increases. Each point is the average over 100 iterations and the $\pm std$ interval is shown.



(a) Precision

(b) Recall

(c) F1-score

(d) Structural Hamming Distance

(e) Time (seconds)

Figure 7.4: Plots with evaluation metrics obtained from executions of different causal discovery methods for groups of time series as the number of groups increases. Each point is the average over 25 iterations and the $\pm std$ interval is shown.

of the relations of a relatively little group – 5 variables per group, compared with the 16 used in the previous experiments – using its first principal component.

Also, one can note how in this case the SHD between hybrid methods and pcmci is shortened, probably due to the fact that by keeping a conservative dimensionality in the first step of the hybrids algorithms, less spurious relations are found, leading to a lower number of false positives – as it can be seen in the recall plot – and thus the differences between the obtained graph and the ground truth are more based on false negatives, what, as explained in Section 3.3, leads to a lower SHD.

Once our algorithm has been compared wit the others, it would be interesting to contrast how different metrics vary depending on the hyperparameter that our algorithm uses to choose subgroups of variables, or principal components; threshold for explained

variance ratio, κ. A study was done by generating 100 different graphs, and from them the datasets. Results of this study are shown in Figure 7.5. Experiments are performed for each possible threshold in explained variance ratio, κ, in $\{0.05, 0.1, 0.15, ..., 0.95, 1\}$. There are vertical segments indicating the beginning and the end of the recommendation interval for the parameter κ, which is $[0.4, 0.6]$. The rest of algorithms are plotted with a surrounding ellipse whose radius are standard deviation of metrics in each axis. Some theoretical properties can be experimentally confirmed by studying these graphics:

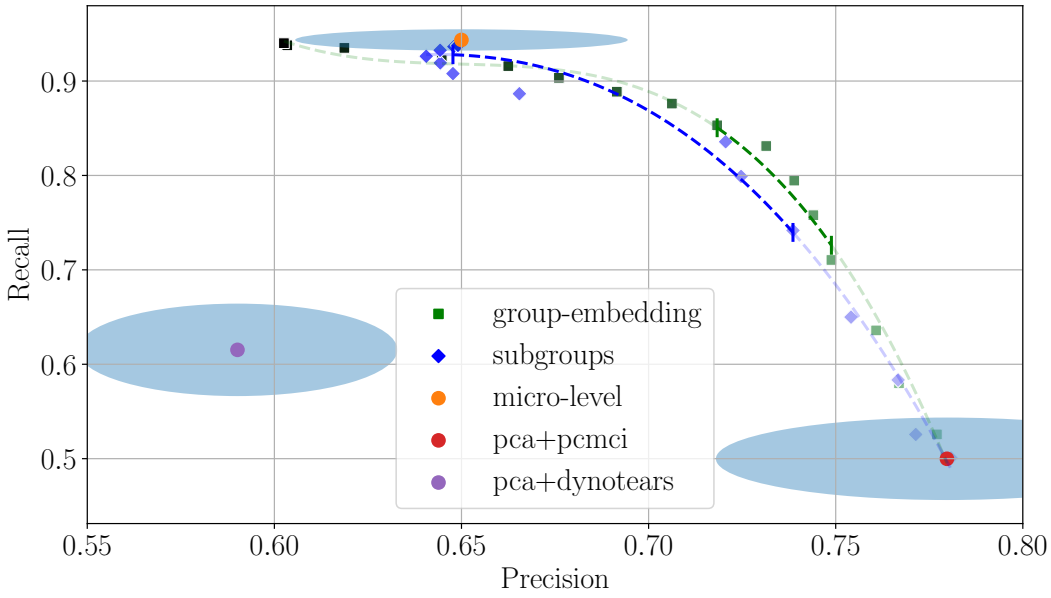
- Both hybrid algorithms are asymptotically equivalent to pca+pcmci when $\kappa \to 0$. This is a direct consequence of the fact that a low κ means that the subgroups algorithm will be very liberal with the groups selection, and many big groups will be converted single timeseries. Also, the algorithm based on group embedding will consider that one principal component is enough to describe the whole group. In both cases, data used for the inner micro-level causal discovery algorithm is quite similar to the one that pca+pcmci uses.

- In both graphics, the hybrid algorithm based in subgroups is asymptotically equivalent to the micro-level algorithm when $\kappa \to 1$, what is completely consistent with the design of the algorithm; a high κ means that the algorithm will divide the groups in more subgroups, until each subgroup has one single time series, in the case $\kappa = 1$.

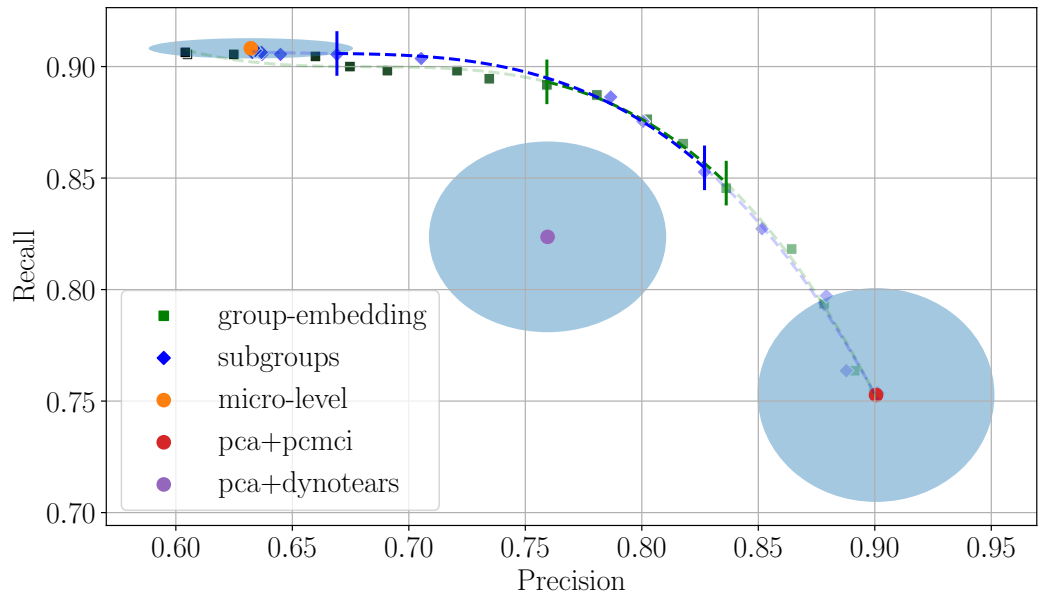
- The group embedding approach is more stable with respect to the increase of κ. This is easily seen, by checking that in Figure 7.5a, the group embedding algorithm has its points relatively uniformly distributed along the Pareto front, while the subgroups approach has many points at the edges of the front, and not as many in the middle areas. This is what implies that the recommended interval is smaller for the group embedding approach, what should give a better stability of the results of the algorithm in different datasets.

- The recommended interval for the κ parameter reaches a tradeoff between precision and recall, thus giving a stable algorithm that counters the drawbacks of micro-level and dimensionality reduction algorithms.

- Figure 7.5b shows that hybrid approaches are able to reach a recall similar to the one that micro-level algorithms reach, but without losing as much precision. This



(a) Metrics obtained from Window Causal DAG.



(b) Metrics obtained from Extended Summary Causal DAG, Definition 2.12.

Figure 7.5: Pareto front showing how recall and precision vary with different values of κ. The darker the point is, the greater κ is used, between 0 and 1.

would be very interesting in case one just wanted to extract the extended summary causal DAG, and wanted to have a high probability of finding all causal relations (having a high recall), what micro-level algorithms are able to give, but he didn't want to have such a low precision as this approach has. By using a hybrid approach with a high κ; between 0.5 and 0.8, that objective could be met.

- Both pca+pcmci and pca+dynotears reach better results in both metrics when they are obtained from the Extended Summary Causal graph. This makes sense, because these algorithms summarize as much information as possible, thus making it simpler to extract relations in the kind of the ones shown in Figure 2.5b. However, those algorithms that do not summarize the data, like micro-level, were, already in the Window Causal DAG case, able to extract most relationships (recall of 0.95), so summarizing the graph simply forces having a similar number of false positives, but a lower number of true positives, because relationships that were previously studied as $X^i_{t-\tau} \to X^j_t$ and $X^i_{t-\tau'} \to X^j_t$, now become $X^i_{t-} \to X^j_t$.

7.2. Real datasets

In this section, we will use a two different standard datasets that either are real data from a process with a known causal structure, where edges are labeled by domain experts, or have been generated through a simulation of a real life process in which the ground truth causal graph is known by definition of the model. In the cases in which no grouping of variables is given by the provider of the dataset, a previous group extraction will be performed, following the genetic approach.

7.2.1. Tennesee Eastman dataset

Also explained in [Menegozzo et al., 2022]. It is a simulated dataset from a well-known chemical process benchmark, originally designed to test control algorithms for industrial plants. It consists of multivariate time series data from various sensors and process variables under different operating modes and disturbances.

It's useful for our case because it has a complex group structure, where variables naturally group into subsystems (e.g., reactant flows, pressures, temperatures), making it ideal for evaluating methods that work with groups rather than individual time series.

Another important feature of the dataset is its high dimensionality; it has 31 variables, what, even though being smaller than our synthetic tests, still is big enough to allow the experimentation in groups of variables. These variables have been divided, through the genetic optimization of the first component explainability score, in 5 groups: [{1, 2, 3, 7, 18}, {0, 4, 5, 6, 8, 9}, {10, 11, 12, 13, 14}, {15, 16, 20, 21, 25, 26, 27, 28, 29, 30}, {17, 19, 22, 23, 24}]. The time series graph induced by this grouping of variables, after applying the rule in Definition 3.1, is the one shown in Figure 7.6.

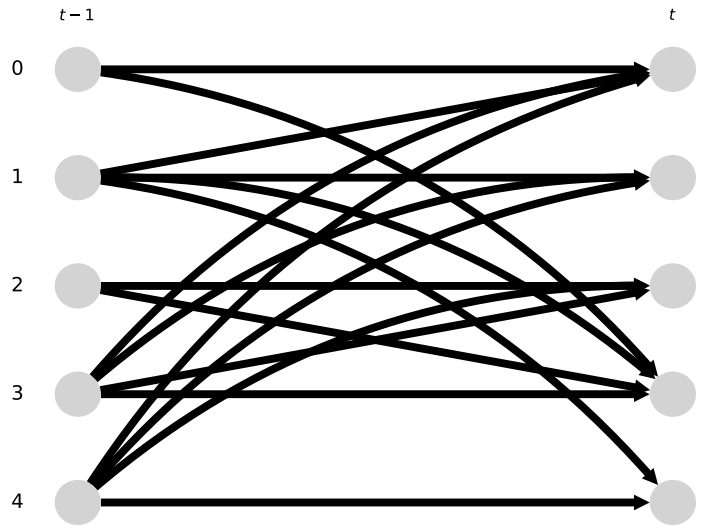


Figure 7.6: Time Series DAG induced by the grouping of the dataset Tennesee Eastman.

After applying on this data the studied group causal discovery methods, results are shown in Table 7.3[2]. These metrics are derived from the following mistakes, divided by algorithm:

1. micro-level algorithm returns 9 false positives: $\{X^2_{t-1} \to X^0_t,\ X^2_{t-1} \to X^1_t,\ X^0_{t-1} \to X^1_t,\ X^1_{t-1} \to X^2_t,\ X^0_{t-1} \to X^2_t,\ X^4_{t-1} \to X^3_t,\ X^2_{t-1} \to X^4_t,\ X^3_{t-1} \to X^4_t,\ X^0_{t-1} \to X^4_t\}$ and 2 false negatives: $\{X^1_{t-1} \to X^0_t,\ X^1_{t-1} \to X^4_t\}$. While there are few false negatives, the high amount of false positives wouldn't allow to trust this algorithm in a real case, since for further causal inference methods or logical conclusions we could be relying on a relation that doesn't exist in reality.

[2]Metrics are obtained in the summary causal graph, since original ground truth graph doesn't discriminate by relation lag.

Algorithm	Precision	Recall	F1	SHD	Time (s)
micro-level	0.609	0.875	0.718	11	41.838
pca+pcmci	0.636	0.438	0.519	13	**2.345**
pca+dynotears	0.000	0.000	0.000	31	165.253
subgroups	0.700	0.875	0.778	8	17.198
group-embedding	**0.789**	**0.938**	**0.857**	**5**	10.872

Table 7.3: Results obtained from applying different group causal discovery algorithms to the Tennesee Eastman dataset.

2. pca+pcmci extracts "just" 4 false negatives: $\{X^2_{t-1} \to X^0_t,\ X^2_{t-1} \to X^1_t,\ X^4_{t-1} \to X^3_t,\ X^3_{t-1} \to X^4_t\}$, but 9 false positives: $\{X^0_{t-1} \to X^0_t,\ X^3_{t-1} \to X^0_t,\ X^1_{t-1} \to X^0_t,\ X^4_{t-1} \to X^1_t,\ X^3_{t-1} \to X^1_t,\ X^3_{t-1} \to X^2_t,\ X^4_{t-1} \to X^2_t,\ X^0_{t-1} \to X^3_t,\ X^1_{t-1} \to X^4_t\}$, what means that, again, this algorithm doesn't return completely faithful graphs, because many causal inference methods need to have found all the possible confounders of a variable in order to calculate counterfactuals or interventions, and in this case we could be missing a lot of confounders.

3. pca+dynotears a big problem derived from the dynotears algorithm; since dependencies arent as strong as with synthetic datasets, the term related with dependencies in the dynotears optimization formulation wasnt able to overcome the regularization factor related with the acyclity, leading to an empty graph.

4. subgroups method leads to 6 false positives: $\{X^2_{t-1} \to X^1_t,\ X^0_{t-1} \to X^1_t,\ X^1_{t-1} \to X^2_t,\ X^4_{t-1} \to X^3_t,\ X^2_{t-1} \to X^4_t,\ X^3_{t-1} \to X^4_t\}$ and just 2 false negatives: $\{X^4_{t-1} \to X^2_t,\ X^0_{t-1} \to X^3_t\}$. Note how all the false positives that this algorithm obtains are also false positives for the micro-level algorithm, what means that the subgroups method was able to reduce the number of false positives without influencing to the size of the false negatives group. The other way around, both false positives are also false positives for the pca+pcmci algorithm, meaning that this algorithm reduced the number of false positives while also reducing the false negatives.

5. group-embedding obtains 4 false positives: $\{X^2_{t-1} \to X^1_t,\ X^4_{t-1} \to X^3_t, X^3_{t-1} \to X^4_t, X^0_{t-1} \to X^4_t\}$ and one single false negative: $\{X^1_{t-1} \to X^0_t\}$, showing its ability to mix the discovery strength of micro-level with the more conservative approach of pca+pcmci.

7.2.2. Ultra Processed Food dataset

Explained in [Menegozzo et al., 2022]. A real-world dataset derived from nutritional, chemical, and ingredient features of ultra-processed foods. It includes time-dependent or hierarchically structured data across product categories. The fact of being a real dataset, despite giving less confidence in the ground truth causal graph, allows testing the algorithms in the same context that it is aimed at being applied. This dataset has 17 different variables and 23 132 time stamps, what allows testing for the speed of our algorithms in the case with a high number of samples. Note that all of studied algorithms have a linear computational complexity, $\mathcal{O}(n)$, with respect to the number of samples.

First, since there isn't a default set of groups, the genetic algorithm for the optimization of the first component explainability score was executed, returning the following set of groups: [{1, 3, 4, 5}, {6, 8, 9, 10, 11}, {12, 13, 14, 15}, {16, 0, 7, 2}], which, after the extraction of group-level relations through the rule in Definition 3.1, induces the graph shown in Figure 7.7.

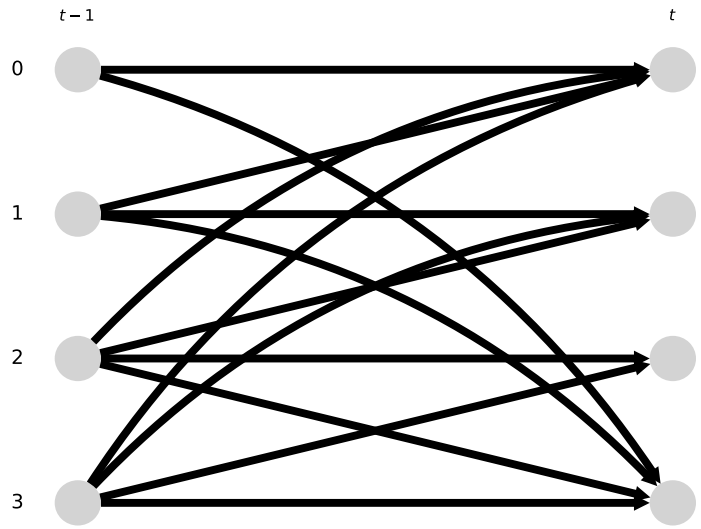


Figure 7.7: Time Series DAG induced by the grouping of the dataset Tennesee Eastman.

Results for this dataset are shown in Table 7.4 are obtained[3]. In this dataset, micro-

[3]These metrics are obtained in the summary causal graph, since original ground truth graph doesn't discriminate by relation lag.

Algorithm	Precision	Recall	F1	SHD	Time (s)
micro-level	0.812	**1.000**	0.897	3	97.640
pca+pcmci	**0.909**	0.769	0.833	4	**4.984**
pca+dynotears	0.000	0.000	0.000	14	20.895
subgroups	0.800	0.923	0.857	4	34.911
group-embedding	0.867	**1.000**	**0.929**	**2**	73.369

Table 7.4: Results obtained from applying different group causal discovery algorithms to the Ultra Processed Food dataset.

level and group-embedding have a recall of 1, meaning that they have found all of the edges. Similarly, the DRCD algorithm, pca+pcmci, obtains the best precision, what isn't surprising, knowing that this is a conservative algorithm, so it manages not to fail in any predicted edge in this case.

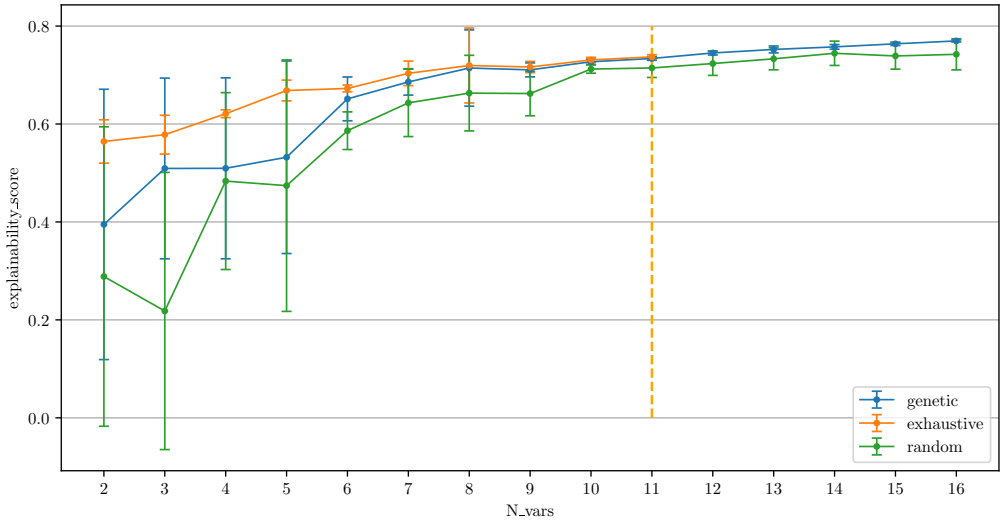
However, neither micro-level nor pca+pcmci would be recommendable for this case, since the first has a low precision, what means that it has predicted some edges that aren't actually there, and pca+pcmci has a much lower recall, what means that many potential confounders could be dismissed due to the missing edges. That way, the algorithm that reaches a best tradeoff, getting the best F1-score, is the group-embedding.

These examples demonstrate that, at least in these cases, results obtained for synthetically created datasets are extrapolable to real world problems.

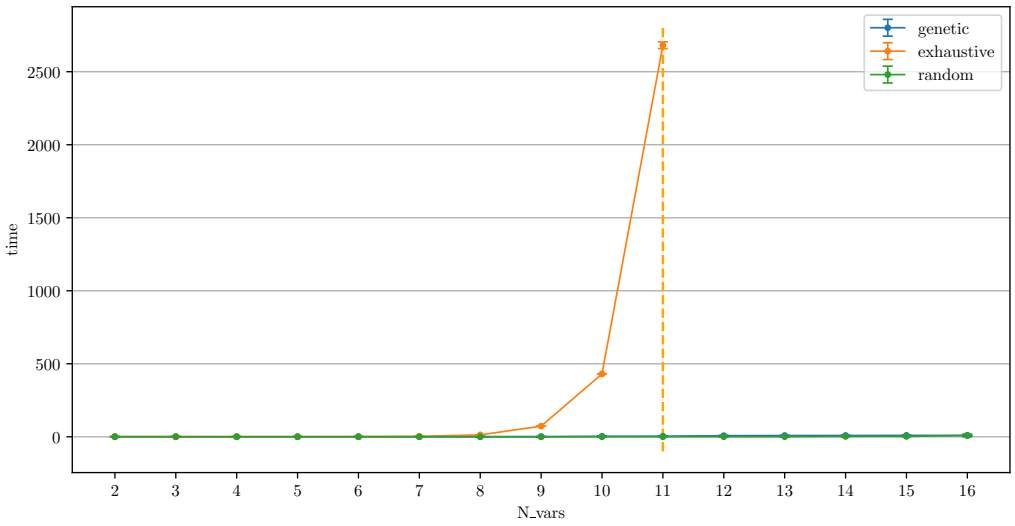
7.3. Group Extraction

As explained in Section 6.2, the number of variables is a critical parameter when estimating a correct group, because of the rapid increase in the search space. For that reason, we have performed an study on how our score varies as the number of variables in the dataset increases. Results of this study are shown in Figure 7.8. Each point is averaged over 10 experiments. Variables have linear relations and noise is $\mu_i^t \sim \mathcal{N}(0, 0.2)$. The inner- and outer-group crosslinks density have been fixed to 0.5, autocorrelation coefficients are 0.4 and dependency coefficients are randomly chosen in $\{-0.3, 0.3\}$. The orange dashed line represents a threshold from which the exhaustive group extraction becomes unfeasible.

First clear thing to note is that, as expected, the exhaustive method becomes completely unfeasible even with a relatively low number of variables like 16. If for 11 vars, being the eleventh Bell number 678 570, the time was of more than 40 minutes, one could expect that for 15 variables, where the fiftieth Bell number is 1 382 958 545, the



(a) First Component Explainability Score



(b) Time (seconds)

Figure 7.8: Average score and time, with the standard deviation thresholds, obtained from an empirical study of the three methods used for groups extraction.

spent time is around $\frac{1\,382\,958\,545}{678\,570} \cdot 40$ minutes $\sim$ 56 days, and for datasets even much smaller than the ones used in previous studies, with 20 variables, the expected time is around 5 800 years. This confirms that the use of a metaheuristic is necessary to extract groups in a general case.

Also, one can note how, starting at 8-9 variables, the genetic algorithm reaches results that are very similar to the ones obtained by the exhaustive method, so it could be expected that the values that the exhaustive method would get for more than 11 variables could be near the ones that the genetic algorithm reaches. On top of these the average results obtained by the genetic are always greater than the ones obtained by the random algorithm, and the standard deviation is always smaller, so in a general case, using the genetic approach seems to be a more intelligent approach.

8. Conclusions and Recommendations

Understanding causal mechanisms is essential for accurate modeling, prediction, and decision-making in various scientific and industrial fields. Time series data, present in disciplines like economics, neuroscience, and engineering, has unique challenges for causal discovery derived by having temporal dependencies and specific types of potential confounding factors. While progress has been made for individual time series, many real-world applications involve groups of interrelated time series, necessitating methodologies capable of handling such systems.

8.1. Group Causal Discovery for Time Series

This project addressed the lack of tested algorithms in Group Causal Discovery for Time Series by presenting a computational framework and proposing new approaches for this task. The framework and algorithms were implemented in the *group-causation* library, `https://github.com/JoaquinMateosBarroso/group-causation`.

We conducted comprehensive experiments using synthetic data to evaluate the performance of different group causal discovery methods: Micro-level Causal Discovery, Dimension Reduction + Causal Discovery (DRCD), and our proposed hybrid approaches, Subgroups Causal Discovery and Group Embedding Causal Discovery. The results from these experiments provided crucial insights. Key findings from the synthetic data experiments include:

- The Micro-level Causal Discovery algorithm demonstrated the best recall, meaning it was most successful at identifying true causal relationships, showing significant differences with most other algorithms. However, it is computationally expensive and prone to finding false positives (lower precision) as the dimensionality of the groups increases, as a single micro-level false positive can lead to a group-level false positive. This means that the algorithm could behave poorly in real-world contexts,

where

- DRCD algorithms (specifically pca+pcmci) achieved the highest precision, significantly outperforming other methods. This precision comes at the cost of lower recall, as it requires strong evidence for a causal link to be observed even after significant dimensionality reduction. DRCD methods also tend to have more false negatives in the high-dimensionality case.

- Our proposed Group Embedding Causal Discovery method achieved the best F1-score, indicating a better balance between precision and recall compared to Micro-level and DRCD methods. It maintained a high and stable performance in both metrics as the average number of variables per group increased. While it did not significantly differ from Micro-level in recall or DRCD in precision, its balanced performance resulted in a better overall F1-score and Competitive SHD. Note that the F1-score is particularly important in the causal discovery framework, as both false negatives and false positives lead to bad predictions in the posterior causal inference tasks, [Pearl, 2009], and lead to DAGs that do not represent faithfully the underlying causal process. The group embedding approach also proved more stable with respect to changes in the variance threshold parameter (κ) than the subgroups method.

- The Subgroups Causal Discovery method also showed a good balance between precision and recall, with its F1-score not significantly differing from the Group Embedding method. Both hybrid approaches demonstrated the ability to achieve recall similar to micro-level algorithms without losing as much precision.

- Experiments varying the number of groups showed that while pca+pcmci maintained high precision, other methods experienced a decline in F1-score as the number of groups increased. Our hybrid methods exhibited improved SHD compared to pcmci in this scenario.

The performance of the algorithms was further validated on real and simulated datasets, specifically the Ultra Processed Food dataset and the Tennessee Eastman dataset. The results on these datasets were consistent with the synthetic benchmarks: Micro-level methods generally yielded higher recall, DRCD methods higher precision, and the

hybrid approaches, particularly Group Embedding, provided a better balance as reflected in F1-score and SHD. The Group Embedding approach notably achieved the best Recall, F1, and SHD on the Ultra Processed Food dataset, even with a relatively low number of variables, demonstrating the potential applicability of findings from synthetic data to real-world problems.

8.2. Group Extraction

Furthermore, we addressed the need to prefix the groups in Group Causal Discovery when they are not predefined. We proposed a new metric, the First Component Explainability Score, to evaluate the quality of variable partitions. An empirical study on different optimization methods for this score showed that the exhaustive search is computationally unfeasible for even a moderate number of variables. The genetic algorithm proved to be an effective metaheuristic, reaching scores very similar to the exhaustive method for variable sets where the latter was still feasible, suggesting its utility for larger problems.

8.3. Conclusions

In conclusion, this project successfully implemented a framework for causal discovery on groups of time series and proposed novel hybrid approaches that offer a compelling balance between the strengths and weaknesses of existing methods. The Group Embedding Causal Discovery method, in particular, stands out as a robust and effective algorithm for this task, achieving the best F1-score in comparative synthetic studies. Additionally, the empirical evaluation of group extraction methods highlights the necessity and effectiveness of using optimization algorithms like the genetic algorithm for finding suitable group partitions when prior knowledge is unavailable.

8.4. Future Work

Based on the findings and limitations encountered during this project, several avenues for future work can be explored:

- Explore alternative dimensionality reduction techniques: Investigate the use of dimensionality reduction methods other than PCA within the Group Embedding framework, potentially those better suited for time series data or non-linear relationships.

- Enhance Subgroups Causal Discovery: Develop more sophisticated division functions or information functions for the subgroups method to potentially improve its performance and interpretability.

- Investigate group extraction: Further research into group extraction methods, including exploring different scores or optimization algorithms, and assessing their impact on the final group causal discovery results.

- Create and Explore more Real and Simulated Datasets: There aren't labeled datasets specific for causal discovery in the groups of time series context. It would be needed to apply the developed framework and algorithms to a wider range of real-world datasets with varying characteristics (e.g., non-linear dependencies, different noise distributions, presence of latent confounders) to thoroughly evaluate their robustness.

- Extend algorithms to the multiple time series case: All studied algorithms extract DAGs parting from a single time series, with several timestamps. Given that the algorithms have similar mathematical foundations, it would be interesting to explore the extension of these foundations to the case in which there are various time series that are known to have been extracted from the same causal process.

- Extend theoretical foundations: Given that causal discovery in the context of groups of time series is in a very primeval state, theoretical guaranties for specific algorithms and methods is still in process of being settled. In particular, the mathematical extension of interventions and counterfactuals to this context would be a very significant contribution.

Bibliography

[Ahmad et al., 2024] Ahmad, W., Shadaydeh, M., and Denzler, J. (2024). Deep learning-based group causal inference in multivariate time-series. *arXiv preprint arXiv:2401.08386*.

[Ashimine, 2023] Ashimine, I. E. (2023). GitHub - salesforce/causalai: Salesforce CausalAI Library: A Fast and Scalable framework for Causal Analysis of Time Series and Tabular Data — github.com. `https://github.com/salesforce/causalai`. [Accessed 16-03-2025].

[Assaad et al., 2022] Assaad, C. K., Devijver, E., and Gaussier, E. (2022). Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73:767–819.

[Bandurchin, 2024] Bandurchin, A. (2024). Why fastapi is the best choice for your next python web project. `https://towardsdatascience.com/why-fastapi-is-the-best-choice-for-your-next-python-web-project-f90c3f1c5c32`. Accessed: 2025-05-05.

[Bell, 1934] Bell, E. T. (1934). Exponential polynomials. *Annals of Mathematics*, 35(2):258–277.

[Beretta et al., 2017] Beretta, S., Castelli, M., Gonçalves, I., and Ramazzotti, D. (2017). A quantitative assessment of the effect of different algorithmic schemes to the task of learning the structure of bayesian networks. *CoRR*, abs/1704.08676.

[Bernasconi et al., 2024] Bernasconi, A., Zanga, A., Lucas, P. J., Scutari, M., Di Cosimo, S., De Santis, M. C., La Rocca, E., Baili, P., Cavallo, I., Verderio, P., et al. (2024). From real-world data to causally interpretable models: A bayesian network to predict

cardiovascular diseases in adolescents and young adults with breast cancer. *Cancers*, 16(21):3643.

[Bogachev and Ruas, 2007] Bogachev, V. I. and Ruas, M. A. S. (2007). *Measure theory*, volume 1. Springer.

[Bojinov et al., 2019] Bojinov, I., Tu, Y., Liu, M., and Xu, Y. (2019). Causal inference from observational data: Estimating the effect of contributions on visitation frequency atlinkedin. *BMC Medical Research Methodology*.

[Bondy and Murty, 2008] Bondy, J. A. and Murty, U. S. R. (2008). *Graph theory*. Springer Publishing Company, Incorporated.

[Box et al., 2015] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

[Bystrova et al., 2024] Bystrova, D., Assaad, C. K., Arbel, J., Devijver, E., Gaussier, E., and Thuiller, W. (2024). Causal discovery from time series with hybrids of constraint-based and noise-based algorithms. *Transactions on Machine Learning Research*.

[de Campos et al., 2002] de Campos, L., Fernández-Luna, J., Gámez, J., and Puerta, J. (2002). Ant colony optimization for learning bayesian networks. *International Journal of Approximate Reasoning*, 31:291–311.

[Entner and Hoyer, 2010] Entner, D. and Hoyer, P. (2010). On causal discovery from time series data using fci. *Proceedings of the 5th European Workshop on Probabilistic Graphical Models, PGM 2010*.

[Faller and Janzing, 2025] Faller, P. M. and Janzing, D. (2025). On different notions of redundancy in conditional-independence-based discovery of graphical models.

[for Geeks, 2025] for Geeks, G. (2025). Introduction to jinja2 templating engine. `https://www.geeksforgeeks.org/jinja2-templating-engine/`. Accessed: 2025-05-05.

[Galvani et al., 2021] Galvani, M., Bardelli, C., Figini, S., and Muliere, P. (2021). A bayesian nonparametric learning approach to ensemble models using the proper bayesian bootstrap. *Algorithms*, 14(1):11.

[GitHub, 2021] GitHub (2021). Github copilot. `https://copilot.github.com/`. Accessed: 2025-05-05.

[Granger, 1969] Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438.

[Günther et al., 2023] Günther, W., Ninad, U., and Runge, J. (2023). Causal discovery for time series from multiple datasets with latent contexts.

[Hasan et al., 2024] Hasan, U., Hossain, E., and Gani, M. O. (2024). A survey on causal discovery methods for i.i.d. and time series data.

[Hunter, 2007] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

[Hyvärinen et al., 2010] Hyvärinen, A., Zhang, K., Shimizu, S., and Hoyer, P. O. (2010). Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(56):1709–1731.

[Ibarrola and Rumeau, 1977] Ibarrola, R. V. and Rumeau, T. P. (1977). *Procesos estocásticos*. Universidad Nacional de Educación a Distancia.

[Iqbal et al., 2015] Iqbal, K., Yin, X.-C., Hao, H.-W., Ilyas, Q. M., and Ali, H. (2015). An overview of bayesian network applications in uncertain domains. *International Journal of Computer Theory and Engineering*, 7(6):416.

[Kahn, 1962] Kahn, A. B. (1962). Topological sorting of large networks. *Communications of the ACM*, 5(11):558–562.

[Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

[Lewis and Groth, 2020] Lewis, A. D. and Groth, K. M. (2020). A dynamic bayesian network structure for joint diagnostics and prognostics of complex engineering systems. *Algorithms*, 13(5):64.

[Malinsky and Spirtes, 2019] Malinsky, D. and Spirtes, P. (2019). Learning the structure of a nonstationary vector autoregression. In Chaudhuri, K. and Sugiyama, M., editors,

Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, volume 89 of *Proceedings of Machine Learning Research*, pages 2986–2994. PMLR.

[Marsden et al., 1991] Marsden, J. E., Tromba, A. J., and Mateos, M. L. (1991). *Cálculo vectorial*, volume 69. Addison-Wesley Iberoamericana México.

[Melançon et al., 2001] Melançon, G., Dutour, I., and Bousquet-Mélou, M. (2001). Random generation of directed acyclic graphs. *Electronic Notes in Discrete Mathematics*, 10:202–207. Comb01, Euroconference on Combinatorics, Graph Theory and Applications.

[Menegozzo et al., 2022] Menegozzo, G., DallAlba, D., and Fiorini, P. (2022). Cipcad-bench: Continuous industrial process datasets for benchmarking causal discovery methods. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 2124–2131.

[Moraffah et al., 2021] Moraffah, R., Sheth, P., Karami, M., Bhattacharya, A., Wang, Q., Tahir, A., Raglin, A., and Liu, H. (2021). Causal inference for time series analysis: Problems, methods and evaluation. *Knowledge and Information Systems*, 63:3041–3085.

[Nouri, 2023] Nouri, A. (2023). diva-portal.org. `https://www.diva-portal.org/smash/get/diva2:1749596/FULLTEXT01.pdf`. [Accessed 16-05-2024].

[Paloma and Pérez, 1988] Paloma, V. Q. and Pérez, A. G. (1988). *Lecciones de cálculo de probabilidades*. Ediciones Díaz de Santos.

[Pamfil et al., 2020] Pamfil, R., Sriwattanaworachai, N., Desai, S., Pilgerstorfer, P., Beaumont, P., Georgatzis, K., and Aragam, B. (2020). Dynotears: Structure learning from time-series data.

[Partovi et al., 2022] Partovi, M., Amra, M., Pahlevanzadeh, M., Alwardi, A., and Fathi, M. R. (2022). Predictable maintenance: A bayesian network-based model. *International Journal of Reliability, Risk and Safety: Theory and Application*, 5(2):97–105.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

[Pearl, 2009] Pearl, J. (2009). *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

[Pearl and Mackenzie, 2018] Pearl, J. and Mackenzie, D. (2018). *The Book of Why*. Basic Books.

[PyGraphViz, 2014] PyGraphViz (2014). Legal; PyGraphviz 1.3rc1 documentation — pygraphviz.github.io. `https://pygraphviz.github.io/documentation/pygraphviz-1.3rc1/reference/legal.html`. [Accessed 14-04-2025].

[Ramírez, 2023] Ramírez, S. (2023). Fastapi documentation. `https://fastapi.tiangolo.com/`. Accessed: 2025-05-05.

[Rao, 1973] Rao, C. R. (1973). *Linear statistical inference and its applications*, volume 2. Wiley New York.

[Reiser, 2022] Reiser, C. (2022). Causal discovery for time series with latent confounders.

[Runge, 2020] Runge, J. (2020). Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In Peters, J. and Sontag, D., editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 1388–1397. PMLR.

[Runge et al., 2023] Runge, J., Gerhardus, A., Varando, G., Eyring, V., and Camps-Valls, G. (2023). Causal inference for time series. *Nature Reviews Earth & Environment*, 4.

[Runge et al., 2025] Runge, J., Gillies, E., Strobl, E., and Rabel, M. (2025). GitHub - jakobrunge/tigramite: Tigramite is a python package for causal inference with a focus on time series data. The Tigramite documentation is at — github.com. `https://github.com/jakobrunge/tigramite`. [Accessed 16-03-2025].

[Runge et al., 2019] Runge, J., Nowack, P., Kretschmer, M., Flaxman, S., and Sejdinovic, D. (2019). Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, 5(11):eaau4996.

[Shimizu et al., 2006] Shimizu, S., Hoyer, P. O., ;rinen, A., and Kerminen, A. (2006). A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(72):2003–2030.

[Shimizu et al., 2012] Shimizu, S., Hyvarinen, A., Kano, Y., and Hoyer, P. O. (2012). Discovery of non-gaussian linear causal models using ica.

[Spirtes and Glymour, 1991] Spirtes, P. and Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72.

[Stack, 2025] Stack, B. (2025). Top python templating engines in 2025. `https://betterstack.com/community/comparisons/python-templating-engines/`. Accessed: 2025-05-05.

[Stigler, 1990] Stigler, S. M. (1990). *The history of statistics*. Harvard University Press.

[Tsamardinos et al., 2006] Tsamardinos, I., Brown, L., and Aliferis, C. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65:31–78.

[Vélez Ibarrola, 2019] Vélez Ibarrola, R. (2019). *Cálculo de probabilidades 2*. Universidad Nacional de Educación a Distancia.

[Vélez Ibarrola and Hernández Morales, 1995] Vélez Ibarrola, R. and Hernández Morales, V. (1995). *Cálculo de probabilidades 1*. Universidad Nacional de Educación a Distancia.

[Vélez Ibarrola and Pérez, 2013] Vélez Ibarrola, R. and Pérez, A. G. (2013). *Principios de inferencia estadística*. UNED, Universidad Nacional de Educación a Distancia.

[Wahl et al., 2023] Wahl, J., Ninad, U., and Runge, J. (2023). Vector causal inference between two groups of variables. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

[Wahl et al., 2024] Wahl, J., Ninad, U., and Runge, J. (2024). Foundations of causal discovery on groups of variables.

[Waskom, 2021] Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

[Zheng et al., 2018] Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. (2018). Dags with no tears: Continuous optimization for structure learning.

[Zielke, 1987] Zielke, G. (1987). Horn, ra; johnson, cr, matrix analysis. cambridge etc., cambridge university press 1985. xiii, 561 s.,£ 35.00. isbn 0-521-30586-1. *Zeitschrift Angewandte Mathematik und Mechanik*, 67(3):212–212.

A. Mathematical Appendix

A.1. Probability Theory

Probability theory is an area of mathematics of great historical relevance [Stigler, 1990]. It is the common framework where many fields that try to model relationships between non-deterministic events in the real world work; among them, statistical inference and causal inference. However, modern science and engineering degrees tend to leave aside issues of definition and properties of random variables and other theoretical aspects to study statistical methods.

Therefore, in order to understand the starting point and what are the advantages of approaching problems from a causal point of view, it is interesting to briefly review this important theory.

What is a probability?

The definition of probability is one of those fascinating questions that are wrapped around mathematics, and it is definitely a non-trivial matter. Many books have wonderful chapters on this definition [Vélez Ibarrola and Hernández Morales, 1995, Chapters 1 and 2] [Vélez Ibarrola, 2019, Chapters 1, 2 and 3], and in this section there have been left some small pills on this subject.

Probability is a question of **uncertainty** . When we say that a coin has a 50% probability of landing on its face, we do not mean that the future state of the coin is completely impossible to predict. A good physical modelization of the problem would be able to predict exactly how the coin will fall and stay in the end.

However, an affirmation about probability is an affirmation about **uncertainty**. It states that, according to the information we have, if we threw the coin, with unknown

conditions, "an infinite amount of times", the coin is expected to land on its face half of the times. The family of Central Limit Theorems [Vélez Ibarrola and Hernández Morales, 1995] is able to make this affirmation more robust, but for the moment, we will focus on the problem of assigning a certain number, between 0 and 1, to specific events that are contained in a sample space.

Definition A.1 (Probability)**.** *Given a measurable space* $(\Omega, \mathcal{F})$ *[Bogachev and Ruas, 2007], a* ***probability*** *(or probability measure) is an application*

$$P : \mathcal{F} \to \mathbb{R}$$

such that

1. $P(A) \geq 0, \ \forall A \in \mathcal{F}$
2. *For all numerable collection of events,* $\{A_n\} \subseteq \mathcal{F}$*, if* $A_i \cap A_j = \emptyset \ \forall i \neq j$*, then*

$$P(\cup_i A_i) = \sum_i P(A_i)$$

3. $P(\Omega) = 1$

The first event will be immediate to assume is the second one, but it is easy to check that if we have a set of events (e.g. the set of dices falling on faces $i \in \{i_1, ..., i_m\}$ in a dice of faces n) that are disjoint (the dice cannot fall at the same time in faces 2 and 3), then the probability of the union of these events (probability of obtaining any of the faces in the set) is the sum of the probabilities of each individual event $\big(P(\text{Face}_{i_1} \cup \cdots \cup \text{Face}_{i_m}) = \sum_{j=1}^{m} P(\text{Face}_{i_j})\big)$.

There are many very interesting properties of probability that can be checked in [Paloma and Pérez, 1988, Vélez Ibarrola and Hernández Morales, 1995, Vélez Ibarrola and Pérez, 2013, Vélez Ibarrola, 2019], but, in order to be concise, we will focus on those that are relevant for the task at hand.

Bayes Theorem

Bayes theorem is a classical predicate that relates conditional probabilities with more simple events. The motivation for the inclusion of it in this study is mainly based on the

fact that Bayes Theorem is one of the first attempts to study causality directly through probability.

Before facing this theorem, it is important to understand what a conditional probability is (very different from the interventions we will see in the context of causal inference):

Definition A.2 (Conditional Probability)**.** *Given 2 events in a probability space* $(\Omega, \mathcal{F}, P)$*,* $A, B \in \mathcal{F}$*, the probability of* A *conditioned by* B *is*

$$P(A|B) := \frac{P(A \cap B)}{P(B)}$$

Knowing this basic definition, we can get to the desired

Theorem A.1 (Bayes' Theorem)**.** *Given an event* A *in a probability space, and a numerable collection of disjoint by pairs events* $\{B_i\} \subseteq \mathcal{F}$*, with* $P(B_i) > 0$*, and* $\cup_i \{B_i\} = \Omega$*, then*

$$P(B_j|A) = \frac{P(B_j)P(A|B_j)}{P(A)} = \frac{P(B_j)P(A|B_j)}{\sum_i P(B_i)P(A|B_i)}$$

Proof. The first equality is trivial, checking that $P(B_j|A) = \frac{P(B_j \cap A)}{P(A)} = \frac{P(B_j)P(A|B_j)}{P(A)}$.
For the second one, the only step done is the application of the total probability formula:

$$P(A) = P(A \cap \Omega) = P\big(A \cap (\cup_i \{B_i\})\big) = P\big(\cup_i (A \cap B_i)\big)$$

So, applying the second property of the probability definition:

$$P(A) = \sum_i P(A \cap B_i) = \sum_i P(B_i)P(A|B_i)$$

□

In simple terms, if we know the implications that observing the events $\{B_i\}$ has in A, then we can extract the implications that observing the event A has on each particular B_i.

When trying to extract conclusions about similar, but different questions, such as "What would be the probability of obtaining certain medical result, B_j, if on a patient

over which certain medicament A was applied, we had applied the medicament A'?" (**counterfactual**; [Pearl, 2009]), we might think that obtaining the probability $P(B_j|A')$ is a good idea. Though, as Judea Pearl explains magnificently in his book [Pearl and Mackenzie, 2018], this is not the best idea.

When calculating $P(B_j|A')$, we are considering many situations that have not necessarily happened. When using, for example, the Bayes' Theorem to obtain it, we are using all the probabilities $P(A'|B_i)$, which are obtained in cases in which a doctor decided that the best medication for certain patient was A'.

This affirmation is much clearer with an example: If B_1 is the event of of the patient's survival, and B_2 is its complementary ($B_2 = B_1^c$), then, if we have applied certain simple medicament, A, to a patient that had a slight cough, and he has survived, we might think about calculating the probability of survival when applying another, also simple, medicament, A', that is only applied on patients that have a terminal illness.
We would obtain that $P(B_1|A') = 0$, and conclude that the patient, who arrived to the hospital with a slight cough, is going to die with a probability of 100% if we give to him a simple medicament.

This case seems ridiculous, but when using inferencist and machine learning algorithms, we are constantly making similar assumptions, and these examples were one of the main motivations for the growth in recent years of **Causal Inference**.

Random Variables

The last theme to treat before entering in Causal Inference is, probably, one of the biggest tools that made statistics and data science what it is nowadays: *random variables*.
In simple terms, they are the tool through which we are able to study the behavior, in probabilistic terms, of real (or any subset of $\mathbb{R}^k$) encoding problems, and combinations of various of these problems.

Definition A.3 (Random Variable)**.** *A* 1*-dimensional,* ***random variable*** *in a probability space* $(\Omega, \mathcal{F}, P)$ *is any function*

$$X : \Omega \to \mathbb{R}$$

that is measurable [1].

Every absolutely continuous random variable is associated with a ***probability distribution****:*

$$P_X(B) := P\{\omega \in \Omega | X(\omega) \in B\}$$

and with the corresponding ***distribution function****:*

$$F_X(x) := P\{\omega \in \Omega | X(\omega) < x\}$$

and its associated density/probability function $f_X : \Omega \to \mathbb{R}$*.*

If $\Omega = \mathbb{R}$, $\mathcal{F} = \mathbb{B}$ *and the variable is absolutely continuous[Vélez Ibarrola, 2019], then we have:*

$$F_X(x) = \int_{-\infty}^{x} f(t)dt$$

This definition is usually joined to the classical notation

$$\{X \in B\} := \{\omega \in \Omega | X(\omega) \in B\}$$

which allows obtaining probabilities in a more understandable way.

With this notation, given a random variable X with a distribution $\mathcal{N}(\mu, \sigma^2)$ $\big($notated $X \sim \mathcal{N}(\mu, \sigma^2)\big)$, the probability of X being lower than certain value $x \in \mathbb{R}$ is:

$$F_X(x) = P\{X < x\} = \int_{-\infty}^{x} f_X(t)dt = \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt$$

It is easy to proof that

$$P(\Omega) = F_X(\infty) = \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1$$

using the multivariable, integration variable change [Marsden et al., 1991], by applying the Jacobian of $(x, y) = (r\cos\theta, r\sin\theta)$ (details are left as an exercise for the reader).

This means that, using the properties in Definition 1.1 and 1.3, the normal distribution defines a correct random variable over a sample space Ω.[2]

[1] I.e., such that $\vec{X}^{-1}(B) \in \mathcal{F}$, for all $B \in \mathbb{B}$ (Borel σ-algebra [Bogachev and Ruas, 2007]).

[2] This sample space cannot be any set. In particular, it must contain, over a surjective application, to $\mathbb{R}$.

Expectation and Covariance

When we speak about the expectation of a Random Variable, we are speaking about what is the "outcome that will happen on average". This does not mean that it is the most probable outcome (mode); in fact, it doesn't even have to be a possible outcome.

Definition A.4 (Expectation and Variance)**.** *Given an absolutely continuous random variable, X, with density $f_X(x)$, its* ***expectation*** *is*

$$\mathbb{E}[X] := \int_{-\infty}^{\infty} x f_X(x) dx$$

On the other hand, it X is a discrete random variable, with distribution $P_X\{X = i\}$, its ***Expectation*** *is*

$$\mathbb{E}[X] := \sum_{x \in Image(X)} x \cdot P_X\{X = x\}$$

The ***variance*** *of any random variable, X, is the expectation of the square of the difference between X and $\mathbb{E}[X]$:*

$$VAR(X) := \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2 + \mathbb{E}[X]^2 - 2X\mathbb{E}[X]] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

The ***covariance*** *between 2 random variables X, Y is the expectation of the square of the difference between XY and $\mathbb{E}[XY]$:*

$$COVAR(X, Y) := \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$$

This means that the covariance between X and Y is the expectation of the product of the differences of these random variables with their expectations.
The covariance definition could seem a little arbitrary if we didn't check the proof of the following

Lemma A.1. *2 random variables X, Y, have null covariance iff,*

$$VAR(X + Y) = VAR(X) + VAR(Y)$$

Proof. ($\Rightarrow$) If we denote $\mu_X = \mathbb{E}[X], \mu_Y = \mathbb{E}[Y]$, then it is $\mathbb{E}[X + Y] = \mu_X + \mu_Y$ and we

have:

$$\text{VAR}(X+Y) = \mathbb{E}[(X+Y-\mu_X-\mu_Y)^2] =$$

$$= \mathbb{E}[(X-\mu_X)^2] + \mathbb{E}[(Y-\mu_Y)^2] + 2\mathbb{E}[(X-\mu_X)(Y-\mu_Y)] =$$

$$= \mathbb{E}[(X-\mu_X)^2] + \mathbb{E}[(Y-\mu_Y)^2]$$

($\Leftarrow$) It is important to note that we have been able to perform the last equality just due to the supposition that $2\mathbb{E}[(X-\mu_X)(Y-\mu_Y)] = 2\text{COVAR}(X,Y) = 0$.

□

This means that the variance of the random variable[3] $X+Y$ is explained by the addition of the variances of X and Y iff their covariance is null.

Likelihood Ratio

Lastly, the likelihood ratio is a very powerful tool that will be needed will be needed to construct some conditional independence tests between random variables.
This important tool is very intuitively interpretable; we are going to contrast how little is the probability of the null hypotheses, compared to the probability of the whole space. The main theorem about Likelihood Ratios is[4]:

Theorem A.2. *Let X be a random variable over a sample space Ω, with density $f_\theta(x)$, whose parameter θ is in a space $\Theta \subseteq \mathbb{R}^k$, over which we want to contrast the null hypothesis $H_0 : \theta \in \Theta_0 = \{\theta \in \Theta | \theta_i = g_i(\omega_1, ..., \omega_q), with (\omega_1, ..., \omega_q) \in \Omega\}$, being Θ_0 an arbitrary subset of Θ. If we denote as likelihood ratio of a simple random sample, $(X_1, ..., X_n)$,*

$$\Lambda := \frac{\max_{\theta\in\Theta_0} f_\theta(x_1, ..., x_n)}{\max_{\theta\in\Theta} f_\theta(x_1, ..., x_n)}$$

Then, if the actual parameter is $\theta_0 \in \Theta_0$, we have the following distribution limit:

$$-2\log\Lambda(X_1, ..., X_n) \xrightarrow{d_{\theta_0}} \chi^2_{k-q}$$

[3] Or, equivalently, the variance given by any random variable $aX + bY + c$, with $a, b, c \in \mathbb{R}$, $a, b \neq 0$.
[4] An elegant proof of this theorem can be consulted at [Rao, 1973], page 418.

This means that if we are able to obtain the Maximum Likelihood estimator[5] of θ, then we will be able to find the statistical distribution that a function of our sample space, supposing $\theta \in \Theta_0$, follows. This is easily usable for the construction of statistical tests[6].

[5]The estimator of θ, T, that is defined by the rule $f_T(x_1, ..., x_n) = \max_{t \in \Theta} f_t(X_1, ..., X_n)$ is, by itself, an important measure of our space, and is named after Maximum Likelihood estimator [Vélez Ibarrola and Pérez, 2013].

[6]In particular, this kind of tests are usually known as Pearson's χ^2 tests.

A.2. Graph Theory

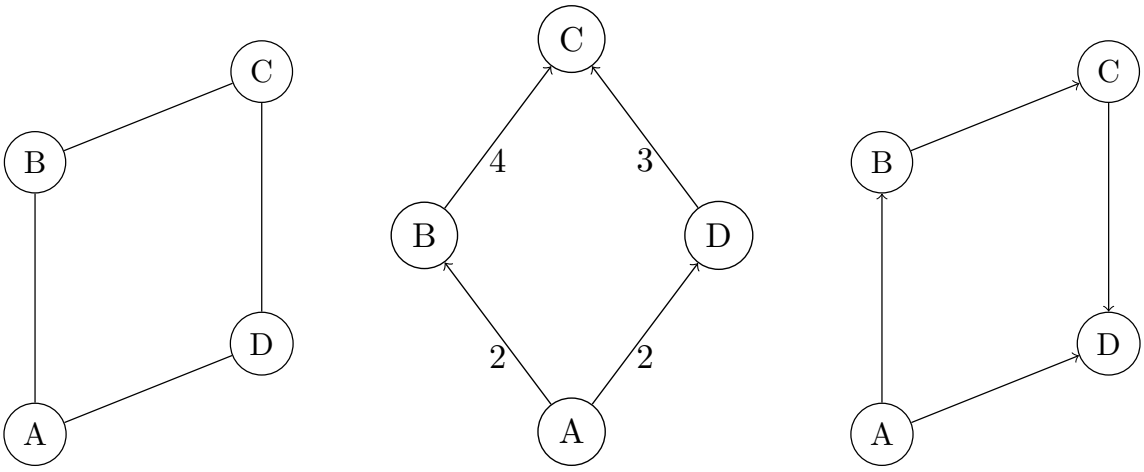
When modeling real world events and processes one often encounters the need to create points connected by paths, to join ideas through logical connections or to follow a series of steps. For all these purposes and many more theoretical needs, graphs provide a strong background [Bondy and Murty, 2008]:

Definition A.5 (Graph)**.** *A* ***graph****, or undirected graph, is a pair* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, where* $\mathcal{V}$ *is the set of* ***vertices*** *or* ***nodes*** *that will compound the basic elements of our model, and* $\mathcal{E}$ *is the set of* ***edges****; a set of unordered pairs of nodes* $\{A, B\} = \{B, A\}$*, being* $A, B \in \mathcal{V}$*, that connects them. The edge* $\{A, B\}$ *is also notated* $A - B$*.*
If, on the other hand, edges in $\mathcal{E}$ *are ordered;* $(A, B) \neq (B, A)$*, then* $\mathcal{G}$ *is said to be a* ***directed graph****. The edge* (A, B) *is also notated* $A \to B$ *or* $B \leftarrow A$*.*
If $\mathcal{E}$ *is a set that can be partitioned in* $\mathcal{E} = \mathcal{E}_U \cap \mathcal{E}_D$*, being* $(\mathcal{G}, \mathcal{E}_U)$ *and* $(\mathcal{G}, \mathcal{E}_D)$ *a directed and undirected graph, respectively, and we interpret an undirected edges as a bidirected edge,* $A \leftrightarrow B$*, then we say that* $(\mathcal{G}, \mathcal{E})$ *is a* ***directed mixed graph****.*
If each edge $e \in \mathcal{E}$ *has an associated value, generally in* $\mathbb{R}$*, then* $\mathcal{G}$ *is said to be a* ***weighted graph****.*

Each of these types of graph may be useful for different purposes. A classic example of the use of undirected graphs is the modelization of the roads connections between cities of a country, where the outward journey is equivalent to the return one. A visual representation of a graph of this kind is shown in Figure A.1a.
A directed graph, on the other hand, may be useful when the paths cannot always be reversed. An example of this kind of graph, when there also are weights associated with each edge, is plotted in Figure A.1b.

In any of these cases, A and C are not directly connected, i.e., there's no edge $A \to C$ in the set of edges $\mathcal{E}$. However, they seem to be indirectly connected through either B or D, in two different ways, and one could argue that the distance in the directed graph from A to C should be 5, as going through D is the "fastest" way to get to C. This idea is formalized in the following definition:

Definition A.6 (Graph walk and path)**.** *Given a graph* $(\mathcal{V}, \mathcal{E})$*, a* ***walk*** π *between 2 nodes* $A, B \in \mathcal{V}$ *is an ordered tuple* $(e_1, e_2, ..., e_{n-1})$*, with* $e_i \in \mathcal{E}$*, s.t.* $\exists \pi_1, \pi_2, ..., \pi_n \in \mathcal{V}$*, being*



(a) Undirected unweighted graph **(b)** Directed weighted acyclic graph **(c)** Directed acyclic graph

Figure A.1: Representations of basic graphs.

$\pi_1 = A, \pi_n = B$, *that are directly connected through* e_i, *i.e., such that* $e_i = \{\pi_i, \pi_{i+1}\}$ $\forall i \in \{1, ..., n-1\}$.

If a walk does not pass through the same node twice, $\pi_i \neq \pi_j$, $\forall i \neq j$, *then the walk is said to be a* ***path****.*

A ***directed*** *walk (path) is a walk (path) in a directed graph, where previous properties are true for edges* $e_i = (\pi_i, \pi_{i+1}) \equiv \pi_i \rightarrow \pi_{i+1}$

An important definition that will be necessary to construct consistent causal models is related with paths:

Definition A.7 (Acyclic graph)**.** *If a path* π *starts and ends in the same point,* $\pi_1 = \pi_n$, *then the path is said to be a* ***cycle****.*

An ***acyclic graph*** *is a graph with no cycles.*

For example, in Figure A.1c, the walk $A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$ is a cycle, but in Figure A.1b there is no cycle, so it is an acyclic graph. It will be important for our purposes to detect whether a given graph is acyclic or not. A simple way to check acyclity in directed graphs is looking for a topological sorting of the graph, what means an ordering of all the nodes $V_1, ..., V_n \in \mathcal{V}$ such that if $V_i \rightarrow V_j \in \mathcal{E}$ then $i < j$. To do so, [Kahn, 1962] designed an algorithm that finds a topological sorting of a graph always that is possible, with a time complexity of $\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|)$. It is shown in Algorithm 7.

Algorithm 7 Kahn's algorithm for topological sorting of a graph

1: **Input: Directed graph** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
2: **Output:** Topological sorting of the graph; $V_1, ..., V_n$
3: **procedure** KAHN'S ALGORITHM
4: $\mathcal{L} \leftarrow$ Empty list that will contain sorted nodes
5: $\mathcal{S} \leftarrow$ Set of all nodes without incoming edge
6: **while** $\mathcal{S} \neq \emptyset$ **do**
7: remove any $V \in S$
8: add V to $\mathcal{L}$
9: **for** $U \in \mathcal{V}$ such that $V \rightarrow U$ **do**
10: remove $V \rightarrow U$ from $\mathcal{E}$
11: **if** U has no other incoming edges **then**
12: insert U into $\mathcal{S}$
13: **return** $\mathcal{L}$

A.3. Time Series and Stochastic Processes

Time series

Many sets of data are presented in an ordered way, generally indexed by time: the value of a financial asset across the day, the condition of a vehicle during its time of use, or the number of hospital admissions due to covid each day of quarantine. For that reason, it is interesting to have a mathematical object that represents the collected information in this way [Box et al., 2015]:

Definition A.8 (Time series)**.** *A* ***time series*** *is a sequence of observations (usually values in* $\mathbb{R}^n$*) taken sequentially in time (i.e., values* x_t *have an index* $t \in T$*, being* T *the set of parameters (usually, possible moments for the observation).*

When the time series is discrete (i.e., T is numerable), it is common to find the representations $(x_1, x_2, ...)$; $\{x_t\}_{t=1}^{\infty}$, but, for any kind of time series, the common specification is in the kind $\{x_t : t \in T\}$. There are 2 main types of mathematical models for the analysis and acquisition of these time series:

- *Deterministic models*, which give one simple value $x_t \in \mathbb{R}^n$ for each time $t \in T$.
- *Stochastic models*, which, by means of random variables, give a set of possible values for the time series at a moment t, and certain probabilities for these values. As seen previously, the use of random variables gives great statistical and computational power.

Deterministic models

They are generally extracted from an abstract modelization of the problem[7]. Though, sometimes, when we have a non-deterministic process, X_t, we use some estimator dependent of X_t to specify a deterministic time series, e.g., $x_t := E[X_t]$, or defining x_t as the maximum-likelihood estimator of X_t.

There are some special cases:

[7]Because real world measurements and estimations are, obviously, not able to obtain 100% precise measures.

- When $T = \mathbb{N}$, then the sequence is $\{x_t\}_{t\in\mathbb{N}} \subseteq W$, what, over the proper topology, can be seen as an abstract, mathematical, sequence, and has many special convergence properties.

- When $T \subseteq \mathbb{R}$, and the application $s : T \to W, \quad s(t) := x_t$ is continuously differentiable (C^1), then the series is the parametrization of $s(T)$, and it can be interpreted as a mathematical *trajectory*, which may represent certain property of a particle during its movement, or the mass of an object along its surface/hypervolume.

Stochastic Processes

As mentioned before, real world measurements are usually extracted, not without any additional information, but following certain inner processes and having probabilities for the obtained values, which are related with other properties of a system or with the time at which it was extracted. [Ibarrola and Rumeau, 1977]

Definition A.9 (Stochastic Process)**.** *Given a probability space (Ω, $\mathcal{F}$, P), a* ***stochastic process*** *with set of parameters T and state space E is a family of random variables $\{X_t\}_{t\in T}$, such that, for each $t \in T$,*

$$X_t : \Omega \to E.$$

Given an event $\omega \in \Omega$, there is a time series, named ***trajectory*** *of the process, associated with ω,*

$$X(\omega) : T \ni t \mapsto X_t(w) \in E.$$

That way, there could be defined a stochastic process with the possible values for the temperature obtained from a thermometer during a 24-hour day (measured in hours, $T = (0, 24]$), and the records obtained one day would be the trajectory of the process for that particular day.

A.4. Conditional Independence Tests

Definition A.10 (Conditional independence)**.** *Let X, Y, Z be 3 random variables (or processes). X and Y are said to be* ***conditionally independent*** *given Z, $X \perp\!\!\!\perp Y$, if and only if the joint conditional distribution factors as*

$$f_{X,Y|Z}(x, y|z) = f_{X|Z}(x|z) f_{Y|Z}(y|z)$$

for almost every[8] *value of z.*

This definition extends naturally to time series $\{X_t\}, \{Y_t\}, \{Z_t\}$ and to groups of time series.

Discrete Conditional Independency Tests

Even though conditional independence is very easy to define, it is not to check numerically. On the other hand, when variables are discrete and with a low number of possible values it is relatively easy to find an optimal hypothesis test that contrasts conditional independency.

χ^2 Independence test

The study of different independence test is a very extensive and formal theme, so here there will be explained a simple one; the χ^2 independence test[9].

This test requires the discretization of our sample space, which usually is $\Omega = \mathbb{R}^2$, in k sets, $\{A_1, ..., A_k\}$ for the first variable, X, and r sets, $B_1, ..., B_r$ for the second variable, Y.

Once a simple sampling is performed over our space, its information can be compiled in a *contingency table*, which stores, in the position (i, j), the number of instances from the sampling, that, being (x, y), comply with $x \in A_i$ and $y \in B_j$.

That way, if we denote $p_{ij} := P\{(x, y) \in \Omega | x \in A_i, y \in B_j\}$, the probability of obtaining the contingency table in Table A.1 can be easily obtained via a multinomial

[8] Note that in probability, as in measure theory, an event A is said to happen *almost everywhere* or *almost surely* iff the complementary set has measure zero, or, equivalently, if $P(A) = 1$.

[9] It is a non-parametric test, so, in general, we will not have problems with the needed hypotheses. These hypotheses can be consulted at [Vélez Ibarrola and Pérez, 2013], page 395.

	B_1	B_2	$\cdots$	B_r	Total
A_1	n_{11}	n_{12}	$\cdots$	n_{1r}	$n_{1\cdot}$
A_2	n_{21}	n_2	$\cdots$	n_{2r}	$n_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
A_k	n_{k1}	n_{k2}	$\cdots$	n_{kr}	$n_{k\cdot}$
Total	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot r}$	n

Table A.1: Representation of a general contingency table.

distribution, and it is the following:

$$n! \prod_{i=1}^{k} \prod_{j=1}^{r} \frac{1}{n_{ij}!} p_{ij}^{n_{ij}}$$

We will use the likelihood ratio test. For its calculus we will need the maximum of this function, so let's calculate it.

Our only restriction is $\sum_{i,j} p_{ij} = 1$, so it is easy to calculate this maximum via the Lagrange multipliers. In order to apply this method, we construct

$$h(\vec{p}) = n! \prod_{i,j} \frac{1}{n_{ij}!} p_{ij}^{n_{ij}} + \lambda(\sum_{i,j} p_{ij} - 1)$$

and find the points where the gradient of h is null:

$$\vec{\nabla} h(\vec{p}) = 0 \Leftrightarrow n! \Big(\prod_{(i,j) \neq (r,s)} \frac{p_{ij}^{n_{ij}}}{n_{ij}!} \Big) \frac{p_{rs}^{n_{rs}-1}}{n_{rs} - 1} - 1 = 0, \quad \forall r, s$$

That way, $\forall r, r', s, s'$ s.t. $r \neq r'$ and $s \neq s'$, it is true that

$$\frac{p_{r's'}^{n_{r's'}}}{n_{r's'}!} \cdot \frac{p_{rs}^{n_{rs}-1}}{(n_{rs}-1)!} = \frac{p_{rs}^{n_{rs}}}{n_{rs}!} \cdot \frac{p_{r's'}^{n_{r's'}-1}}{(n_{r's'}-1)!}$$

from where we can extract that

$$p_{r's'} = \frac{n_{r's'}}{n_{rs}} p_{rs}$$

what implies

$$1 = \sum_{r's'} p_{r's'} = \sum_{r's'} \frac{n_{r's'}}{n_{rs}} p_{rs} = \frac{p_{rs}}{n_{rs}} \sum_{r's'} n_{r's'}$$

Ergo we can finally conclude that the probabilities that maximize the probability function

of our table is

$$\hat{p}_{ij} = \frac{n_{ij}}{n}$$

On the other hand, our null hypotheses is

$$H_0 : p_{ij} = p_{i\cdot}p_{\cdot j} \qquad \text{for every } i, j$$

And, under H_0, the probability of obtaining the previous contingency table is

$$n! \prod_{i=1}^{k} \prod_{j=1}^{r} \frac{1}{n_{ij}!} p_{i\cdot} p_{\cdot j}^{n_{ij}} = n! \frac{1}{\prod_{i,j} n_{ij}!} \prod_{i=1}^{k} p_{i\cdot} \prod_{j=1}^{r} p_{\cdot j}^{n_{ij}}$$

And following a similar reasoning we obtain that the maximum probability function of our table is reached at the points

$$\hat{p}_{i\cdot} = \frac{n_{i\cdot}}{n},\ \hat{p}_{\cdot j} = \frac{n_{\cdot j}}{n}.$$

That way, using the notation in Appendix A.1, the likelihood ratio for the contrast of H_0 is

$$\Lambda := \frac{\max_{\theta \in \Theta_0} f_\theta(x_1, ..., x_n)}{\max_{\theta \in \Theta} f_\theta(x_1, ..., x_n)} = \frac{\prod_{i=1}^{k} (n_{i\cdot}/n)^{n_{i\cdot}} \prod_{j=1}^{r} (n_{\cdot j}/n)^{n_{\cdot j}}}{\prod_{i,j} (n_{ij}/n)^{n_{ij}}}$$

Ergo, using the theorem in A.1,

$$D = -2 \log \Lambda = 2 \sum_{i,j} n_{ij} \Big(\log \hat{p}_{ij} - \log(\hat{p}_{i\cdot}\hat{p}_{\cdot j}) \Big) \xrightarrow{d_{\theta_0}} \chi^2_{k-q}$$

That way we have concluded the theoretical aspects of the χ^2 independence test.
Though, this last statistic might seem strange for people who are unfamiliar with statistics. This is because the traditional statistic, discovered by Pearson, as explained in [Vélez Ibarrola and Pérez, 2013], page 432, is another one that is very similar to this one obtained. Despite D shows to have all the features we would ask to a statistic to be able to define a critical set, most of basic books stick to the traditional Pearson statistic:

$$D' = \sum_{i,j} \frac{(n_{ij} - n_{i\cdot}n_{\cdot j}/n)^2}{n_{i\cdot}n_{\cdot j}/n}$$

Continuous Conditional Independence Tests

We will suppose two time series $\{X_t\}, \{Y_t\}$ and a set of conditioning time series $\{\boldsymbol{Z}_t\} = \{Z_t^1, ..., Z_t^k\}$. More information about these algorithms can be consulted in [Runge et al., 2019, Appendix S2].

Partial Correlation

For Partial Correlation (ParCorr) test, multivariate regression (generally linear) of X_t and Y_t, separately, is performed on $\boldsymbol{Z}_t$, followed by a correlation test on residuals. This implies that we are contrasting if the information about X_t and Y_t that $\boldsymbol{Z}_t$ does not have is correlated.

To perform this test it is necessary to assume a multivariate Gaussianity of data and linear dependencies.

GPDC

Gaussian Process Distance Correlation (GPDC) performs a Gaussian process regression similarly to ParCorr, and tests for the independence of uniformized residuals with the *distance correlation coefficient*. The estimator for distance correlation is explained in [Runge et al., 2019, Appendix S2.2].

This test relaxes the linearity assumption of ParCorr, but still assumes multivariate Gaussianity, and assumes additive noise.

CMI

Conditional Mutual Information (CMI) is an information-theoretic measure that captures any kind of dependency by quantifying the reduction in uncertainty about one variable given the knowledge of another while conditioning on a third:

$$I(X_t; Y_t|\boldsymbol{Z}_t) = \int_{\mathbb{R}^{dim(\boldsymbol{Z}_t)}} \int_{\mathbb{R}} \int_{\mathbb{R}} f_{X,Y,Z}(x, y, \boldsymbol{z}) \log \frac{f_{X,Y|\boldsymbol{Z}}(x, y|\boldsymbol{z})}{f_{X|\boldsymbol{Z}}(x|\boldsymbol{z}) f_{Y|\boldsymbol{Z}}(y|\boldsymbol{z})} dx dy d\boldsymbol{z}$$

which is 0 iff $X_t \perp\!\!\!\perp Y_t|\boldsymbol{Z}_T$ (under independency assumption, $\frac{f_{X,Y|Z}(x,y|z)}{f_{X|Z}(x|z)f_{Y|Z}(y|z)} = \frac{f_{X|Z}(x|z)f_{Y|Z}(y|z)}{f_{X|Z}(x|z)f_{Y|Z}(y|z)} = 1$).

CMI is more flexible and robust than previous methods, while it still assumes

stationarity, and in general, as most non-parametric tests, it needs a lot of data to be reliable, what contrast with the much higher computational resources it takes.

A.5. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a classical dimensionality reduction method, based on extracting linear combinations of input variables in such a way that all linear correlations are deleted and the highest possible variance from data is explained.

Let $\boldsymbol{X} = (\boldsymbol{x}_i)_{i=1,...,n}$ be a dataset, where each column vector represents a variable and each row an instance. To extract these linear combinations, first, the covariance matrix is calculated as seen in Definition A.4,

$$\boldsymbol{\Sigma} = \big(COVAR(\boldsymbol{x}_i, \boldsymbol{x}_j)\big)_{i,j} = \Big(\sum_k \frac{(x_{i,k} - \bar{x}_i)(x_{j,k} - \bar{x}_j)}{n}\Big)_{i,j} = \boldsymbol{X}\boldsymbol{X}^t.$$

Since $\boldsymbol{\Sigma}$ is symmetric[10] we can calculate its eigenvectors $\boldsymbol{v}_1, ..., \boldsymbol{v}_n$, associated to their eigenvalues $\lambda_1, ..., \lambda_n$. We would like to extract the vectors that have the highest possible amount of variance along all directions[11],

$$\boldsymbol{v}_1 = \arg\max_{||\boldsymbol{w}||=1} \boldsymbol{w}^t \boldsymbol{\Sigma} \boldsymbol{w}$$

Rayleigh-Ritz Theorem, in [Zielke, 1987, Section 4.2], shows that the eigenvector of $\boldsymbol{\Sigma}$ with the highest eigenvalue reaches precisely this maximum, and the following eigenvectors with highest eigenvalues are the orthogonal vectors that represent the highest possible variance at each point. Also, the proportion of variance explained by the first k eigenvectors is $\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$.

That way, constructing the principal components matrix $\boldsymbol{V}_k := (\boldsymbol{v}_1, ..., \boldsymbol{v}_k)$, we can obtain the projection of $\boldsymbol{X}$ onto the principal components as

$$\boldsymbol{Z}_k := \boldsymbol{X}\boldsymbol{V}_k.$$

[10] $COVAR(X, Y) = COVAR(Y, X)$.

[11] $\boldsymbol{\Sigma}$ can be interpreted as a quadratic form that returns the variance of the data along a vector, $Var(\boldsymbol{X}\boldsymbol{w}) = \boldsymbol{w}^t \boldsymbol{\Sigma} \boldsymbol{w}$.

B. Additional Resources

All graphics, visualizations, and results presented in this project have been generated using code or procedures executed within open source software environments. These tools are distributed under open licenses, and every effort has been made to ensure compliance with their respective licensing terms.

The node-edges static graphs have been generated using LaTeX, and the code can be consulted in `https://github.com/JoaquinMateosBarroso/Causal-Inference/tree/main/DocumentAuxiliars/Graphs/latex`.

Time series graphs have been generated in Python, with the libraries PyGraphViz [PyGraphViz, 2014] and Tigramite [Runge et al., 2025], and the code can be consulted in `https://github.com/JoaquinMateosBarroso/Causal-Inference/tree/main/DocumentAuxiliars/Graphs/tigramite`.

Graphs with results of experiments have been generated using the Python libraries Seaborn [Waskom, 2021] and Matplotlib [Hunter, 2007], inside our library Group-causation, and the code used to generate both experiments and graphics can be consulted in `https://github.com/JoaquinMateosBarroso/Causal-Inference/tree/main/experiments`.

C. Web Interface

In order to make more accessible the use of this algorithm to people with non-technical formation, we have implemented a web server with the basic functions needed to perform time series causal discovery from different types of time series, using our library *group-causation*, with the help of Github Copilot, [GitHub, 2021]. Its code and instructions can be consulted in `https://github.com/JoaquinMateosBarroso/Causal-Inference/tree/main/web`.

C.1. Implementation and Architecture

The backend of the application is implemented in FastAPI, chosen for its combination of high performance and modern Python features. It allows to easily use `async def` endpoints for non-blocking I/O, it has automatic generation of interactive API docs, and fewer runtime errors. Together, these qualitiesasync support, type-annotated data handling, and simple Python integration make FastAPI a natural choice for serving computationally intensive AI services [Ramírez, 2023, Bandurchin, 2024].

Throughout development, FastAPIs automatic validation and documentation (OpenAPI/Swagger) simplify testing and iteration. The combination of a modular asynchronous backend and simple Jinja2 frontend allows researchers to quickly deploy a web interface for causal analysis while relying on Python end-to-end [Ramírez, 2023, Stack, 2025, for Geeks, 2025].

C.2. User Guide

The web application provides an intuitive interface for performing causal discovery on time series data and groups of time series. Upon accessing the application, the user is presented with a main page structured in three columns, each corresponding to a core functionality:

- **Causal Discovery**
- **Benchmarks**
- **Toy Datasets Generation**

Each column features two options: **Time Series** and **Groups of Time Series**, leading to dedicated modules as described below.

C.2.1. Causal Discovery

This module allows users to upload their own data and apply causal discovery algorithms.

Time Series

Users must upload a CSV file containing the time series data. Then, they can select a causal discovery algorithm from the available list and configure its corresponding parameters. Once executed, the system outputs the estimated causal graph in the form of a directed acyclic graph (DAG), provided as an image.

Groups of Time Series

This option extends the previous functionality to handle grouped time series. In addition to uploading a CSV file and specifying the algorithm and parameters, users can define the grouping structure interactively using draggable boxes. The resulting group-level causal DAG is then generated and returned as a PNG.

C.2.2. Benchmarks

In this module, users can evaluate the performance of different causal discovery algorithms.

Input

The user must upload a ZIP file containing one or more datasets. This ZIP file can be obtained through the *Toy Datasets Generation* module. The user then selects multiple algorithms and configures their parameters.

Output

The system runs each algorithm on the datasets and returns performance evaluation metrics, including:

- F1-score
- Structural Hamming Distance (SHD)
- Precision
- Recall
- Time (s)
- Memory

The results are displayed as a series of plots for easy comparison.

C.2.3. Toy Datasets Generation

This module allows users to generate synthetic datasets with controlled properties.

Functionality

Users specify the desired parameters (e.g., number of variables, time steps, sparsity, noise level, etc.) to generate datasets suitable for causal discovery experiments. The resulting dataset is downloadable as a ZIP file and can be used directly in the *Benchmarks* module.

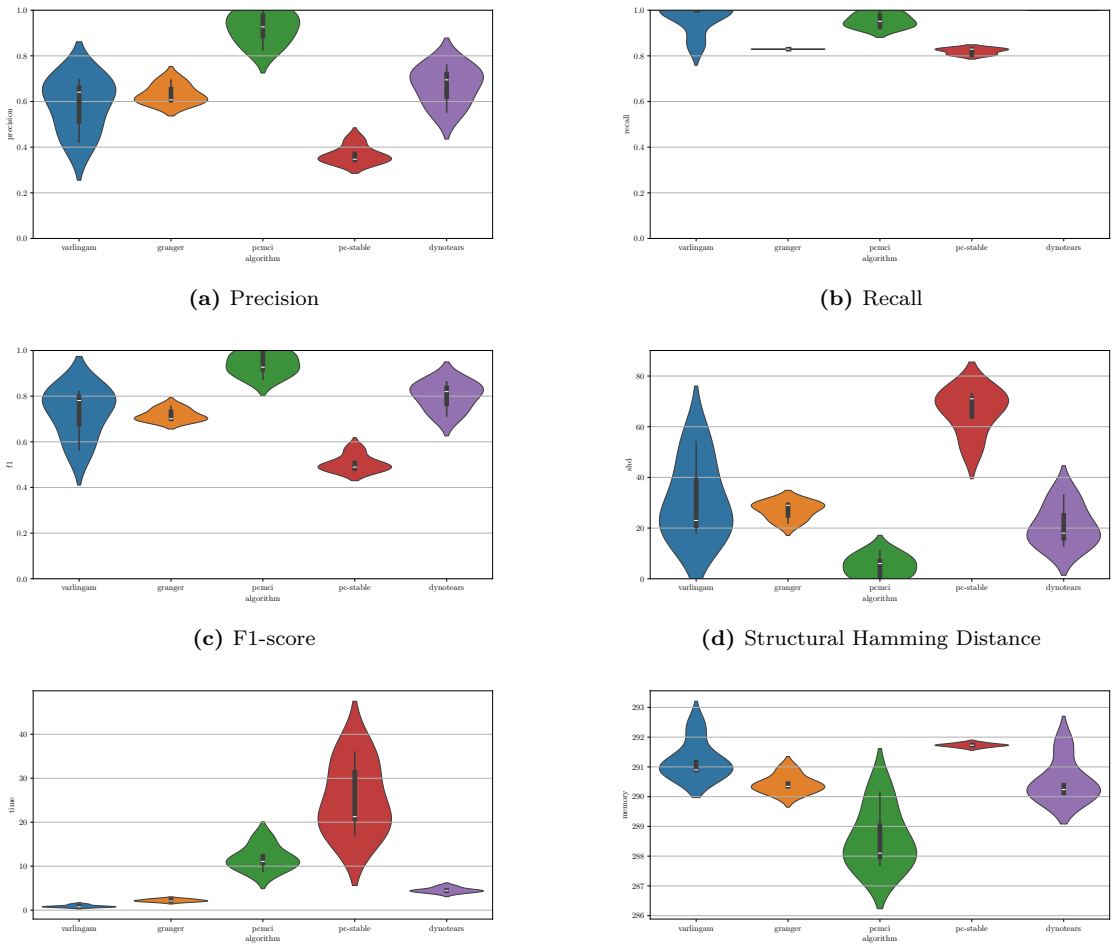
D. Micro Time Series Causal Discovery Benchmark

Some first experiments of applying our benchmark to single time series are shown in this Appendix.

1. A benchmark with a low dimensionality (10 variables) and just linear relations is shown in Figure D.1. These metrics are obtained from 10 executions of different causal discovery methods for a set of 10 time series. Boxplots are shown in inner black boxes and in color there are shown the distributions of the metrics. Datasets have been generated synthetically through methods explained in Section 2.3. Variables have linear, negative-exponential and trigonometric relations and noise is randomly taken from gaussian and weibull, with standard deviation of 0.2. The contemporaneous crosslinks density has been fixed to 0.25, autocorrelation coefficients are 0.5 and dependency coefficients are randomly chosen in $\{-0.3, 0.3\}$. Algorithms are using the recommended hyperparameters in original papers. Hybrid approaches follow the parameters recommendations in Chapter 4.

2. A benchmark with a high dimensionality (80 variables) and more complex relations is shown in Figure D.2. These metrics are obtained from 10 executions of different causal discovery methods for a set of 80 time series. The rest of the parameters are the same as in previous experiment.

3. A benchmark increasing the number of variables (from 10 to 80) and complex relations is shown in Figure D.3. To obtain these metrics, each point is averaged from 10 executions, of different causal discovery methods for a set of an increasing number of time series (N_variables). The rest of the parameters are the same as in previous experiment.



(a) Precision

(b) Recall

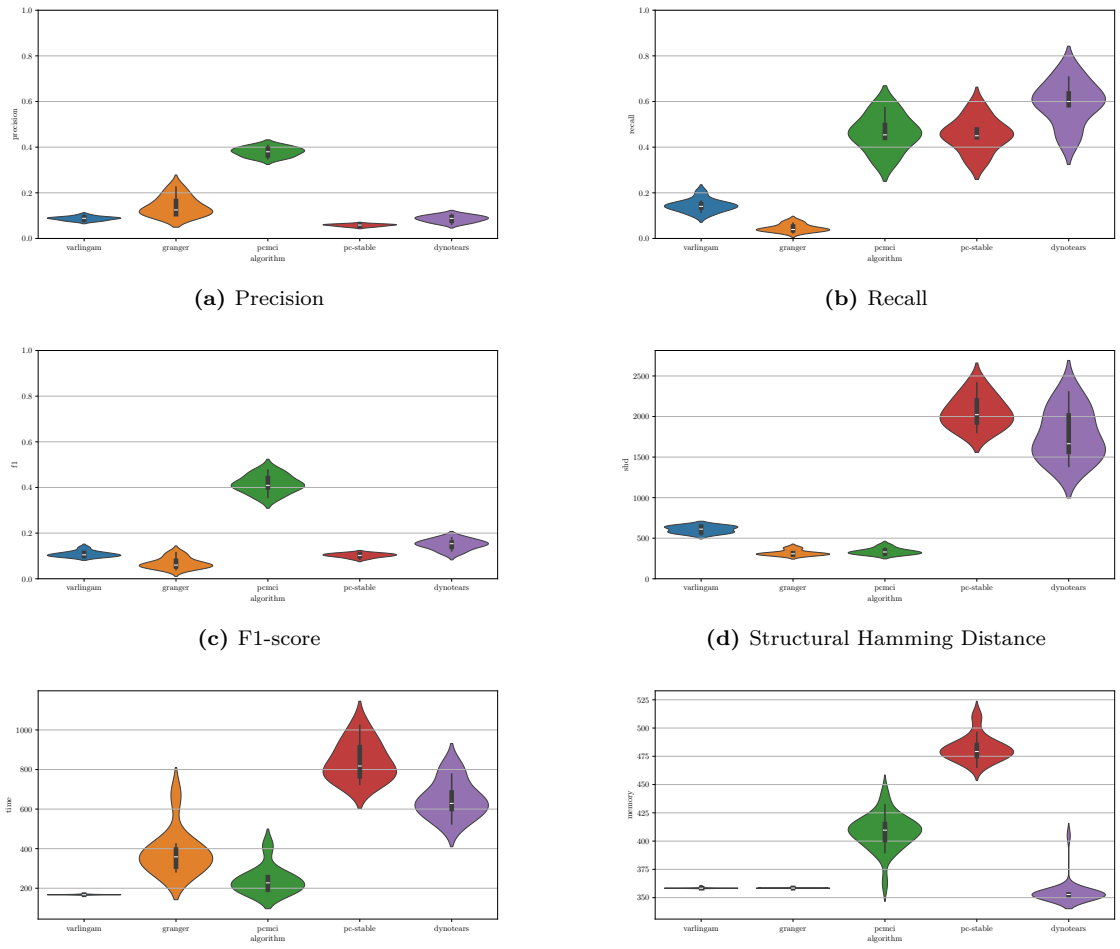
(c) F1-score

(d) Structural Hamming Distance

(e) Time (seconds)

(f) Memory (megabytes)

Figure D.1: Violin plot with the low dimensionality micro time series causal discovery benchmark.



(a) Precision

(b) Recall

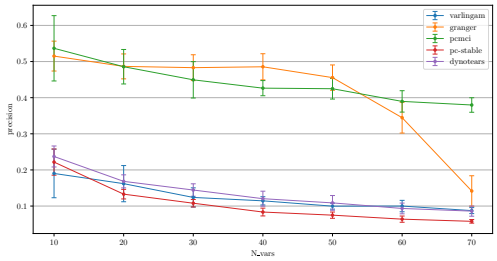
(c) F1-score

(d) Structural Hamming Distance

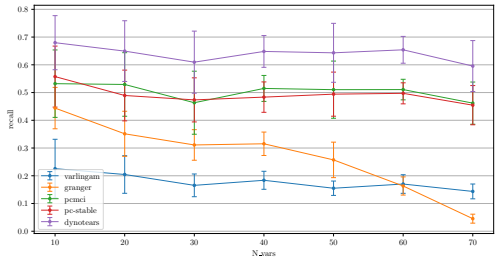
(e) Time (seconds)

(f) Memory (megabytes)

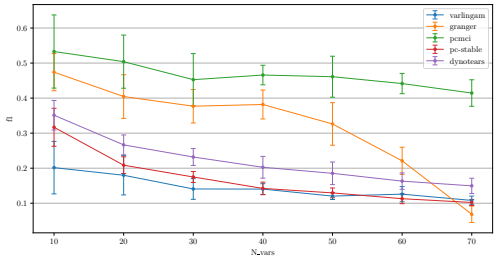
Figure D.2: Violin plot with the high dimensionality micro time series causal discovery benchmark.



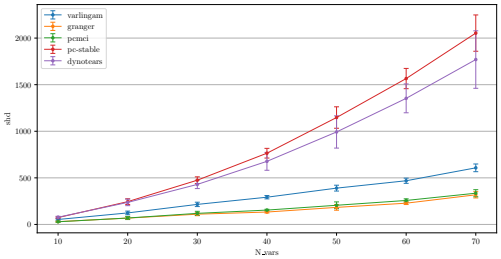
(a) Precision



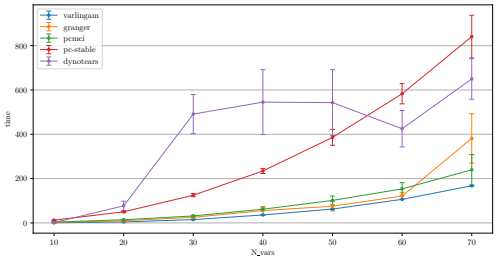
(b) Recall



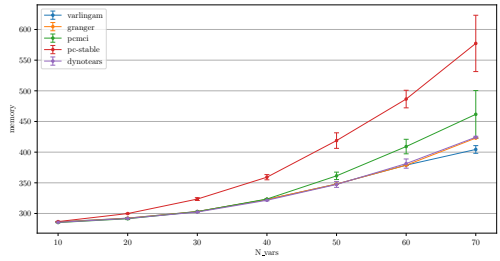
(c) F1-score



(d) Structural Hamming Distance



(e) Time (seconds)



(f) Memory (megabytes)

Figure D.3: Plot with an increasing number of variables for the micro time series causal discovery benchmark. Average points $\pm std$ are shown for each algorithm.