다변량 시계열 데이터에 대한 LPCMCI의 인과적 추론과 성능 분석

Causal Inference with LPCMCI for Multivariate Time-series and Performance Analysis

박성훈

August 19, 2022

Abstract

복잡하고 거대한 시스템에 산재하는 다수의 변수들 사이에서 인과관계를 파악하고, 인과적 추론을 하고자 하는 시도는 이전부터 계속해서 이루어져 왔다. 앞선「다변량 시계열 데이터에 대한 인과적 추론(Causal Inference with Multivariate Time-series)」문서에서는 네 개의 인과 추론 방식 DoWhy, TCDF, LPCMCI, PCMCI$^+$에 대해서 비교 및 분석하였으며, 그 중 LPCMCI[1]가 네트워크 분야로의 적용에 있어 가장 적절할 것으로 판단하였다. 그러나, LPCMCI는 성능 분석에 있어 정상성(stationarity)을 만족하는 이산적 벡터 자기 회귀 과정(discrete-time structural vector-autoregressive process)을 따르는 임의로 생성된 다변량 시계열 데이터를 사용하였으며, 데이터들의 특징적인 가정(assumption)들이 성립되는지 확인되지 않은 실제 데이터에 대해서는 성능 분석이 이루어지지 않았다. 따라서, 본 문서에서는 LPCMCI에 대해 보다 자세히 조명하고, 실제 다변량 시계열 데이터에 적용하여 그 성능을 분석하고자 한다.

1 서론

복잡한 시스템에서 발생하는 다양한 현상들의 원인을 규명하는 일은 많은 분야에서 시도되어 왔으며, 네트워크 분야 역시 그러한 분야 중 하나이다. LPCMCI는 시스템이 지나치게 복잡하여 분석할 수 없는(infeasible) 경우나, 실시간으로 동작하는 시스템에서 반복적인 실험을 충분한 변인 통제 하에서 행할 수 없는 경우에 사용될 수 있는 데이터 기반의 인과관계 파악 알고리즘이다. LPCMCI는 관측된 데이터들의 특징적인 가정(assumption)들을 설정하여 이를 바탕으로 인과관계를 파악하는 방식을 채택하고 있다. 특히, 대상이 되는 변수들을 인과관계 그래프(causal graph)로 모델링한 후 인과관계를 분석하고자 하는 구조적 인과 모델(structural causal model)과 CI(Conditional Independence)를 기반으로 인과적 구조(causal structure)를 학습하고자 하는 시도가 현재 데이터 기반 인과관계 파악의 주된 방향이며, LPCMCI 역시 이러한 방향을 그대로 따르고 있다. LPCMCI는 PC 알고리즘[2]과 FCI(Fast Causal Inference) 알고리즘[3]의 주요한 아이디어를 기반으로 하고 있으며, 관측되지 않은 숨겨진 변수(hidden variable)가 존재하는 상황에서 인과관계를 파악하고, FCI 계열의 알고리즘들이 겪고있는 저조한 효과 크기(effect size)와 재현율(recall)의 문제를 개선하려는 목적으로 제시되었다.

LPCMCI의 성능에 대한 분석에서는 LPCMCI가 강한 자기 상관(autocorrelation) 아래에서도 안정적인 FPR 수준을 유지하며, 방향 결정 재현율(orientation recall)과 효과 크기를 개선하였음을 확인하였다. 그러나, LPCMCI의 성능 측정은 이하의 식으로 나타나는 이산적 벡터 자기 회귀 과정을 따르는 임의로 생성된 데이터에 대해서 이루어졌으며, 실제 데이터에 대한 적용은 확인되지 않는다.

$$V_t^j = a_j V_{t-1}^j + \sum_i c_i f_i(V_{t-\tau_i}^i) + \eta_t^j \quad \text{with } j = 1, \cdots, \tilde{N} \tag{1}$$

따라서, 본 문서에서는 *faithfulness*와 정상성에 대해 가정하기 어려운 실제 다변량 시계열 데이터에 대해서 LPCMCI를 적용하고 그 성능을 분석하고자 한다. 또한, LPCMCI의 성능 측정은 변수의 개수가 15개 이하인 환경에서 이루어졌는데 변수의 개수가 증가함에 따라 실행 시간이 급격하게 증가하는 양상을 보였다. 그러므로 성능 분석에 사용되는 데이터셋은 적어도 20개 이상의 변수로 구성되게 하여 다변량 환경에서의 LPCMCI의 수렴(convergence)과 실행 시간 관점에서의 실행 가능성(feasibility)에 대해서 확인하고자 한다.

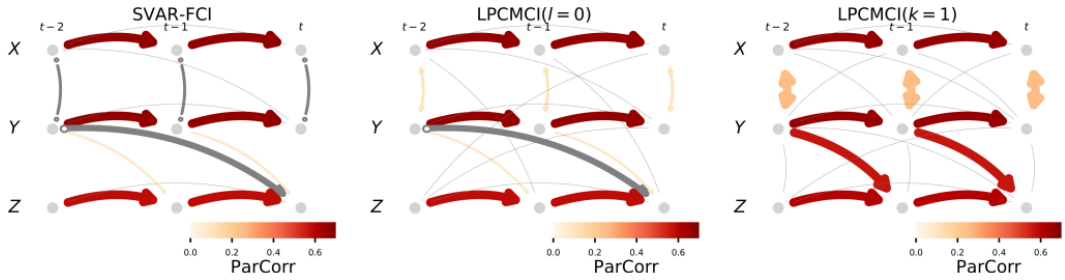

Figure 1: Latent confounder example

2 이론적 배경

2.1 LPCMCI(Latent PC Momentary Conditional Independence)

시계열의 인과관계 파악은 숨겨진 교란변수나 고차원성(high-dimensionality), 그리고 비선형 의존성(nonlinear dependencies)과 같은 문제에 직면해왔다. 특히, 현존하는 인과관계 파악 방식들은 자기상관(autocorrelation)이 산재하고 교란변수(confounder)가 존재하는 상황에서 인과관계를 파악하는 데에 어려움을 겪고 있다. LPCMCI의 핵심적인 기여는 낮은 효과 크기(effect size)를 식별하여 기존의 인과관계 파악 방식의 문제점을 효과적으로 개선함에 있다. LPCMCI에서 다루는 다변량 시계열 데이터 $\mathbf{V}^j = (V_t^j, V_{t-1}^j, \cdots)$ for $j = 1, \cdots, \tilde{N}$는 식 (7)에서 나타낸 바와 같은 구조적 인과 모델(structural causal model)로 기술되는 정상성(stationary)을 가지는 이산적 벡터 자기회귀 과정(discrete-time structural vector-autoregressive process)을 따른다.

$$V_t^j = f_j(pa(V_t^j), \eta_t^j) \quad \text{with } j = 1, \cdots, \tilde{N} \tag{2}$$

이때, 함수 f_j는 입력으로 주어지는 모든 인자에 비선형적으로(non-trivially) 의존하며, 잡음 변수(noise variable) η_t^j는 서로 독립적이고, 집합 $pa(V_t^j) \subseteq (\mathbf{V}_t, \mathbf{V}_{t-1}, \cdots, \mathbf{V}_{t-p_{t_s}})$는 V_t^j의 인과적인 부모(causal parent)에 해당한다. 시계열 데이터에는 순환적 인과관계(cyclic causal relationship)이 존재하지 않는 것으로 가정하였으며, 관측되지 않은 변수가 존재하는 것을 허용하였다. 또한, *faithfulness* [3]을 가정하였는데, 이는 구조적 인과 모델에 의해서 생성된 관측 분포 $P(\mathbf{V})$에서 조건부 독립(CI, conditional independence)이 d-seperation으로 구축된 DAG $\mathcal{G}$와 동일하다는 것을 의미한다.

2.2 기존 인과 추론 방식의 문제

Figure 1에서는 관측되지 않은 변수들에 의해 기존의 인과적 추론 방식이 보이는 문제를 나타내고 있다. 부분적 상관관계(ParCorr, partial correlation) CI를 사용한 SVAR-FCI는 각 변수의 자기 연결(auto-link)은 잘 파악했지만, 실제 연결 $Y_{t-1} \to Z_t$을 파악하지 못하고 잘못된 연결 $Y_{t-2} \to Z_t$(회색 화살표)를 반환하였다. 또한, 많은 경우에서 동시 인접성(contemporaneous adjacency) $X_t \leftrightarrow Y_t$을 발견하지 못했으며, 발견되었더라도 양방향 연결이라는 것을 파악하지 못했다. 이러한 문제는 간선 제거(edge removal)와 방향 결정(orientation) 단계에서의 잘못된 CI 테스트에서 기인한다. 간선 제거 단계에서, X와 Y의 높은 자기상관은 분산을 증가시키고 SN비(signal-to-noise ratio)를 감소시키면서 상관성 $\rho(X_t; Y_t)$를 무시하게 만든다. 또한, 방향 결정 단계에서도 잘못된 CI 테스트로 인한 문제가 발생한다. 원칙적으로는 SVAR-FCI의 $\mathcal{R}_0$에 의하여 $X_t \leftrightarrow Y_t$가 식별되어야 하나 실제로는 순서 의존성(order-dependence)를 회피하기 위해 Y_{t-1}와 X_t에 대한 추가적인 CI 테스트가 수행되며 자기상관의 영향으로 X_t이 독립인 것으로 판별된다. 이에 따라 $X_t \leftrightarrow Y_t$ 관계는 인정되지 않는다. 이 연구에서는 자기상관이 낮은 효과 크기를 발생시켜 잘못된 CI 테스트로 이끌고, 이에 따라 인과관계 파악 오류가 발생한다고 보고 이를 해결하기 위한 LPCMCI를 제안하였다.

2.3 인과관계 파악에서의 효과 크기

실제 연결 $X_{t-\tau}^i * \to X_t^j$의 발견력(detection power)은 해당 연결이 잘못된 CI 테스트로 인해서 제거되지 않을 가능성에 의존한다. 이는 *i*) 표본 크기(sample size, 일반적으로 고정됨), *ii*) CI 테스트의 유의수준 α (significance level, 연구자가 원하는 긍정오류 수준으로 고정), *iii*) CI 테스트의 추정 차원(estimation dimension), 그리고 *iv*) 효과 크기에 의존한다. 이때, 효과 크기를 테스트되는 모든 조건 집합(conditioning set) $\mathcal{S}$에 대해서 취해진 CI 테스트의 통계값 $I(A; B|\mathcal{S})$의 최소값으로 정의한다. LPCMCI의 핵심적인

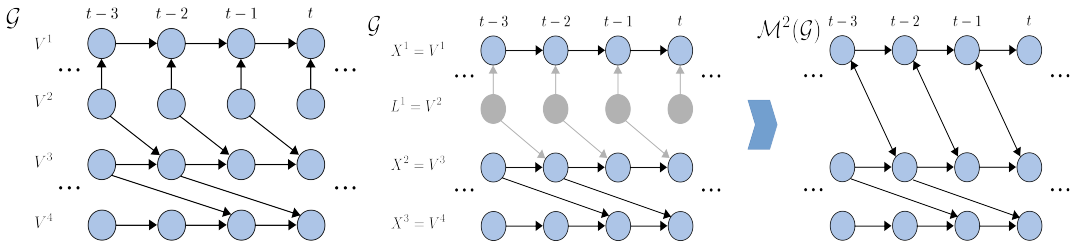

Figure 2: Example of Directed Acyclic Graph (Left) and Directed Maximal Ancestral Graph (Right)

아이디어는 a) 조건 집합 $\mathcal{S}$를 제한하고, b) 디폴트 조건(default condition) $\mathcal{S}_{def}$와 함께 테스트되는 $\mathcal{S}$를 확장함으로써 효과 크기를 증가시키는 것이다. LPCMCI는 $\mathcal{S}_{def}$를 $X^i_{t-\tau}$와 X^j_t의 부모들의 합집합으로 취함으로써 효과 크기를 개선하였으며, 이는 PCMCI의 근간이 되는 MCI(Momentary Conditional Independence)를 일반화한다.

이에 따른 LPCMCI의 핵심적인 디자인은 크게 두 가지로 설명할 수 있다. 첫째, $X^i_{t-\tau}$와 X^j_t의 조건부 독립을 테스트함에 있어 $X^i_{t-\tau}$와 X^j_t의 조상이 아닌 것(non-ancestor)을 포함하는 조건 집합들을 버린다. 둘째, $X^i_{t-\tau}$와 X^j_t의 알려진 부모를 디폴트 조건으로 사용한다. 이렇게 얻어진 높은 효과 크기는 추정 차원의 증가에 의해서 약화되지 않는다면 더 높은 발견력과 실제 연결의 기억을 이끌어 낼 수 있다. 이러한 디자인을 위해서는 CI 테스트가 완료되기 전에 조상관계(ancestorship)의 일부를 알고 있어야 한다. LPCMCI는 간선 제거 단계와 방향 결정 단계를 결부시켜 모든 잘못된 연결이 제거되기 전에 조상관계를 학습하게 함으로써 이를 달성하였다.

2.4 DAG, DMAG, DPAG

구조적 인과 과정(structural causal process)에 의한 인과관계는 DAG(directed acyclic graph) $\mathcal{G}$로 나타낼 수 있다. DAG는 정점과 간선으로 구성되며, 정점은 변수 V^j_t를, 간선은 $V^j_{t-\tau} \in \mathcal{P}(V^j_t)$인 경우에 해당하는 $V^j_{t-\tau} \to V^j_t$를 의미한다. Figure 2의 왼쪽 그림에서는 식 (3)-(6)에서 나타내고 있는 네 변수의 DAG를 도시하고 있다. DAG는 기본적으로 사이클(cycle)을 가지지 않으며, 정상성 가정에 의해 시간에 따른 반복적인 구조가 나타난다.

$$V^1_t = 0.9V^1_{t-1} + 0.6V^2_t + \eta^1_t \tag{3}$$
$$V^2_t = \eta^2_t \tag{4}$$
$$V^3_t = 0.9V^3_{t-1} + 0.4V^2_{t-1} + \eta^3_t \tag{5}$$
$$V^4_t = 0.9V^4_{t-1} - 0.4V^3_{t-2} + \eta^4_t \tag{6}$$

LPCMCI와 PCMCI 및 PCMCI$^+$의 주요한 차이점은 LPCMCI가 데이터에서 일부 변수가 관측되지 않는 상태를 허용한다는 점이다. LPCMCI는 관측된 변수들만으로 관측되지 않은 잠재 변수들의 인과관계를 나타내기 위하여 DMAG(directed maximal ancestral graph)를 이용한다. DMAG는 일반적인 간선 $X \to Y$ 이외에도 $X \leftrightarrow Y$로 표현되는 양방향 간선을 갖는다. LPCMCI에서는 사이클을 허용하지 않으므로 $X \to Y$이 존재하면 $Y \to X$는 존재할 수 없다. 따라서, DMAG에서 $X \leftrightarrow Y$가 의미하는 바는 X와 Y 사이에 인과관계가 존재하지 않으나 관측되지 않은 동일한 교란 변수에 의해 영향을 받고 있다는 의미이다. Figure 2의 오른쪽 그림에서는 변수 V^2가 관측되지 않는 상황에서의 DMAG를 나타내고 있다. V^1_{t-1}과 V^3_t 사이에는 어떠한 인과관계도 존재하지 않으나 관측되지 않은 변수 V^2에 의하여 영향을 받고 있으므로, 양방향 간선 $V^1_{t-1} \leftrightarrow V^3_t$으로 두 변수 사이의 관계가 표현된다.

LPCMCI는 데이터의 조건부 독립을 이용하여 제약조건(constraint) 기반의 접근을 취함으로써 관측된 시계열로부터 DMAG를 학습한다. 그러나, DMAG만으로는 결정적인 학습이 불가한데(underdetermined) 이는 서로 다른 DMAG가 정확히 같은 형태의 독립성을 나타낼 수 있기 때문이다. 이것을 마르코프 등가(Markov equivalence)라고 부른다. 이러한 이유로 LPCMCI는 DMAG $\mathcal{M}(\mathcal{G})$의 마르코프 등가 클래스(Markov equivalence class)들이 공유하는 특성(feature)들만 학습할 수 있으며 이러한 공유되는 특성들을 DPAG(directed partial ancestral graph)로 나타낸다. DPAG에는 DMAG가 가지는 두 가지 형태의 간선 이외에도 $X \circ\!\!\to Y$라는 새로운 형태의 간선이 존재한다. 이때, $X \circ\!\!\to Y$가 의미하는 바는 X가 Y의 원인이 될 수도 그렇지 않을 수도 있지만(X may or may not cause Y), Y는 X의 원인이 될

수 없다는 것이다. DPAG $\mathcal{P}(\mathcal{G})$는 LPCMCI 알고리즘의 실행 동안 데이터의 인과적 조상 관계(causal ancestral relationship)를 표현한다.

2.5 중간 표시(middle mark)의 도입과 LPCMCI-PAG의 방향 결정

간선들의 초기 방향 결정을 용이하게 하기 위해서 그래프에 분명한 인과적 해석을 넣었다. 이는 중간 표시(*middle mark*)로 간선들을 늘리는 것으로 수행되었다. 중간 표시는 '?', 'L', 'R', '!', ''(empty)으로 총 5개가 존재하며 각각이 특별한 의미를 갖는다. 이러한 중간 표시를 통해 확장된 PAG(partial ancestral graph) $\mathcal{C}(\mathcal{G})$를 $\mathcal{M}(\mathcal{G})$에 대한 LPCMCI-PAG라고 부른다. LPCMCI-PAG은 $A*-*B$에 $A \in adj(B, \mathcal{M}(\mathcal{G}))$를 할당하여 모호함을 없애고, 인접성에 대한 인과적 정보를 명시적으로 전달할 수 있다. $\mathcal{C}(\mathcal{G})$에서 $A \overset{!}{\to} B$, $A \overset{\mathrm{L}}{\to} B \; for \; A > B$, 그리고 $A \overset{\mathrm{R}}{\to} B \; for \; A < B$는 $A \to B$로 대체될 수 있다. LPCMCI가 완료되면 모든 중간 표시는 공백(empty)이 되고 $\mathcal{C}(\mathcal{G})$는 PAG가 된다.

방향 결정 단계에 대해서 설명하기에 앞서 Weakly minimal separating set에 대해서 정의한다.

Definition 1. (Weakly minimal separating set) MAG $\mathcal{M}(\mathcal{G})$에서 A와 B가 $\mathcal{S}$에 의해서 m-separate 된다고 가정한다. 만약 $\mathcal{S}$가 $\mathcal{S} = \mathcal{S}_1 \dot{\cup} \mathcal{S}_2$으로 분해되고, $\mathcal{S}_1 \subseteq an(A, B, \mathcal{M}(\mathcal{G}))$가 만약 $\mathcal{S}' = \mathcal{S}_1 \dot{\cup} \mathcal{S}_2'$가 A와 B를 m-separate하면 $\mathcal{S}_2 = \mathcal{S}_2'$인 관계가 성립하면 이때 집합 $\mathcal{S}$를 weakly minimal separating set이라고 정의한다.

이러한 weakly minimal separating set은 $\mathcal{S}_1 = \emptyset$일때만 성립하는 minimal separating set의 개념을 확장한 것이다. LPCMCI는 알려진 조상들을 이용하여 조건 집합을 확장하도록 디자인되었기 때문에 찾아낸 separating set은 보통의 경우 최소(minimal)는 아니나 약한 최소(weakly minimal)는 만족한다. LPCMCI는 이렇게 얻어진 weakly minimal separating set을 다음의 Lemma 1 (Strong unshielded triple rule)에 따라 방향 결정을 수행한다.

Lemma 1. (Strong unshielded triple rule) LPCMCI-PAG $\mathcal{C}(\mathcal{G})$에 존재하는 관계 $A* \overset{*}{-} *B* \overset{*}{-} *C$를 unshielded triple라고 하고 $\mathcal{S}_{AC}$를 A와 C의 separating set이라고 가정한다. *1)* 만약 $B \in \mathcal{S}_{AC}$이고 $\mathcal{S}_{AC}$가 약한 최소(weakly minimal)이면 $B \in an(\{A, C\}, \mathcal{G})$이다. 또한, 임의의 $T_{AB} \subseteq an(A, B, \mathcal{M}(\mathcal{G}))$와 $T_{CB} \subseteq an(C, B, \mathcal{M}(\mathcal{G}))$에 대하여 *2)* 만약 $B \notin \mathcal{S}_{AC}$이고, A와 B가 $\mathcal{S}_{AC} \cup T_{AB} \backslash \{A, B\}$로 m-separate되지 않으며, C와 B가 $\mathcal{S}_{AC} \cup T_{CB} \backslash \{C, B\}$로 m-separate되지 않으면 $B \notin an(\{A, C\}, \mathcal{G})$이다.

위 Lemma 1에 따라 LPCMCI의 바탕이 되는 FCI에서 사용하는 일련의 규칙들을 일반화할 수 있고, 알고리즘 진행 중 어느 부분에서든지 규칙이 적용될 수 있게 된다. 이는 기존의 FCI에서 PAG가 발견된 이후에 방향 결정이 진행될 수 있었던 것과는 다르게 잘못된 연결이 제거되기 전에 조상 관계를 학습할 수 있게 하여 효과 크기를 증대시킬 수 있다.

2.6 LPCMCI 알고리즘

Algorithm 1에서는 LPCMCI의 알고리즘을 나타내고 있다. $\mathcal{C}(\mathcal{G})$를 완전 그래프(complete graph)로 초기화한 후에 LPCMCI은 line 2-4에 해당하는 예비 단계(preliminary phase)를 수행한다. 이 단계에서 많은 잘못된 연결들이 제거되며, 방향 결정 규칙들이 반복적으로 적용된다. 이 규칙들은 $\mathcal{G}$의 조상관계를 파악하고 $\mathcal{C}(\mathcal{G})$의 간선들에 이를 표시한다. LPCMCI는 조상관계가 아닌 것들에 대해서는 조건 집합 $\mathcal{S}$를 제한하며, 조상관계인 것들에 대해서는 $\mathcal{S} \cup \mathcal{S}_{def}$로 확장한다.

Algorithm 1 LPCMCI

Require: Time series dataset $\mathbf{X} = \{\mathbf{X}^1, \ldots, \mathbf{X}^N\}$, maximal considered time lag $\tau_{\max}$, significance level α, CI test $\mathrm{CI}(X, Y, \mathcal{S})$, non-negative integer k

1: Initialize $\mathcal{C}(\mathcal{G})$ as complete graph with $X^i_{t-\tau} \circ\!\!\overset{L}{\to} X^j_t$ $(0 < \tau \leq \tau_{\max})$ and $X^i_{t-\tau} \circ\!\!\overset{?}{-}\!\!\circ X^j_t$ $(\tau = 0)$
2: **for** $0 \leq l \leq k-1$ **do**
3: Remove edges and apply orientations using Algorithm S2
4: Repeat line 1, orient edges as $X^i_{t-\tau} \overset{?}{\to} X^j_t$ if $X^i_{t-\tau} \overset{*}{\to} X^j_t$ was in $\mathcal{C}(\mathcal{G})$ after line 3
5: Remove edges and apply orientations using Algorithm S2
6: Remove edges and apply orientations using Algorithm S3
7: **return** PAG $\mathcal{C}(\mathcal{G}) = \mathcal{P}(\mathcal{G}) = \mathcal{P}(\mathcal{G})^{\tau_{\max}}_{statAO}$

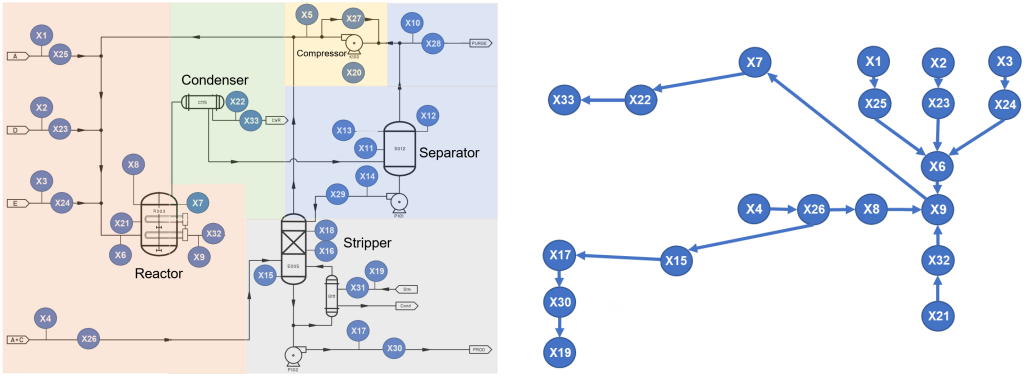

Figure 3: (a) Structure of the TE production plant with corresponding variables selected for causal discovery, (b) Ground truth causal graph for TE dataset

이렇게 Algorithm S2 (line 3)과 $\mathcal{C}(\mathcal{G})$를 다시 초기화(line 4)하는 것을 반복하는 것은 $\mathcal{G}$에 있는 부모관계의 부분집합을 정확하게 파악하기 위함이다. 그 이후에는 line 5-6에 해당하는 마지막 단계로 넘어간다. 이 단계에서는 이전 단계에서 찾아내지 못한 잘못된 간선을 지우고 방향 결정을 적용해 최종적으로 PAG $\mathcal{P}(\mathcal{G})$를 찾아내는 단계이다. 예비 단계의 반복 횟수 k는 하이퍼 파라메터(hyperparameter)이며, 알고리즘의 매 단계마다 만족되도록 한다.

3 성능 측정 결과 및 분석

3.1 실행 환경

3.1.1 데이터셋

LPCMCI의 성능 측정을 위한 데이터셋으로는 테네시-이스트만 공정(Tennessee- Eastman Process)의 시계열 데이터를 이용하였다.[4] 테네시-이스트만 공정은 화학 공학 연구에서 성능 측정을 위해 사용되는 데이터셋이며 Figure 3 (a)에서 도시하고 있는 것처럼 리엑터(reactor), 응축기(condenser), 압축기(compressor), 액기분리기(liquid-vapor separator), 스트리퍼(stripper)의 다섯 가지 설비에 의한 공정으로 구성된다. 테네시-이스트만 공정은 41개의 측정 가능 변수와 12개의 통제 변수를 갖는다. 데이터셋 구성을 위해 측정된 변수들은 압력, 온도, 농도 등과 같이 시간에 따른 연속적 표본 추출이 가능한 것들로 선정되었으며, $0.1Hz$로 추출되었다. 원본 테네시-이스트만 공정 데이터셋은 길이 1500의 33개의 변수로 구성되나, LPCMCI의 성능 측정은 변수를 20개로 축소한 데이터셋으로 이루어졌다. 그 이유는 25개의 다변량 환경에서 LPCMCI가 12시간 이상 수렴하지 않는 현상을 보였기 때문이다. 이러한 실행 시간의 급격한 증가는 단순한 변수의 증가에 따른 것이 아닌 변수 사이의 모종의 순환 관계가 형성됨에 따라 방향 결정 단계에서 수렴하는 그래프를 찾을 수 없는 교착상태에 빠지면서 발생하는 현상으로 보인다. 이러한 경우에 최대 지연 시간 τ_{max}와 신뢰 수준 α_{PC}를 적절히 조정하거나, 교착상태가 해결되고 LPCMCI가 수렴할 때까지 실행하는 방법도 있으나 본 문서에서는 변수를 적절히 축소하여 동작시키고 그 결과를 [4]에서 측정한 다양한 인과관계 파악 알고리즘들의 SOTA(State-of-the-art)와 비교하고자 한다. Figure 3 (b)에서는 LPCMCI의 성능 분석에 사용된 테네시-이스트만 공정의 실제(ground truth) 인과관계 그래프를 나타내고 있다.

3.1.2 성능 척도

Table 1에서는 LPCMCI의 성능 분석을 위해 사용된 성능 척도를 나타내고 있다. 이때, FDR, PR, RE, F1은 이하의 식을 따른다.

$$FDR = \frac{R + FP}{TP + FP} \qquad PR = \frac{TP}{TP + FP} \tag{7}$$

$$RE = \frac{TP}{TP + FN} \qquad F1 = \frac{2 * RE * PR}{RE + PR}$$

Name	Acr	Description
True Positive	TP	Detected with correct direction
False Positive	FP	Estimated but not present in true graph
False Negative	FN	Missed true causation in estimated graph
Reverse	R	Reversed direction in the estimated graph
False Discovery Rate	FDR	Rate for false discovered
Precision	PR	Ratio of correct links with respect to FP
Recall	RE	Ratio of correct links discovered
F1-score	F1	Measure of test's accuracy

Table 1: Description of metrics

Method	FDR	PR	RE	F1
Ground Truth	0	1	1	1
SOTA	**0.75 (PC)**	0.25 (PC)	0.1667 (FCI)	0.1758 (PC)
LPCMCI (Direct)	1.1052	0.1904	0.2105	0.2
LPCMCI (Indirect)	0.875	**0.4286**	**0.375**	**0.4**

Table 2: Results of performance measurement

3.2 결과 및 분석

Table 2에서는 FDR, PR, RE, F1 네 개의 성능 척도에 대한 이상치(ground truth), SOTA, 그리고 LPCMCI의 결과를 나타내고 있다. 이때, 이상치를 제외한 가장 높은 값은 굵게 표시하였다. 또한, LPCMCI의 경우 LPCMCI(Direct)와 LPCMCI(Indirect)로 두 가지 결과를 표시하였는데 LPCMCI(Direct)는 간접 인과(indirect cause)를 FP로 LPCMCI(Indirect)는 간접 인과를 TP로 처리한 결과이다. 즉, 실제 인과관계가 $X \to Y \to Z$인 경우에 LPCMCI가 $X \to Z$ 간선을 예측하였다면, LPCMCI(Direct)에서는 이를 FP로, LPCMCI(Indirect)에서는 TP로 처리한다.

먼저 정확도 PR의 경우 LPCMCI(Direct)는 SOTA인 PC 알고리즘보다는 다소 낮은 수치를 보였지만, 간접 인과를 인정한 LPCMCI(Indirect)의 경우에는 SOTA보다 상당히 높은 수치를 보였다. 재현율 RE의 경우에는 LPCMCI(Direct)가 SOTA인 FCI를 압도하는 모습을 보이고 있으며 LPCMCI(Indirect)는 훨씬 높은 수치를 보이고 있다. 이러한 결과는 LPCMCI 연구에서 주장하는 대로 LPCMCI가 효과 크기를 개선함으로써 재현율을 효과적으로 증대시켰다는 사실을 뒷바침한다. F1-score의 경우 LPCMCI(Direct)가 이미 SOTA인 PC를 넘어서고 있는데 이는 LPCMCI(Direct)가 PC보다는 다소 낮은 정확도를 보이지만, PC에 비해서 상당히 높은 재현율을 보이기 때문이다. FDR의 경우에는 LPCMCI(Direct)와 LPCMCI(Indirect) 모두 SOTA인 PC보다 높은 수치를 보이고 있다. 이는 LPCMCI가 두 변수 사이에 인과관계가 존재한다는 사실은 파악하였지만 방향을 올바르게 결정하지 못한 경우가 많다는 것을 의미한다. LPCMCI(Direct)의 경우 FDR이 특히 높은데 이 때문에 많은 변수들에 대해 인과관계의 존재를 파악하였음에도 다소 낮은 정확도를 보이고 있다.

또한, LPCMCI가 잠재적인 교란변수가 존재하는 상황에서 실행될 수 있는지 확인하기 위하여 $X8$과 $X15$의 공통 원인인 $X26$을 데이터셋에서 제거하고 LPCMCI를 동작시키는 추가적인 실험을 진행하였다. 그 결과 LPCMCI는 $\tau_{max} = 2, \alpha_{PC} = 0.01$ 인 설정에서 $X15 \circ\!\!\to X8$ 간선을 예측하였다. 이는 $X8$과 $X15$ 사이에 잠재적인 관계가 존재하나 $X8$은 $X15$의 원인이 아님을 의미한다. 즉, LPCMCI는 $X8$과 $X15$ 사이에 모종의 관계가 존재하고 그것이 $X8 \to X15$가 아님을 파악한 것이다. 그러나, $X8 \leftrightarrow X15$ 간선을 완전하게 예측하지는 못했고 따라서 둘 사이에 공통적인 원인 변수가 존재한다는 사실을 파악하지 못했다.

정리하자면 LPCMCI는 현존하는 인과관계 파악 알고리즘들의 SOTA와 비교하였을 때, 재현율과 F1-score에서는 우수한 성능을 보이나 정확도에 있어서는 다소 낮은 성능을 보인다. 이는 LPCMCI가 보이는 두 가지 특징에 기인한다. 첫 번째로 LPCMCI는 실제 인과관계가 $X \to Y \to Z$인 경우에 $X \to Y$와 $Y \to Z$의 관계는 파악하지 못하고 $X \to Z$의 관계만을 파악하는 경우가 많다는 것이다. 따라서, 간접 인과를 TP로 인정하는 LPCMCI(Indirect)의 경우에는 FDR을 제외한 모든 척도에서 SOTA를 압도하는 모습을 보인다. 두 번째 특징은 인과관계를 역으로 예측하는 경우가 많다는 것이었다. LPCMCI(Direct)와 LPCMCI(Indirect)에서 모두 높은 FDR를 확인할 수 있는데 이는 인과관계의 존재는 파악하였지만 방향 결정 단계에서 올바르지 않은 방향 결정을 내린 경우가 많다는 사실을 의미한다. 다만, 이러한 두 가지 특징들은 최대 지연 시간 τ_{max}와 신뢰 수준 α_{PC}을 지나치게 낮게 설정한 결과일 수 있다. 본 실험에서는 $\tau_{max} = 3, \alpha_{PC} = 0.002$를 사용하였는데 테네시-이스트만 공정 데이터셋의 경우 독립성 테스트(independence test) 결과 변수 사이의 상관성이 5(time)까지 유지되는 경우가 많았기 때문에 $\tau_{max} = 3$는

완전한 인과관계를 파악하기에는 부족하다. 또한, 유의수준 α_{PC} 까지 다소 낮게 설정하면서 올바른 간선들을 기각하였을 가능성이 존재한다. 그러나, 높은 τ_{max} 와 α_{PC} 은 상당한 실행 시간 증가를 야기하며, 20개의 변수가 존재하는 경우 $\tau_{max} = 5, \alpha_{PC} = 0.01$ 이상의 설정에서는 LPCMCI가 오랜 시간 수렴하지 않는 모습을 보였다. 따라서, 다변량 환경에서 LPCMCI는 수렴성과 정확도 사이에서 트레이드 오프(trade-off) 관계가 존재할 것으로 보이며 τ_{max} 와 α_{PC} 를 적절히 설정함으로써 수렴을 보장하는 동시에 정확도를 최대한 잃지 않도록 하는 것이 중요 하다고 생각된다. 또한, LPCMCI를 숨겨진 교란변수가 존재하는 상황에서 실행시킨 결과 교란변수의 존재를 완전하게 파악하지는 못했으나 그 관계를 부분적으로 파악하는 데에는 성공하였다. 따라서, LPCMCI가 관측되지 않은 변수가 존재하는 상황에서도 실행 가능하며 부분적으로 교란변수의 파악이 가능하다는 사실을 확인할 수 있었다.

4 결론

LPCMCI는 PC 알고리즘과 FCI 알고리즘의 주요한 아이디어를 기반으로 하며 관측되지 않은 숨겨진 변수(hidden variable)가 존재하는 상황에서 인과관계를 파악하고, FCI 계열의 알고리즘들이 겪고있는 저조한 효과 크기(effect size)와 재현율(recall)의 문제를 개선하였다고 주장한다. 그러나, LPCMCI의 성능 분석은 이산적 벡터 자기 회귀 과정을 따르는 임의로 생성된 데이터에 대해서 이루어졌으며 *faithfulness*와 정상성에 강하게 의존하고 있다. 따라서, 본 문서에서는 *faithfulness*와 정상성을 완전히 보장하기 어려운 실제 다변량 시계열 데이터에 대해서 LPCMCI를 적용하고 그 성능을 분석하였다. LPCMCI의 성능 측정을 위한 데이터셋으로는 테네시-이스트만 공정(Tennessee- Eastman Process)의 시계열 데이터를 이용하였다. 테네시-이스트만 공정은 화학 공학 연구에서 성능 측정을 위해 사용되는 데이터셋이며 33개의 관측 변수를 가진다. 그러나, 25개의 다변량 환경에서 LPCMCI의 급격한 실행 시간 증가가 확인됨에 따라 LPCMCI의 성능 측정에서는 변수의 개수를 20개로 축소시켰다.

성능 측정 결과 LPCMCI는 현존하는 인과관계 파악 알고리즘들의 SOTA와 비교하였을 때, 재현율과 F1-score에서는 우수한 성능을 보이나 정확도에 있어서는 다소 낮은 성능을 보였다. 이는 LPCMCI가 보이는 두 가지 특징에 기인한다. 첫 번째는 직접 인과가 아닌 간접 인과를 파악하는 경우가 많다는 것으로 간접 인과를 TP로 인정하는 LPCMCI(Indirect)이 상당히 높은 정확도와 재현율을 보인다는 사실로부터 확인할 수 있었다. 두 번째는 인과관계를 역으로 예측하는 경우가 많다는 것으로 LPCMCI의 높은 FDR에서 이를 확인할 수 있으며 LPCMCI가 인과관계의 존재는 옳게 파악하였지만 방향 결정 단계에서 올바르지 않은 방향 결정을 내린 경우가 많다는 사실을 시사한다.

다만, 이러한 두 가지 특징들은 지나치게 낮은 최대 지연 시간 τ_{max} 와 신뢰 수준 α_{PC} 에서 기인하였을 가능성이 있다. LPCMCI의 성능 측정에서는 $\tau_{max} = 3, \alpha_{PC} = 0.002$를 사용하였는데 테네시-이스트만 공정 데이터셋의 최대 지연 시간은 5 이상이므로 $\tau_{max} = 3$는 완전한 인과관계를 파악하기에는 부족하다. 또한, 유의수준 α_{PC} 까지 다소 낮게 설정하면서 올바른 간선들을 기각하였을 가능성이 존재한다. 그러나, 높은 τ_{max} 와 α_{PC} 은 상당한 실행 시간 증가를 야기하며, 다변량 환경에서는 오랜 시간 수렴하지 않는 모습을 보인다. 따라서, 다변량 환경에서는 τ_{max} 와 α_{PC} 를 적절히 설정함으로써 수렴을 보장하는 동시에 정확도를 최대한 잃지 않도록 하는 것이 중요하다고 생각된다. 또한, LPCMCI를 숨겨진 교란변수가 존재하는 상황에서의 실행 결과 LPCMCI는 교란변수의 존재를 완전하게 파악하지는 못했으나 그 관계를 부분적으로는 파악하였다. 따라서, LPCMCI가 관측되지 않은 변수가 존재하는 상황에서도 실행 가능하며 부분적으로 교란변수의 파악이 가능하다는 사실을 확인할 수 있었다.

References

[1] Andreas Gerhardus and Jakob Runge. High-recall causal discovery for autocorrelated time series with latent confounders. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12615–12625. Curran Associates, Inc., 2020.

[2] P. Spirtes and C Glymour. An Algorithm for Fast Recovery of Sparse Causal Graphs. *Social Science Computer Review*, 9:62–72, 1991.

[3] Glymour C. Spirtes, P. and R Scheines. Causation, prediction, and search. *MIT Press, Cambridge, MA, USA*, 2000.

[4] Giovanni Menegozzo, Diego Dall'Alba, and Paolo Fiorini. Cipcad-bench: Continuous industrial process datasets for benchmarking causal discovery methods, 2022.