Causal Discovery for Non-stationary Non-linear Time-series Data Using Just-In-Time Modeling

Author names withheld

Editor: Under Review for CLeaR 2023

Abstract

Causal discovery from multivariate continuous time-series data is becoming more important as the amount of IoT data to analyze increases. However, it is not easy to identify the causal structure from such data using conventional linear causal discovery methods due to their non-stationary characteristics such as distribution shifts, and non-linearity of the system dynamics. The application of non-linear causal discovery methods is also generally limited, and there are still some problems such as their computational complexity, interpretability, and robustness for non-stationarity. To address these challenges, we propose a new causal discovery method JIT-LiNGAM, based on the Linear Non-Gaussian Acyclic Model (LiNGAM) and the Just-In-Time (JIT) framework, which is also called Lazy-Learning or Model-On-Demand. Our method estimates a local linear structural causal model from neighboring samples of the past data every time a new input sample is given. Approximating an inherently globally non-linear model with local linear models, we can benefit from high detection performance of causal relationship for non-linear and non-stationary data, improvements of interpretability of causal effects by linear expression, and reduced computational complexity. We formulate this algorithm based on Taylor's theorem, and show effective neighbor selection algorithms by a simple experiment. The results of numerical experiments using artificial data with non-linearity and non-stationarity demonstrate the effectiveness of our method compared to representative methods for such data, under some general evaluation metrics.

Keywords: Causal Discovery, LiNGAM, Just-In-Time Modeling, Non-Stationarity, Non-Linearity, Time-series

1. Introduction

The importance of time-series data analysis in the manufacturing industry is rapidly increasing due to its broad useful applications (quality prediction, anomaly/failure detection of equipment, automation of operations, etc.) and the increase of the amount of IoT data to analyze. Causal analysis approaches is also useful to addressing these industrial real-world problems, as it enables to clarify and utilize the causal structure of the target system. For example, with the framework of statistical causal inference as a basis, causal relationships can be utilized to determine how the entire data is affected when a particular variable is changed independently of other variables (Pearl et al. (2016)). This treatment is called intervention, and its resulting effects are called the interventional effect. Example applications of the intervention technique include optimizing the amount of material input in a chemical plant by calculating backwards, or utilizing the revealed causal relationships as a reasoning for later business decision making. The knowledge of causal structure also improve the interpretability, i.e., how easily machine learning model's outputs, its reasons and behavior are understood by humans. It is particularly important e.g. in industrial plants due to the potential economic and safety repercussions should an accident occur.

When causal relationships are unknown, we usually estimate them by conducting experiments such as Randomized Controlled Trials (RCTs), where only a certain variable is manipulated and we

© 2023 .

observe the changes in the other variables. However, in many real-world use cases, it is difficult to conduct such an experiment due to practical limitations such as budget, ethical issues, or safety. Causal discovery is a framework for identifying causal relationships only from data in cases like this where experimentation is not practical. The application of causal discovery to time-series data also has been studied extensively in recent years (Hyvärinen et al. (2010)).

One challenge here is that the time-series data appearing in practical applications often violate the ideal conditions assumed by the conventional methods of time-series analysis and causal discovery. For example, in time-series analysis, many models assume stationarity, i.e., that there are no time-dependent trends or seasonal components and that the dynamics and state of the system remain constant over time. This is problematic because actual data may in fact have seasonal deviations in sensor values due to temperature changes, or discrete changes in operating conditions such as changes in production amount or trend components due to aging of the plant.

One of the most representative causal discovery methods is the Linear Non-Gaussian Acyclic Model (LiNGAM) (Shimizu et al. (2006); Shimizu et al. (2011)), which assumes a linear model for the structural equations that express causal relationships. However, in reality, many of the physical phenomena that occur in plants and other facilities are non-linear, so a linear model can only capture causal relationships as a limited approximation. Non-linear causal discovery methods have problems too: for example, it is often difficult for human operators to interpret the causal relationships implicitly expressed by non-linear functions (e.g., MultiLayer Perceptron (MLP)), and their computational complexity of training is extremely high. As is well known, MLP-based models (Zheng et al. (2020); Uemura et al. (2022)) take a long time to train, and REgression with Subsequent Independence Test (RESIT) (Peters et al. (2014)) also has computational complexity because it requires repeated nonlinear regression and examinations of independence for each combination of variables to detect causality. Furthermore, when these methods are extended to time-series analysis, the validity is not necessarily guaranteed for non-stationary data. One of the methods having similar problem setting of robustness for non-stationarity is Constraint-based causal Discovery from heterogeneous/NOnstationary Data (CD-NOD) (Huang et al. (2020)). However, it has another problem that if none of the certain identifiability conditions given in this theory are satisfied, the causal direction of the relationship between two variables suggested by former half of this algorithm remains undetermined with the algorithm of CD-NOD, and an extra direction detect method (e.g. LiNGAM) is needed.

Summery of This Paper

In response to these challenges, we have developed a locally linear (but globally non-linear) causal discovery method that is robust to non-stationary and non-linear time-series data. The proposed method, JIT-LiNGAM, approximates the inherently globally non-linear structural equations by local linear models. Each local linear model is trained in the neighborhood of the query data $\boldsymbol{x}_{\mathrm{Q}}$, located at where we want to know the causality. This query point is given for each estimation, for example, if data are available sequentially in a time-series, the most recently observed data can be treated as the query data. The idea of this local model is based on the fact that a non-linear function can generally be approximated to a linear function from Taylor's theorem if a sufficiently small neighborhood is taken as the neighborhood of the query point. Furthermore, by taking the neighborhood, we also expect proposed method to be robust for non-stationarity, for example, involving multimodal distribution shifts. The learning of theses local linear causal models can be easily solved by a

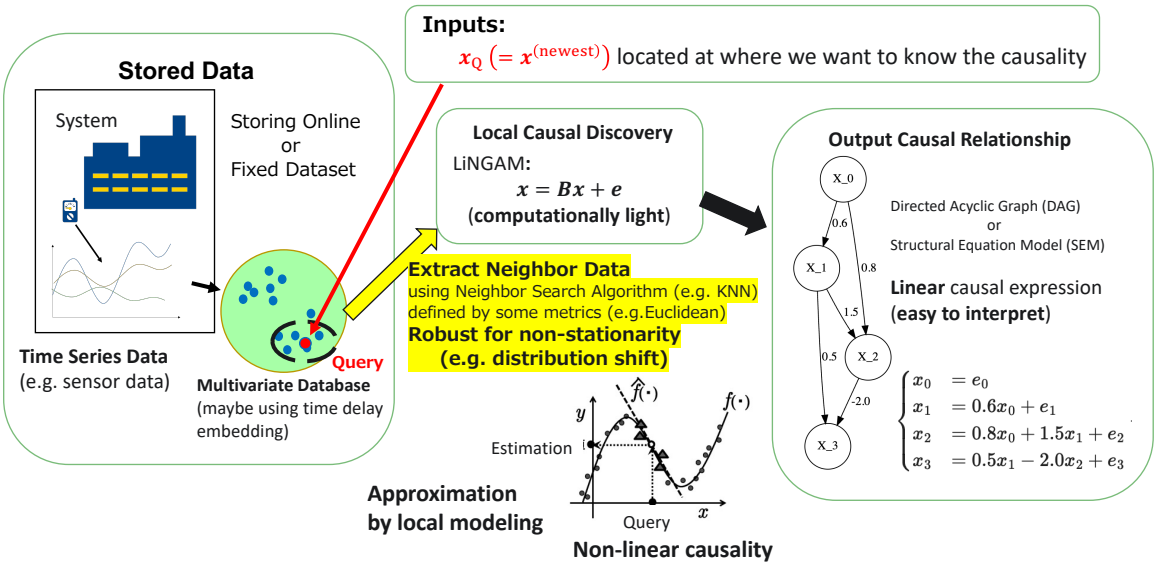

Figure 1: The basic idea of the proposed method.

conventionally used model, LiNGAM. The basic idea of the proposed method is shown in Figure 1.

The main contributions of this paper is below:

- We propose a new causal discovery method having a following features:
  - Strong detection capability for non-linear causality, due to approximation to local linear models
  - Robustness for non-stationary characteristics such as distribution shifts by using neighboring training data around query point
  - Simple algorithms and less computational complexity based on LiNGAM
  - High interpretability of causal representation of local linear functions, with respect to its strength, its evolution over time, etc.
  - Fully directed causal relationships (unlike CD-NOD)
- We show the rough formulation and the application range of our method.
- We suggest the effective way of selecting neighbors especially for causal discovery by a simple experiment.
- We conduct a further experiment using more complex non-linear non-stationary artificial data to confirm effectiveness of our method compared to conventional methods.

2. Related Work

2.1. Analysis for Non-stationary Non-linear Time-series Data (JIT)

As discussed above, the analysis of non-stationary and non-linear time-series data, which often appears in practical data such as the sensor values of industrial plants, is generally difficult. One

framework for resolving this difficulty is the Just-In-Time (JIT) model (Stenman et al. (1996)), which is also called Lazy-Learning (Bontempi et al. (1999)) or Model-On-Demand (Braun et al. (2001)). JIT has been utilized in the context of soft sensors in industrial plants (e.g., pseudo-sensors for difficult-to-measure locations using regression estimation), for forecasting problems, and for system identification and control engineering. It is also known to be robust to non-stationary and non-linear time-series data as appearing in industrial plants.

The basic idea of JIT is simple. Based on an unknown non-linear function $f(\cdot) : \mathbb{R}^P \to \mathbb{R}$, the dataset of inputs and outputs $\mathcal{D} = \left\{(\boldsymbol{x}^{(i)}, y^{(i)}) \mid i = 1, \ldots, N\right\}$ is obtained as

$$y^{(i)} = f(\boldsymbol{x}^{(i)}) + \epsilon^{(i)}, \tag{1}$$

where $\epsilon^{(i)}$ is the noise assuming i.i.d. for i. With JIT, instead of regressing an unknown (non-linear) function $f(\cdot)$ directly from $\mathcal{D}$, we use K-neighbor-data, which is a subset of $\mathcal{D}$ composed of inputs $\boldsymbol{x}^{(i)}$ $(i = 1, 2, \ldots, K)$ neighboring $\boldsymbol{x}_{\mathrm{Q}}$ and the corresponding outputs $y^{(i)}$. Using these neighboring data, we can regress the local function $\hat{f}_{\boldsymbol{x}_{\mathrm{Q}}}(\cdot)$ and then obtain the resulting estimation of $f(\boldsymbol{x}_{\mathrm{Q}})$ by $\hat{f}_{\boldsymbol{x}_{\mathrm{Q}}}(\boldsymbol{x}_{\mathrm{Q}})$. $\boldsymbol{x}_{\mathrm{Q}}$ is called a query point. We usually utilize a linear model as the regression method for $\hat{f}_{\boldsymbol{x}_{\mathrm{Q}}}(\cdot)$ because, by taking a local neighborhood, we should be able to approximate the globally non-linear function to a linear model around the query point based on Taylor's theorem. In addition, using a linear model means we can more easily interpret the output process of the model. Note that, in original Just-In-Time modeling for regression, K-Nearest-Neighbor is normally adopted as the way of selecting neighboring data. When predicting multiple test outputs $y_i = f(\boldsymbol{x_i})$ for inputs $\boldsymbol{x_i}$ $(i = N+1, N+2, \ldots,)$, local linear functions are learned repeatedly in JIT in the procedure above.

JIT can be applied to the regression of time-series data as shown in previous studies. In these time-series extension, we apply Algorithm 2 (showed in Appendix A.) repeatedly to time-series data obtained at discrete times $(t = 1, 2, \ldots, T)$ updating $\mathcal{D}$ and $\boldsymbol{x}_{\mathrm{Q}}$. And sometimes time delay embedding (i.e., making multivariate vector including delayed variable cut out by time window) is used as the input vector $\boldsymbol{x}_i$ there. By appropriately setting future value of target variable as the output $y^{(T)}$, Auto Regressive (AR) models and Vector Auto Regressive (VAR) models can be easily handled within the JIT framework. A similar idea of embedding can be applied to the time-series extension of our proposed method (please refer to Hyvärinen et al. (2010)), but we would like to consider this extension as a future work. JIT is not only expected to approximate non-linear functions linearly: in addition, when applied to time-series data in this way, it should be able to follow time-series data with non-stationarity characteristics such as discrete state changes, seasonality, and trends thanks to taking a neighboring data of the current system state $\boldsymbol{x}^{(T)}$.

2.2. Linear Causal Discovery (LiNGAM)

The (non-linear) Structural Equation Model (SEM), which is one of the most general formulations of causality and describes the causal relationship between variables, is written as

$$x_i = f_i(\boldsymbol{x}_{\backslash i}, e_i) \quad (i = 1, 2, \ldots, P), \tag{2}$$

where e_i means noise and $\boldsymbol{x}_{\backslash i}$ means a vector made by removing only x_i as an element from $\boldsymbol{x} = (x_1, x_2, \ldots, x_P)^{\mathrm{T}}$. Note that $\boldsymbol{x}_{\backslash i}$ in some causal models (e.g., LiNGAM and ANM, as discussed below) include only parental variables of x_i in the meaning of causality. $f_i(\cdot)$ is a non-linear function

that represents the causal relationship from each variable to x_i. A common task setting for causal discovery is to find this structural equation and the functions $f_i(\cdot) \quad (i = 1, 2, \ldots, P)$ from the data only.

LiNGAM further utilize the following linear structural equation model:

$$x_i = \sum_{j=1, j \neq i}^{P} b_{ij} x_j + e_i \quad (i = 1, 2, \ldots, P). \tag{3}$$

In LiNGAM, e_i is assumed to be a non-Gaussian noise independent for each i whose mean zero. It can be written in matrix form as

$$\boldsymbol{x} = \boldsymbol{B}\boldsymbol{x} + \boldsymbol{e}. \tag{4}$$

It is also assumed that when $\boldsymbol{B}$ is seen as the adjacency matrix of a weighted directed graph, the graph is a Directed Acyclic Graph (DAG). When $b_{ij} \neq 0$ holds, it means there is a causal relationship in the direction of $x_j \to x_i$. Causal discovery in LiNGAM is equivalent to finding this $\boldsymbol{B}$.

3. Proposed Method

3.1. Formulation of JIT-LiNGAM

The JIT framework has a simple structure consisting of Neighbor-Search and sequential local model learning, so it can also be widely applied to problem settings other than regression if their algorithm solving is computationally light. As mentioned above, most real time-series data are considered to have non-linear causal relationships between variables. We can also interpret these non-linear relationships in terms of the continuously changing strength of the (linear) relationships corresponding to the values of the variables. On the basis of this idea, we construct a new theory of non-linear causal discovery under the JIT framework and propose a natural extension of JIT to the causal discovery problem. The key point of our method is that it approximates causality based on non-linear structural equations to a local linear structural equation by using neighborhood data and LiNGAM.

First, we consider Additive Noise Model (ANM) (Hoyer et al. (2008)), a non-linear structural equation model that restricts Equation (2) to additive noise. This is formulated as

$$x_i = f_i(\boldsymbol{x}_{\setminus i}) + e_i \quad (i = 1, 2, \ldots, P). \tag{5}$$

It is also written in vector form, as

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{e}, \tag{6}$$

where e_i is assumed to be a non-Gaussian noise independent for each i whose mean is zero. Note that in ANM, the causal relationship defined by $\boldsymbol{f}$ is restricted to directed acyclic models in the meaning of the causal graph. In $B(\boldsymbol{x}_{\mathrm{Q}}, \varepsilon) = \{\boldsymbol{x} \mid d(\boldsymbol{x}, \boldsymbol{x}_{\mathrm{Q}}) < \varepsilon\}$, which is the neighborhood around the query point $\boldsymbol{x}_{\mathrm{Q}}$, we obtain the following approximated equation based on Taylor's theorem:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{f}(\boldsymbol{x}_{\mathrm{Q}} + (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{Q}})) + \boldsymbol{e} \\ &\simeq \boldsymbol{f}(\boldsymbol{x}_{\mathrm{Q}}) + \boldsymbol{J}(\boldsymbol{x}_{\mathrm{Q}})(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{Q}}) + \boldsymbol{e}, \end{aligned} \tag{7}$$

where $\boldsymbol{J}(\cdot)$ is Jacobian. Then, we take the expected value of both sides of Equation (7) by the truncated distribution in $B(\boldsymbol{x}_{\mathrm{Q}}, \varepsilon)$, which is a neighborhood of $\boldsymbol{x}_{\mathrm{Q}}$. When we set each expected value of $\boldsymbol{x}$ and $\boldsymbol{e}$ in the neighborhood as $\bar{\boldsymbol{x}} = \mathbb{E}_{B(\boldsymbol{x}_{\mathrm{Q}},\varepsilon)}[\boldsymbol{x}]$ and $\bar{\boldsymbol{e}} = \mathbb{E}_{B(\boldsymbol{x}_{\mathrm{Q}},\varepsilon)}[\boldsymbol{e}]$, the resulting equation is

$$\bar{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}_{\mathrm{Q}}) + \boldsymbol{J}(\boldsymbol{x}_{\mathrm{Q}})(\bar{\boldsymbol{x}} - \boldsymbol{x}_{\mathrm{Q}}) + \bar{\boldsymbol{e}}. \tag{8}$$

Therefore, subtracting both sides of Equation (8) from Equation (7), we have

$$\tilde{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{x}_{\mathrm{Q}})\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{e}}, \tag{9}$$

with $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \bar{\boldsymbol{x}}$, $\tilde{\boldsymbol{e}} = \boldsymbol{e} - \bar{\boldsymbol{e}}$. This has the same form as Equation (4) (by utilize the assumption of the acyclic causality in ANM, $\boldsymbol{J}(\boldsymbol{x}_{\mathrm{Q}})$ also becomes DAG as a adjacency matrix), and can be solved by the LiNGAM algorithm to obtain the matrix $\boldsymbol{J}(\boldsymbol{x}_{\mathrm{Q}})$.

In actual data analysis, $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \bar{\boldsymbol{x}}$ can be obtained by sample approximation (centralization of neighbor data). Then $\tilde{\boldsymbol{e}}$ is not calculated explicitly but automatically treated as non-Gaussian noise whose mean is zero in the LiNGAM algorithm. We now regard this resulting $\boldsymbol{B} = \boldsymbol{J}(\boldsymbol{x}_{\mathrm{Q}})$ given by LiNGAM as the linearly approximated causality in the neighborhood of the query point $\boldsymbol{x}_{\mathrm{Q}}$ of the original non-linear structural equation (Equations (5) and (6)). In addition, by using the linearly approximated output $\boldsymbol{B} = \boldsymbol{J}(\boldsymbol{x}_{\mathrm{Q}})$, we can more easily interpret the causality in term of its positive/negative effect, its strength, and its change over time. Note that the computational complexity of the causal discovery part in JIT-LiNGAM fully depends on this part (the details is explained Section 3.2). From the above, the general form of the proposed algorithm JIT-LiNGAM can be summarized as Algorithm 1.

3.2. How to Select Neighboring Set

We examined the following ways for selecting the neighboring set Ω.

- $\Omega_{\mathrm{KNN}}(\boldsymbol{x}_{\mathrm{Q}}; d_{\mathrm{E}}, K)$: K-Nearest-Neighbors from the query point measured by Euclidean distance.
- $\Omega_{\mathrm{KNN}}(\boldsymbol{x}_{\mathrm{Q}}; d_{\mathrm{M}}, K)$: K-Nearest-Neighbors from the query point measured by Mahalanobis distance.
- $\Omega_{\mathrm{ERN}}(\boldsymbol{x}_{\mathrm{Q}}; d_{\mathrm{E}}, K, \varepsilon)$: K-Neighbors randomly selected from samples whose distance from the query point measured by Euclidean distance is smaller than ε.
- $\Omega_{\mathrm{ERN}}(\boldsymbol{x}_{\mathrm{Q}}; d_{\mathrm{M}}, K, \varepsilon)$: K-Neighbors randomly selected from samples whose distance from the query point measured by Mahalanobis distance is smaller than ε.

(KNN : K-Nearest-Neighbors, ERN : Epsilon-Random-Neighbors)

The Euclidean and Mahalanobis distances are defined as

$$d_{\mathrm{E}}(\boldsymbol{x}, \boldsymbol{x}_{\mathrm{Q}}) = \sqrt{(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{Q}})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{Q}})},$$
$$d_{\mathrm{M}}(\boldsymbol{x}, \boldsymbol{x}_{\mathrm{Q}}) = \sqrt{(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{Q}})^{\mathrm{T}} W (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{Q}})},$$

Algorithm 1 JIT Algorithm for Time-Series Causal Discovery (JIT-LiNGAM)

Inputs:

stored data $\mathcal{D} = \left\{\boldsymbol{x}^{(t)} \mid t = 1, \ldots, T-1\right\}$,

query point $\boldsymbol{x}_{\mathrm{Q}} = \boldsymbol{x}^{(T)}$, distance function$d(\cdot, \cdot)$, number of neighbors K.

Outputs:

weighted adjacency matrix $\boldsymbol{J}(\boldsymbol{x}^{(T)})$: representing the causality defined in the neighborhood for query point $\boldsymbol{x}^{(T)}$.

Procedure 1

Extract K-data of $\boldsymbol{x}^{(t)}$ from $\mathcal{D}$, based on $d(\boldsymbol{x}^{(t)}, \boldsymbol{x}_{\mathrm{Q}})$, which is the distance from the query point $\boldsymbol{x}_{\mathrm{Q}}$. (The details of how to extract K-data are described in Section 3.2.) The resulting K-data subset $\Omega(\boldsymbol{x}_{\mathrm{Q}}; d, K)$ is:

$$\Omega(\boldsymbol{x}_{\mathrm{Q}}; d, K) = \left\{\boldsymbol{x}^{(\sigma(k))} \mid k = 1, \ldots, K\right\},$$

where $\sigma(k)$ is a function that returns the k-th nearest time index t in $\Omega(\boldsymbol{x}_{\mathrm{Q}}; d, K)$.

Procedure 2

Centralize $\Omega(\boldsymbol{x}_{\mathrm{Q}}; d, K)$ and get $\tilde{\Omega}(\boldsymbol{x}_{\mathrm{Q}}; d, K)$, where mean is subtracted from each element of $\Omega(\boldsymbol{x}_{\mathrm{Q}}; d, K)$ along each dimension of $\boldsymbol{x}$.

Procedure 3

Train LiNGAM using $\tilde{\Omega}(\boldsymbol{x}_{\mathrm{Q}}; d, K)$, and get resulting weighted adjacency matrix $\boldsymbol{J}(\boldsymbol{x}^{(T)})$.

where

$$W = \left(\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{m})(\boldsymbol{x}_i - \boldsymbol{m})^{\mathrm{T}}\right)^{-1},$$

$$\boldsymbol{m} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i.$$

The Mahalanobis distance is a distance metric that considers the correlation between variables by measuring after disentangling it, and it has often been used in the area of anomaly detection.

All the above selection techniques fix the number of neighbors K so as to ensure the upper limit of computational complexity. As a result, when the number of data N is sufficiently larger than the number of variables P, the computational complexity of JIT-LiNGAM mainly depends on the Neighbor-Search part (in KNN, it is $\mathcal{O}(T \log T)$), and the computational complexity of the causal discovery part is the same as the LiNGAM of K-samples.

Note that the algorithm of ERN selects K-data from the neighbor region $B(\boldsymbol{x}_{\mathrm{Q}}, \varepsilon)$, which is closer to the query point than ε, but sometimes the number of elements belonging to $B(\boldsymbol{x}_{\mathrm{Q}}, \varepsilon)$ is less than K and the algorithm breaks. To avoid this problem, we increase the size of ε until the number of elements belonging to $B(\boldsymbol{x}_{\mathrm{Q}}, \varepsilon)$ is more than K, where the value of ε is reset to the initial value every query.

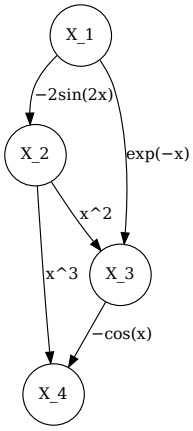

Figure 2: The causal DAG corresponding to the non-linear structural equations in Experiment 1 and 2. Note that implicitly in Experiment 1, two causal effects expressed as two arrows directed to x_3 are added before flow into x_3, but in Experiment 2 they are multiplied.

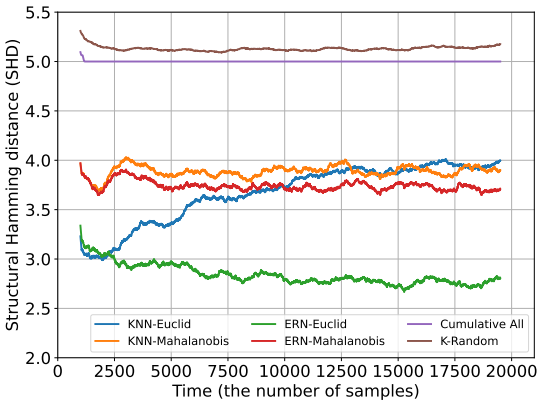

Figure 3: Results of JIT-LiNGAM with various ways of selecting neighbors and the baseline methods in Experiment 1. Moving averages are taken over 1000 samples due to large fluctuations.

4. Experiment

4.1. Simple Independent Data (Experiment 1)

We utilized the following structural equations represented by a non-linear ANM, and we generated $N = 20000$ independent samples:

$$\begin{cases} x_1 = e_1 \\ x_2 = -2\sin(2x_1) + e_2 \\ x_3 = \exp(-x_1) + x_2^2 + e_3 \\ x_4 = x_2^3 - \cos(x_3) + e_4 \end{cases}, \tag{10}$$

where $e_i \sim \mathcal{LA}(0, 0.05)$ is independent Laplace noise for $(i = 1, 2, 3, 4)$. This causal relationship is represented as a causal DAG as shown in Figure 2.

The generated N samples are treated as if a time-series in this experiment although this data is actually i.i.d. data, not time-series. Causal discovery (LiNGAM) at time $T \in [1, N]$ is executed sequentially in the JIT framework (whose number of neighbors $K = 500$) using latests stored data $\mathcal{D} = \left\{ \boldsymbol{x}^{(t)} \mid t = 1, \ldots, T-1 \right\}$ at each time. Here, instead of ICA-based LiNGAM (Shimizu et al. (2006)), we used a method called Direct-LiNGAM (Shimizu et al. (2011)), which repeats regressions and examinations of independence. In Experiment 1, we examined all the ways of selecting neighbors introduced in Section 3.2.

4.2. Complex Non-stationary Time-series Data (Experiment 2)

Further more, we generate $t = 1, 2, \ldots, N$ ($N = 20000$) time dependent samples using the following structural equations extending a non-linear ANM to non-stationary time-series data:

$$\begin{cases} x_1 = z^{(t)} + e_1 \\ x_2 = -2\sin(2x_1) + e_2 \\ x_3 = \exp(-x_1) \cdot x_2^2 + e_3 \\ x_4 = x_2^3 - \cos(x_3) + e_4 \end{cases}, \text{where} \quad z^{(t)} = \alpha t + \beta^{(t)} \quad \beta^{(t)} = \begin{cases} 0 & (1 \leq t < \frac{N}{2}) \\ \beta & (\frac{N}{2} < t \leq N) \end{cases}. \tag{11}$$

Note that in $z^{(t)}$, the first term αt means the time dependent trend and the second term $\beta^{(t)}$ means distribution shift by a bias parameter β occurring at $t = \frac{N}{2}$. It also increases complexity of causality for Equation (11) to have a cross term of $\exp(-x_1) \cdot x_2^2$, which means that the off-diagonal elements of Hessian matrix is nonzero unlike Equation (10). Though there is the cross term and added complexity, the causal relationship is represented as the same causal DAG as Equation (10) showed in Figure 2. This data is processed by the same way as described in Section 4.1 in the experiment task. In this experiment, we use JIT-LiNGAM with the neighbor selection way of ERN-Euclidean, which is the best model of proposed methods as showed in Experiment 1. The detail of this conclusion is explained later in Section 4.4. In Experiment 2, we compared the proposed method to representative causal discovery methods for non-linear and non-stationary data under some general evaluation metrics of causal discovery.

4.3. Baselines and Evaluation Metrics

In Experiment 1, we prepared two linear models as baselines: one that executes a causal discovery using all the stored data $\mathcal{D}$ observed at each time (Cumulative All), and one that executes a causal discovery using randomly selected K samples from the stored data $\mathcal{D}$ regardless of the distance (K-Random). Both of them use LiNGAM as a causal discovery algorithm. For the evaluation metric, we used Structural Hamming Distance (SHD), which has been widely utilized as a performance index in other causal discovery studies (Zheng et al. (2020); Uemura et al. (2022)). This measure does not focus on the strength of the estimated coefficients given by causal discovery (the adjacency matrix $\boldsymbol{B}$ in LiNGAM) but rather on only on the causal structure of zero or nonzero. On the basis of this idea, SHD evaluates a kind of distance from the minimum number of operations (adding/reversing/deleting edges) required to transform the estimated graph into the true graph on a DAG representing causality.

In Experiment 2, as baselines, we adopted RESIT with Random Forest regressor as a representative of non-linear causal discovery methods, and CD-NOD as a representative of non-stationary causal discovery methods in addition to Cumulative All and K-Random. For the evaluation metric, we uses some famous classification evaluation metrics, False Discovery Rate (FDR) (same as $1 - precision$), True Positive Rate (TPR) (same as recall), False Positive Rate (FPR), F1 score (harmonic mean of precision and recall) in addition to SHD. Each metric is measured for the edge predictions considering direction, i.e., when a edge with a certain direction is predicted, it is counted as one prediction positive sample, however if the true edge has opposite direction, this prediction is counted as one false positive sample. CD-NOD's output contains undirected edges. For undirected edges, we propose two ways of measure. One is "Active", which regards the undirected edges appropriately as directed edges to get possibly closer to true edges direction. The other is "Inactive",

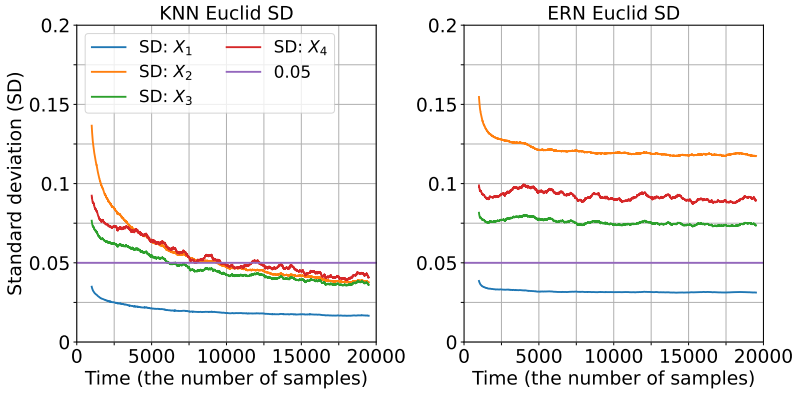

Figure 4: Time evolution of the standard deviation for each feature dimension in the neighborhood set . Because of the large fluctuation, a moving average is taken over 1000 samples. 0.05 is shown as a rough indication of the Laplace noise scale.

which regards the undirected edge as if nothing, i.e. the interpretation that neither of both directions of edges is predicted. Active is more advantageous for CD-NOD than Inactive.

4.4. Results and Discussion

The results of Experiment 1 are presented in Figure 3, where a lower SHD indicates that the estimated graph is closer to the true causal graph. Since there was considerable variation in the accuracy of causal discovery for each query point, a moving average over 1000 samples is shown. As we can see, all of the proposed methods outperformed the baselines over all the time. Thanks to its algorithm, JIT is able to collect similar neighborhoods to the query point as the amount of accumulated data increases and construct more detailed and localized models. This suggests that the error would presumably decrease with the passing of time, but in fact we found that the error tends to increase slightly over time when the KNN method is used to collect neighborhoods (Figure. 3).

One potential reason for the increasing error is the supposition that the causal discovery accuracy depends on the training dataset's variance (radius). Figure 4 plots the time evolution of sample standard deviations in the neighborhood set. The sample's density in the data space increases as the observed samples $\mathcal{D}$ increase over time. Therefore, in KNN, where the number of neighbors K is fixed and the elements are taken from the nearest neighbor in order, the size of the region covering the neighborhood set becomes smaller and smaller as Figure 4 shows. The value 0.05 in Figure 4 is shown as a rough indication of the Laplace noise scale. When the scale (standard deviation) of the neighborhood set is below this, most of the contributions to the data values in the neighborhood set become noise, which makes it difficult to detect tendencies originating from causal effects, as depicted in Figure 5. As a result, combined with the general difficulty of causal discovery, it is thought that causality becomes more difficult to find over time.

In contrast, as we can see in Figure 4, the algorithm by ERN enables causal discovery without shrinking the neighborhood radius. As a result, its accuracy improves with time, as shown in Figure 3. Normally, the reason for fixing the number of neighbors K in the JIT algorithm is to ensure the

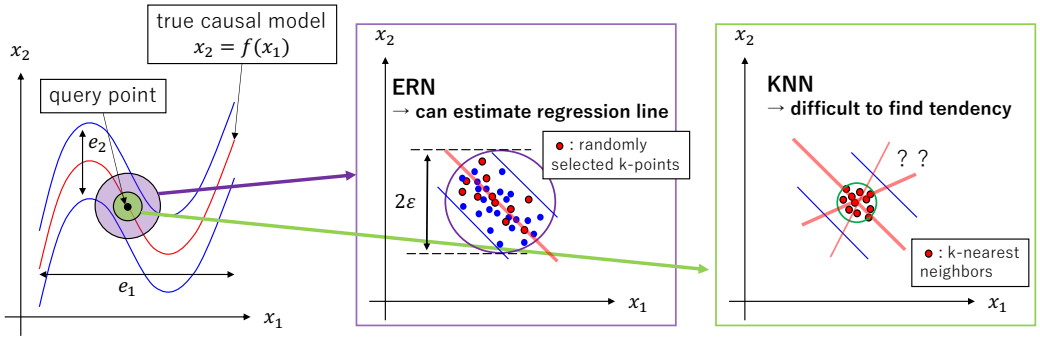

Figure 5: Illustration of the reason why the accuracy of causal discovery decreases as the radius of neighbors' set becomes smaller.

upper limit of computational complexity, and as these results demonstrate, ERN can achieve higher performance than KNN while maintaining this upper limit of computational complexity.

Table 1 and Figure 6 show the results of Experiment 2. It can be seen in them that the proposed method outperforms the conventional methods in most of the measures and in a wide range of time. Seeing Table 1, CD-NOD Active, which is the evaluation for CD-NOD regarding the undirected edge appropriately as a directed edge to get closer to true edge direction, is seeming better than proposed method. However, this evaluation is too much advantageous for CD-NOD, and CD-NOD Inactive indicates extremely low performance in contrast. The proposed method can output fully directed causal graph, so this method is thought sufficiently superior to CD-NOD. (However, if you concern, taking the mean of "Active" and "Inactive" for example in SHD, you can get perhaps an effective evaluation metrics where the operation of removing/directing an undirected edge is treated as costing 1/2). Seeing Figure 6, there are some time sections where the evaluation metrics of the proposed method is slightly worse. JIT-LiNGAM captures the causal relationships for only the latest state of system, the result differs depending on query data $\boldsymbol{x}_{\mathrm{Q}} = \boldsymbol{x}^{(T)}$. Thus, taking the time average of the evaluation values, like in Table 1, is somewhat reasonable, since it means also taking the approximated expected value under the distribution of the (query) data.

5. Conclusion and Future Work

We proposed JIT-LiNGAM as a method to obtain local linear approximated causal models in the neighborhood of the query point for non-linear and non-stationary data. The proposed method combines the JIT framework, which has conventionally been used for non-stationary non-linear time-series regression problems, system identification, or control engineering, and LiNGAM, which is one of the most popular linear causal discovery algorithms. The theory and computational procedure of JIT-LiNGAM was formulated and the scope of its application was clarified. Thanks to its linear approximation, JIT-LiNGAM enables easier interpretation of the originally non-linear causality in the terms of its positive/negative effect, its strength, and its change over time. Moreover, its overall computational complexity can be reduced to the complexity of the JIT framework's Neighbor-Search part, and the computational complexity of the causal discovery part can be limited to LiNGAM's one. Due to these characteristics, JIT-LiNGAM is expected to be more scalable to

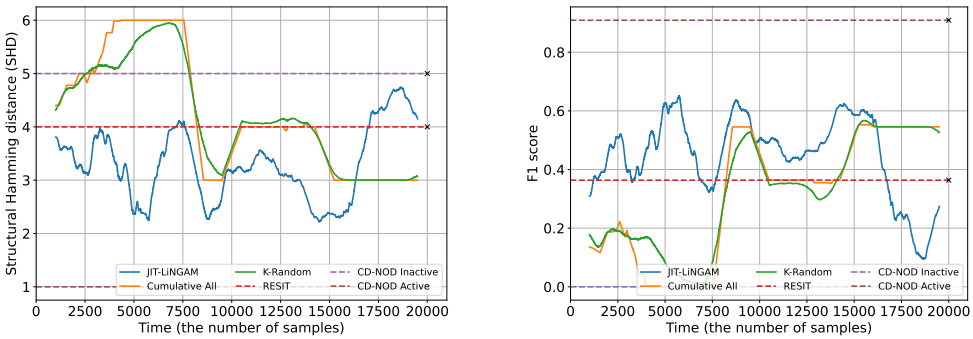

Figure 6: The results of Experiment 2 plotting time evolution of SHD and F1 score for each method. Note that, while Just-In-Time like methods (JIT-LiNGAM, Cumulative All, K-Random) makes estimation every time the new data is observed, evaluation for RESIT and CD-NOD is executed once the all data is finally given.

Table 1: The results of Experiment 2. In JIT-LiNGAM, Cumulative All, K-Random, the time sample means for each metric is taken. Result of *RESIT and *CD-NODs are One-Shot, i.e. the one-estimation after all the observations are given at end of the time.(↓) means lower is better. (↑) means greater is better.

	FDR (↓)	TPR (↑)	FPR (↓)	SHD (↓)	F1 score(↑)
JIT-LiNGAM	0.558	0.452	0.408	3.317	0.443
Cumulative LiNGAM	0.708	0.345	0.588	4.211	0.316
K-Random LiNGAM	0.697	0.351	0.566	4.168	0.325
* RESIT	0.667	0.400	0.571	4.000	0.364
* CD-NOD Inactive	0.000	0.000	0.000	5.000	0.000
* CD-NOD Active	0.167	1.000	0.143	1.000	0.909

the increase of the number of variables P than RESIT. As a future work, we will examine this by experiments for multivariate data with large dimension. Finally, to demonstrate the effectiveness of the proposed method, we conducted experiments on artificial data and compared the accuracies of several neighborhood selection methods, including the baseline methods. As a result, we confirmed the superiority of the proposed method and obtained some suggestions on how to select useful neighborhoods especially for causal discovery, in JIT's neighborhood selection part.

As future works, we consider some issues. Based on its algorithm and formulation, JIT-LiNGAM may be able to handle cases where (not only changing its strength, but also) the structure of the causal graph itself is time-varying, but this also needs to be further tested. It has been already also mentioned in Section 2.1, we can think the extension of JIT-LiNGAM with time delay embedding model, which may be more suitable for the time-series analysis. It also remains as future work. And, whether or not the output linear causal effect becomes the local approximation of original non-linear ANM as its Jacobian should be examined later studies.

References

Gianluca Bontempi, Mauro Birattari, and Hugues Bersini. Lazy learning for local modelling and control design. *International Journal of Control*, 72(7-8):643–658, 1999.

Martin W Braun, Daniel E Rivera, and Anders Stenman. A'model-on-demand'identification methodology for non-linear process systems. *International Journal of Control*, 74(18):1708–1717, 2001.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21:689–696, 2008.

Biwei Huang, Kun Zhang, Jiji Zhang, Joseph D Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. Causal discovery from heterogeneous/nonstationary data. *J. Mach. Learn. Res.*, 21(89):1–53, 2020.

Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11 (5), 2010.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal Inference in Statistics: A Primer*. John Wiley & Sons, 2016.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15:2009–2053, 2014.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O Hoyer, and Kenneth Bollen. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *The Journal of Machine Learning Research*, 12:1225–1248, 2011.

Anders Stenman, Fredrik Gustafsson, and Lennart Ljung. Just in time models for dynamical systems. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 1, pages 1115–1120. IEEE, 1996.

Kento Uemura, Takuya Takagi, Kambayashi Takayuki, Hiroyuki Yoshida, and Shohei Shimizu. A multivariate causal discovery based on post-nonlinear model. In *Conference on Causal Learning and Reasoning*, pages 826–839. PMLR, 2022.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. PMLR, 2020.

Appendix A. The Algorithm Conventional Just-In-Time Model for Regression

Algorithm 2 JIT Algorithm for Time-series Regression

Inputs:
stored data $\mathcal{D} = \left\{(\boldsymbol{x}^{(t)}, y^{(t)}) \mid t = 1, \ldots, T-1\right\}$,
query point $\boldsymbol{x}_{\mathrm{Q}} = \boldsymbol{x}^{(T)}$, distance function $d(\cdot, \cdot)$, number of neighbors K.
Outputs:
prediction $\hat{y}^{(T)}$ (i.e. estimation of output $y^{(T)} = f(\boldsymbol{x}^{(T)})$ for input $\boldsymbol{x}^{(T)}$).

Procedure 1
Extract K-pairs of $(\boldsymbol{x}^{(t)}, y^{(t)})$ from $\mathcal{D}$, in ascending order of $d(\boldsymbol{x}^{(t)}, \boldsymbol{x}_{\mathrm{Q}})$, which is the distance from the query point $\boldsymbol{x}_{\mathrm{Q}}$. The resulting K-pair subset $\Omega(\boldsymbol{x}_{\mathrm{Q}}; d, K)$ is:

$$\Omega(\boldsymbol{x}_{\mathrm{Q}}; d, K) = \left\{(\boldsymbol{x}^{(\sigma(k))}, y^{(\sigma(k))}) \mid k = 1, \ldots, K\right\},$$

where $\sigma(k)$ is a function that returns the k-th nearest time index t.

Procedure 2
Learn the local linear regression model $\hat{f}_{\boldsymbol{x}_{\mathrm{Q}}}(\cdot)$ using $\Omega(\boldsymbol{x}_{\mathrm{Q}}; d, K)$.

Procedure 3
Get the prediction of $\hat{y}^{(T)} = \hat{f}_{\boldsymbol{x}_{\mathrm{Q}}}(\boldsymbol{x}_{\mathrm{Q}})$.

Appendix B. Calculation Time

Table 2: The calculation time of each methods in Experiment 2. "Total" means the time to take making the all of N-estimations for $T = 1, 2, \ldots, N$. "One-Shot" means the time to take making the one-estimation after all the observations are given at end of the time. Note that the two part's result is shown about JIT-LiNGAM since in the program code we use for experiment, we can separately calculate the Neighbor Search part and the Causal Discovery part.

	calculation time [second]
JIT-LiNGAM (Neighbor Search)	658.493
JIT-LiNGAM (Causal Discvory)	300.777
JIT-LiNGAM (Total)	959.270
Cumulative LiNGAM (Total)	1434.036
K-Random LiNGAM (Total)	301.798
RESIT (One-shot)	1175.096
CD-NOD (One-Shot)	0.068