

인과추론과 실무 : Causal Discovery with Industrial Case

가짜연구소 인과추론팀

발표자 : 크립토랩 박수영

Contents

1. Introduction to Causal Discovery
2. Preliminary
 - a. PCH & SCM
 - b. Assumptions
3. Methods
 - a. Causal Discovery with Intervention
 - b. Causal Discovery with Observations
4. Applications

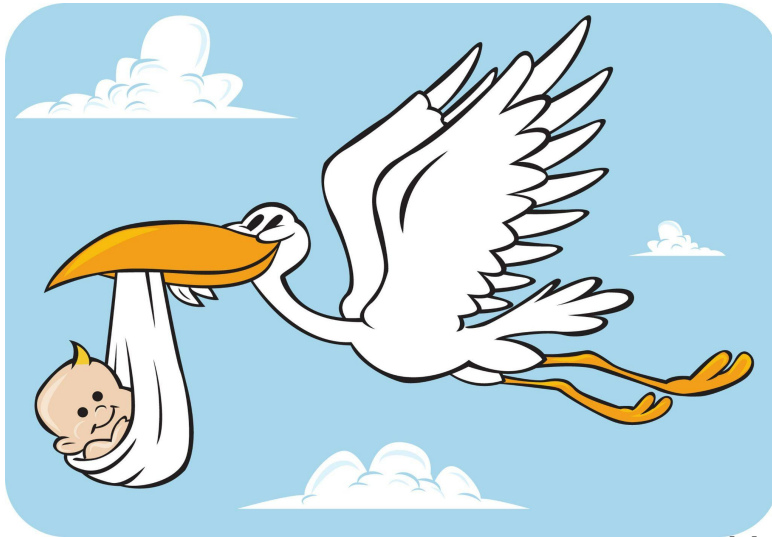
1. Introduction to Causal Discovery

Causal Discovery란?

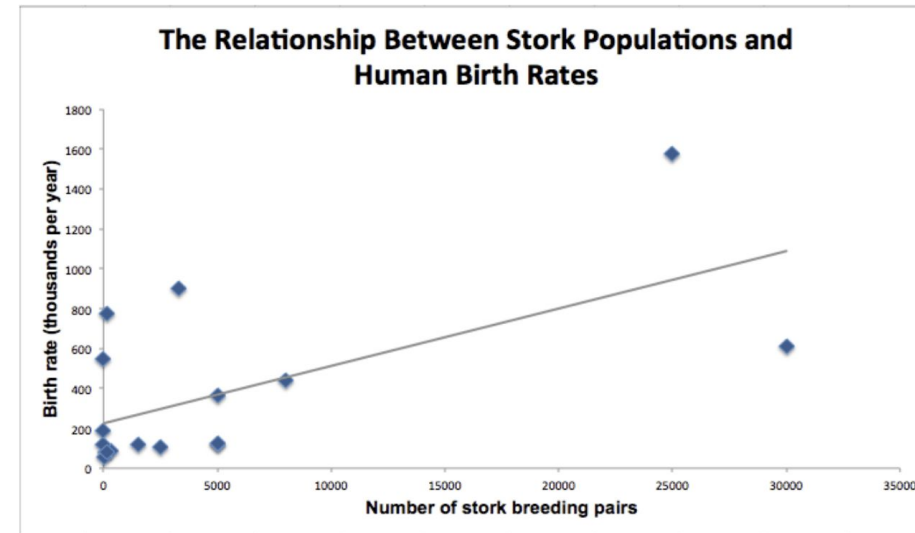
어떤 현상의 원인이 무엇일까? 원인과 결과가 무엇인지 찾을 수 있을까?

즉, **주어진 데이터로부터 원래의 인과 그래프를 찾는(identify) 것**

- Data Generating Process: 인과 그래프 → 데이터
- Causal Discovery: 데이터 → 인과 그래프



[1]



[2]

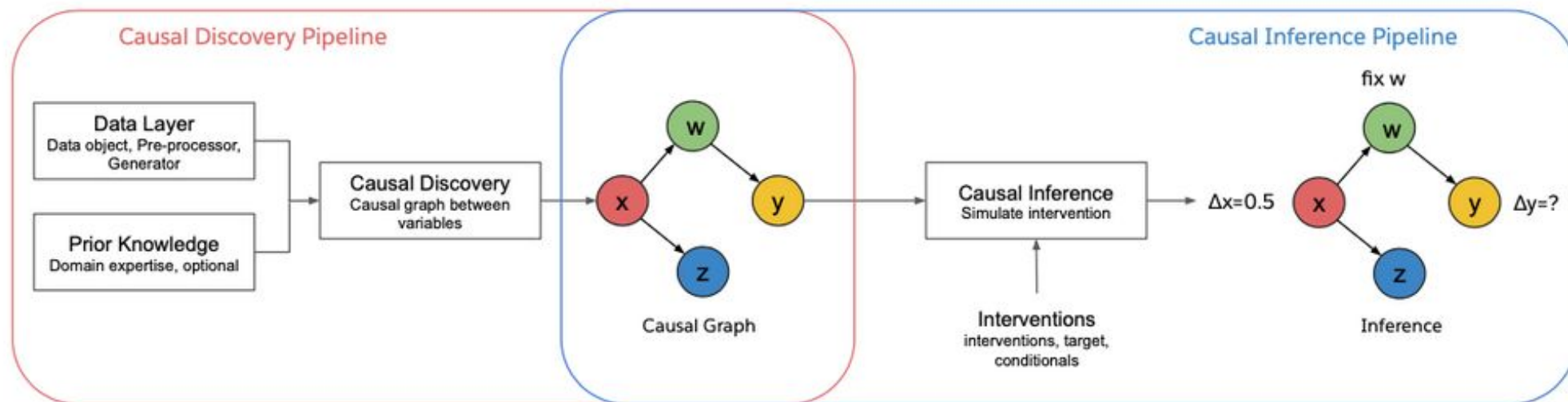
[1] <https://www.calpeculiarities.com/wp-content/uploads/sites/20/2013/02/stork2.jpg>

[2] Matthews, R. (2000). Storks deliver babies ($p=0.008$). Teaching Statistics, 22(2), 36–38.

Causal Discovery vs Causal Inference

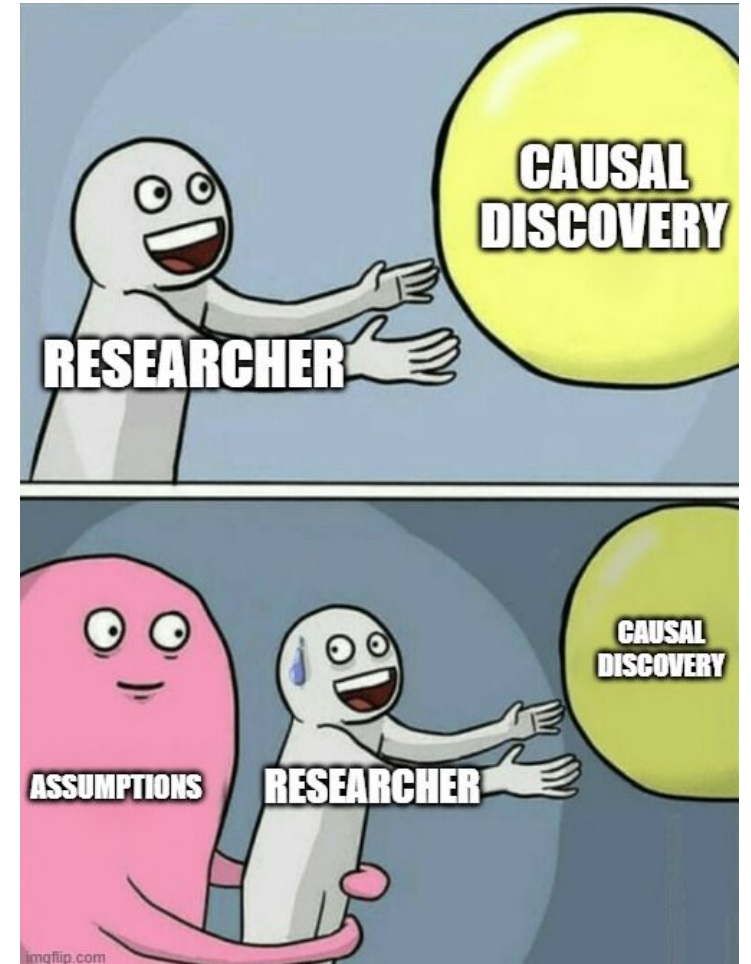
Causal Inference는 그래프가 주어진 상황에서(이미 가정) **처치의 효과를 추정**하는 것

Causal Discovery는 데이터로부터 **그래프를 찾는 것**



Causal Discovery가 대체 왜 어려울까?

- 애초에 인과관계를 명확하게 밝혀 내는게... 어렵다.
우리가 고려하고 있는/데이터로 가진 변수가 세상의 모든 변수들을 고려했을까?
- 기본적으로는 Randomized Experiments로부터만 알 수 있다.
- 실험을 해도 모든 변수에 대해 알 수 없다.
- 가능한 방법들이 있으나, 가정들이 엄격하여 Real-World에서 만족하기 어렵다.

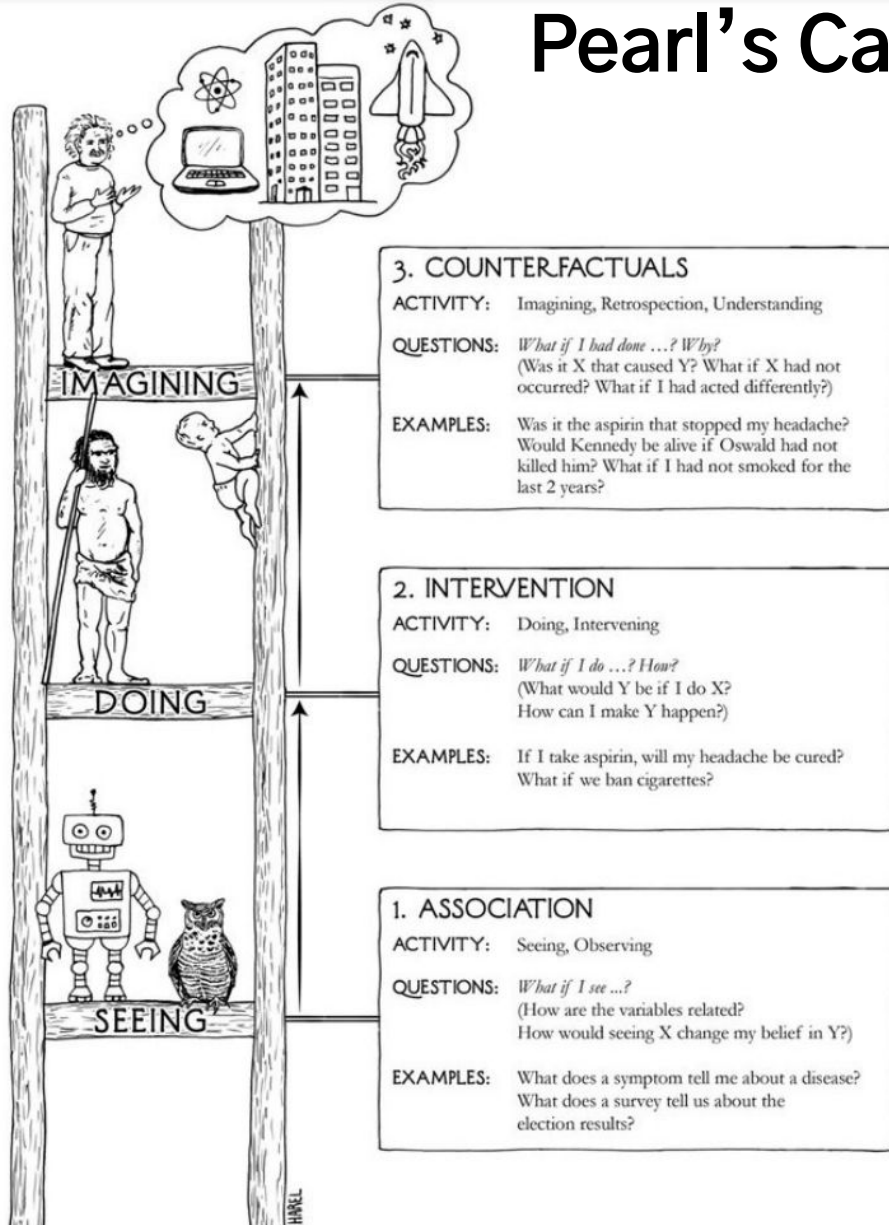


우리가 다룰 내용

- 인과관계를 찾기 위해 필요한 개념이(수학적 notion, 가정 등) 무엇일까?
- Causal Discovery 방법은 어떻게 있을까?
 - intervention을 할 수 있을 때
 - intervention을 할 수 없어, 주어진 데이터만으로 찾고자 할 때
- 실무 적용 예제

2. Preliminary

Pearl's Causal Hierarchy



L3: Counterfactuals

실제로 $(X=x', Y=y')$ 라고 관측했을 때, $(X=x, Y=y)$ 로 나타났을 확률
= Potential Outcomes

$$P(y_x | x', y')$$

L2: Intervention

변수 X에 임의적인 변경을 가하여 값 x로 고정시켰을 때, Y의 확률

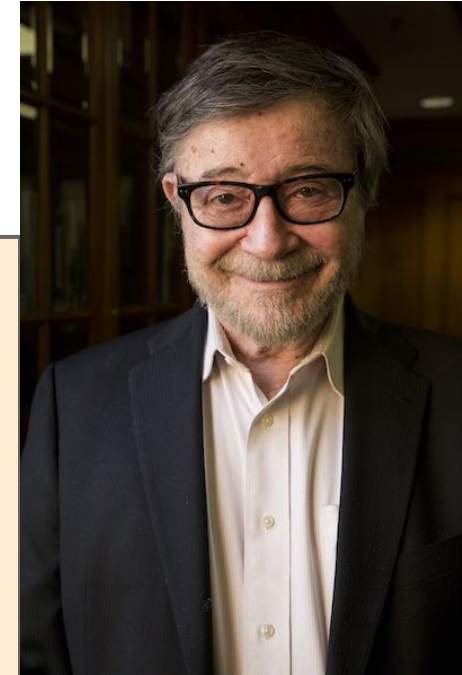
$$P(y | do(x))$$

L1: Observation

조건부 독립

$$P(y | x)$$

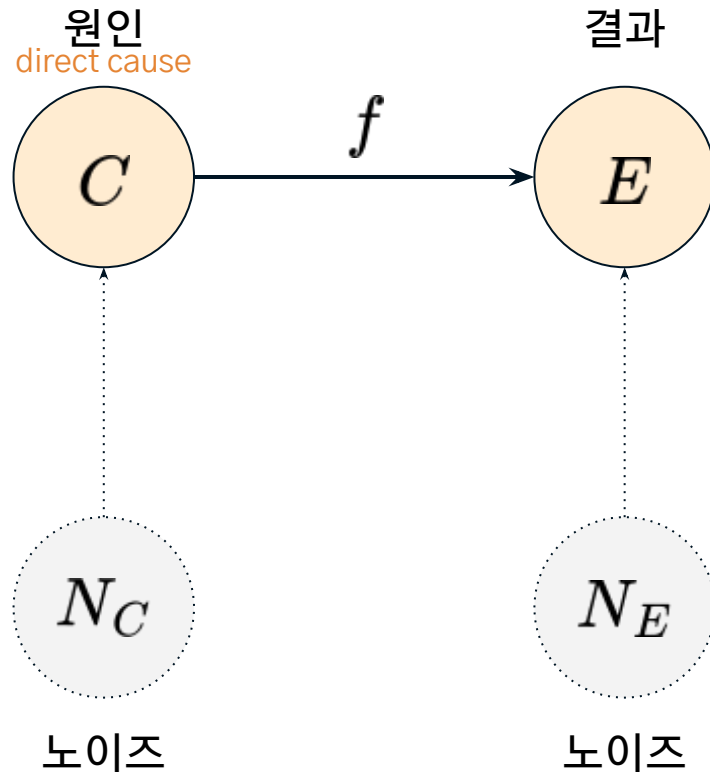
With knowledge from
lower layers,
we cannot say anything
about the higher layers.



하지만, Do-Calculus를 이용하면 실험 없이도 추정할 수 있기도 하다.

SCM (Structural Causal Model)

인과관계를 이해하기 위해서, 세상을 구성하는 변수들이 서로 어떻게 관련있고, 또 상호작용하는지를 설명하는 도구
(이변량의 경우)



causal graph

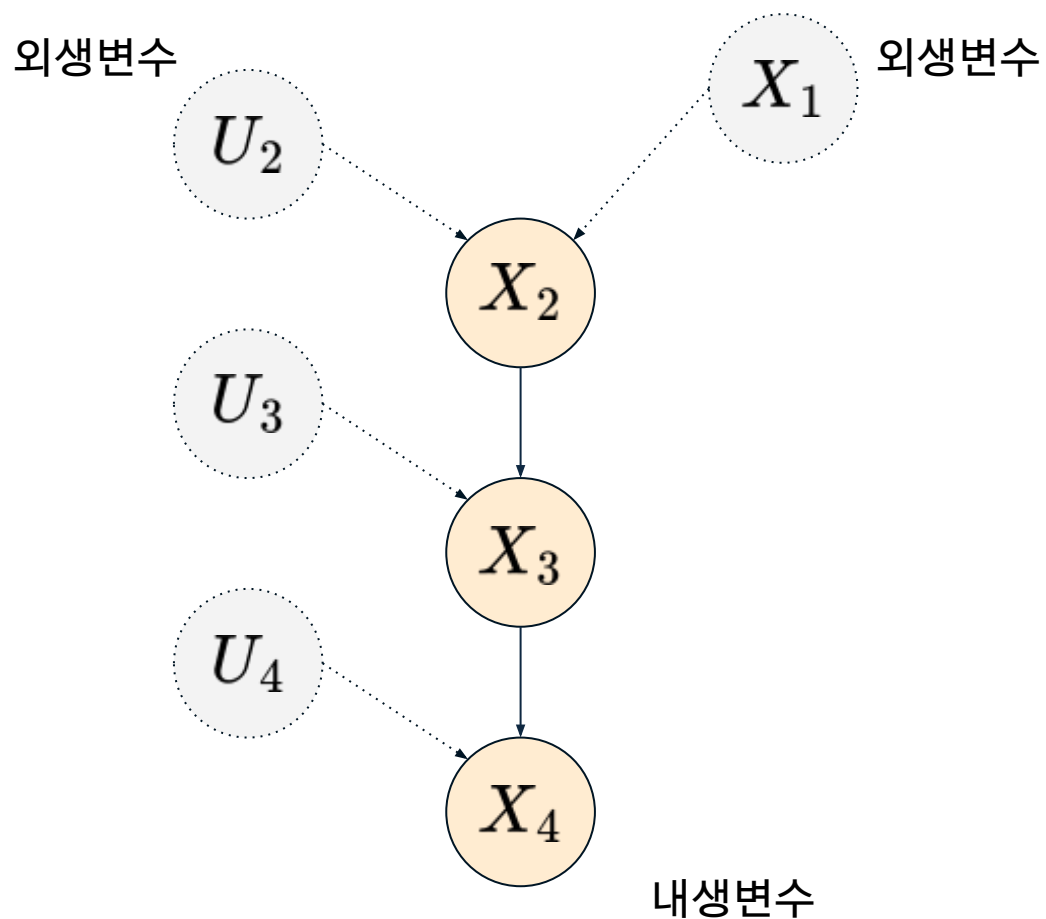
C, E 는 랜덤 변수, N_C, N_E 는 노이즈 변수, f 는 함수일 때,
SCM \mathcal{G} 은 다음과 같은 할당(assignment)을 가진다.

$$C := N_C$$

$$E := f_E(C, N_E)$$

SCM (Structural Causal Model)

인과관계를 이해하기 위해서, 세상을 구성하는 변수들이 서로 어떻게 관련있고, 또 상호작용하는지를 설명하는 도구
(여러 변수의 경우)



X_i 는 내생변수, U_i 는 외생변수, f_{X_i} 는 X_i 값을 결정하는 structural equations 일 때, SCM \mathcal{C} 은 다음 할당을 가진다:

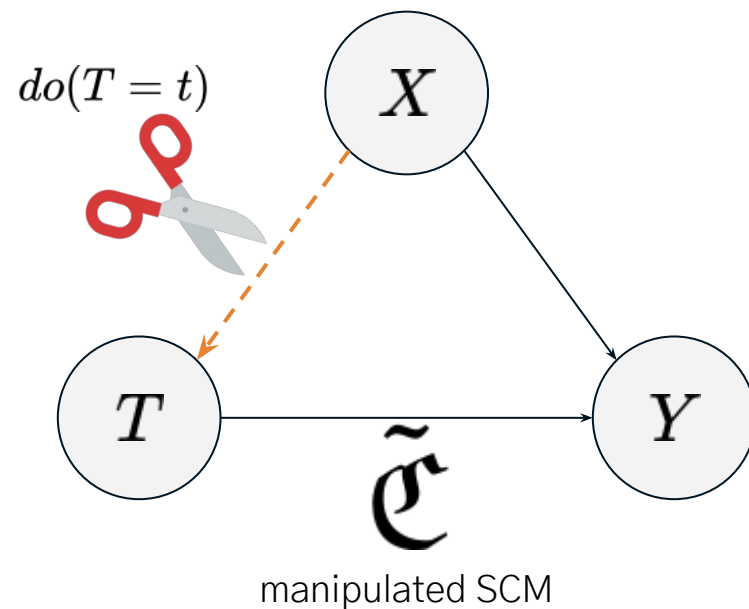
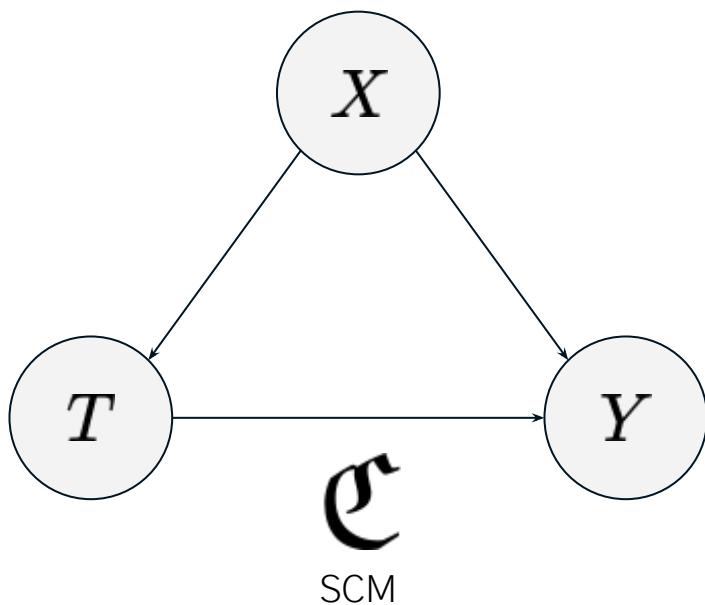
$$X_i := f_{X_i}(PA_{X_i}, U_i)$$

parents

Causal Discovery는 기본적으로 **graph 구조**를 찾는 것을 말하나, SCM에서는 **함수적 관계**를 찾는 것까지 말할 수 있다.

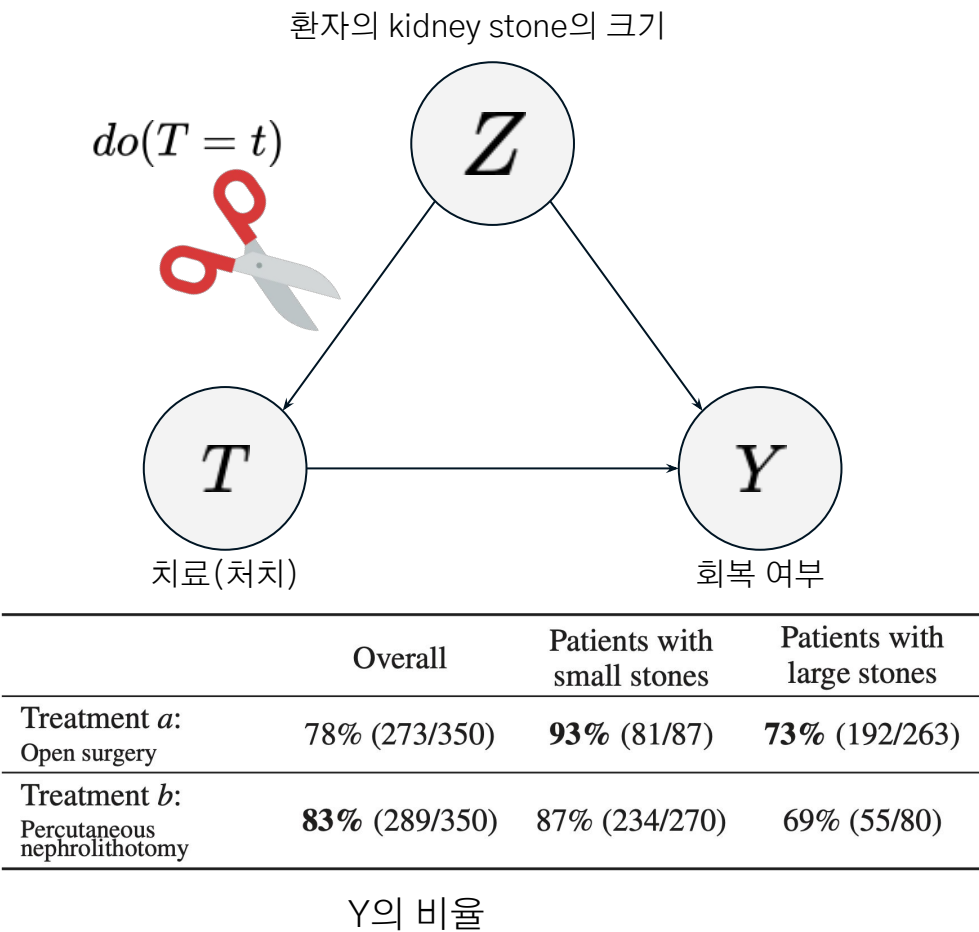
SCM (Structural Causal Model)

SCM을 이용한 Intervention/Counterfactual의 interpretation



Example: Simpson's Paradox

Kidney stone 예시에서 do-operator를 이용해 causal effect를 추정해보자.



$$\begin{aligned} &P(Y|do(T = t)) \\ &:= P^{\tilde{c}}(Y = 1) \\ &= \sum_{z \in \{0,1\}} P^{\tilde{c}}(Y = 1, T = t, Z = z) \quad \text{Marginalize} \\ &= \sum_{z \in \{0,1\}} P^{\tilde{c}}(Y = 1|T = t, Z = z)P^{\tilde{c}}(T = t, Z = z) \quad \text{Bayes' rule} \\ &= \sum_{z \in \{0,1\}} P^{\tilde{c}}(Y = 1|T = t, Z = z)P^{\tilde{c}}(Z = z) \quad \text{Intervention (T=t) \& Manipulated Graph} \\ &= \sum_{z \in \{0,1\}} P^c(Y = 1|T = t, Z = z)P^c(T = t, Z = z) \quad \text{Truncated Factorization} \end{aligned}$$

$$\begin{aligned} &P(Y|T = a) < P(Y|T = b) \\ &78\% \qquad\qquad\qquad 83\% \\ &P(Y|do(T = a)) > P(Y|do(T = b)) \\ &83\% \qquad\qquad\qquad 77\% \end{aligned}$$

Assumptions for Causal Discovery

- **Acyclicity:** SCM은 DAG를 포함하므로, 그래프 내 순환성이 없어야 한다.
- **Causal Markov Assumption:** 그래프 내 각 노드는 오로지 자손(descendant) 노드에만 의존한다.
- **Causal Minimality Assumption:** 불필요한 인과관계는 그래프 내에 존재하지 않아야 한다. (not allow $Y = 0 \cdot X$)
- **Causal Sufficiency:** 그래프 내 모든 변수들의 관측되지 않은 confounder는 존재하지 않는다.
- **Faithfulness Assumption:** 그래프 내 연결된 노드는 확률적으로도 의존한다.

Note: Causal Discovery에 위의 모든 가정이 필요한 것은 아니며, 최근엔 Acyclicity 등 필수 가정도 완화하는 연구들이 있다.



Structural Causal Model vs Potential Outcome Framework

개념	Potential Outcome Framework	Structural Causal Model
처치가 모든 개체에 일관되게 동일	Consistency (SUTVA) (a.k.a. no-multiple-treatment)	Intervention (do-operator)
처치가 대상 외 다른 개체의 결과에 영향 X	No-interference (SUTVA)	Intervention (Modularity assumption)
처치와 (잠재적) 결과 사이의 독립성	conditional Ignorability (a.k.a. exchangeability, exogeneity, unconfoundedness)	Causal sufficiency
데이터 필요	Positivity (for RCT)	Positivity (implicitly)
그래프 비순환성	-	Acyclicity
그래프 내 확률 분포 관련	-	Causal Markov, Faithfulness

양 가정은 비슷한 목표를 가지지만, 다른 구조 상에서 정의되어 본질이 다르므로 동일하다고는 할 수 없음.

Structural Causal Model vs Potential Outcome Framework



	Potential Outcome Framework	Structural Causal Model
Gold Standard	Random Assignment	
Causal Inference Using Observational Data		
(1) Identification (Is it possible to estimate a causal effect?)	Research Design	Backdoor Criterion / do-Calculus
(2) Estimation (How to estimate a causal effect using data?)	Statistical/Econometrics Methods (DID, RD, Matching, IV, SC, etc.)	Statistical/Computational Methods (IPW, Doubly Robust Estimators, Double ML, etc.)

$ATE = \mathbb{E}[Y_i(1) - Y_i(0)]$

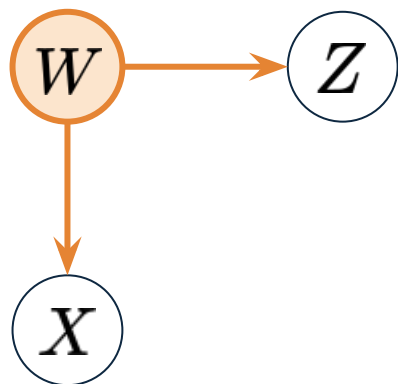
$P(Y \mid do(X = x))$



하지만 선택하는 것이 아니라 둘다 잘 알아야 한다.

Graph와 Conditional Independence

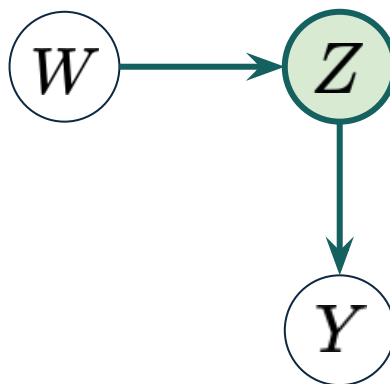
Fork



$$X \not\perp\!\!\!\perp Z$$

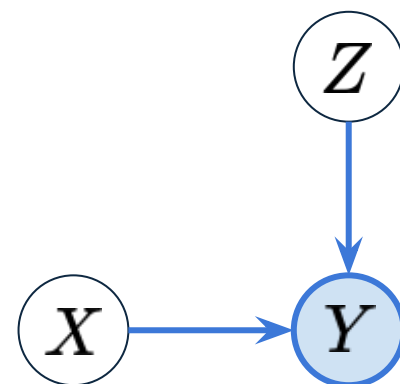
$$X \perp\!\!\!\perp Z \mid W$$

Chain



$$W \not\perp\!\!\!\perp Y$$

$$W \perp\!\!\!\perp Y \mid Z$$

V-structure
(collider/immoralities)

$$X \perp\!\!\!\perp Z$$

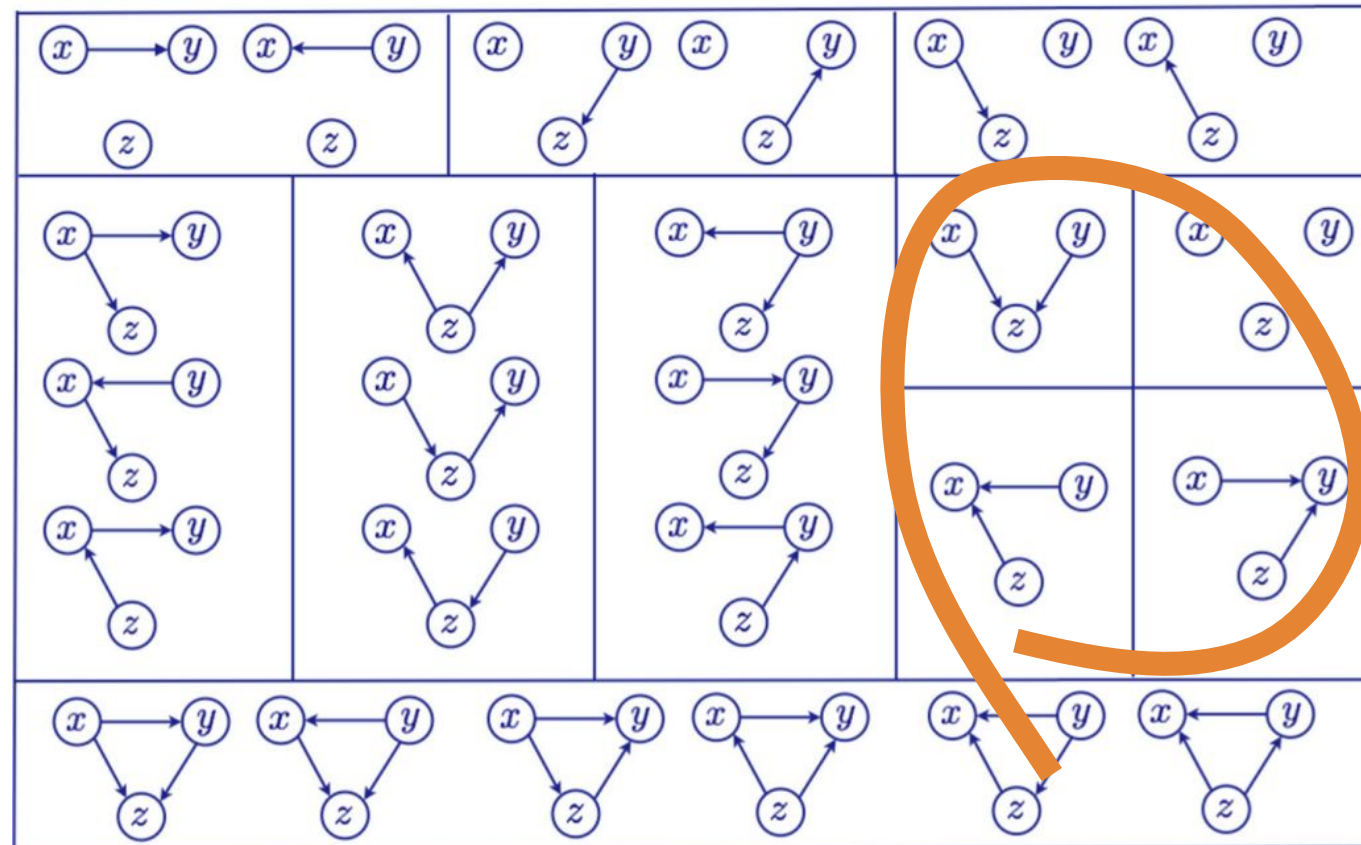
$$X \not\perp\!\!\!\perp Z \mid Y$$

=> 우리는 조건부 독립성과 이 구조들을 이용해 인과관계를 찾을 것이다.

Markov Equivalence Class

하나의 실험을 수행하더라도 수많은 변수들에 대해서 true 인과 그래프를 알 수 없다.

MEC (Markov Equivalence Class)끼리는 조건부 독립 구조로부터 인과관계를 찾아낼 수 없다.



V-structure

그래프의 종류

- DAG (Directed Acyclic Graph): 순환성이 없는 그래프
- PDAG (Partially DAG): 특정 변수 쌍들에 directed path가 없어 부분적으로 DAG인 그래프
- CPDAG (Completed Partially DAG): PDAG 중 모든 MEC를 포함하는 그래프
- BN (Bayesian Network): DAG에 확률적 특성을 부여한 그래프

Note: Causal Discovery 방법 따라 필요한 그래프가 다를 수 있다.

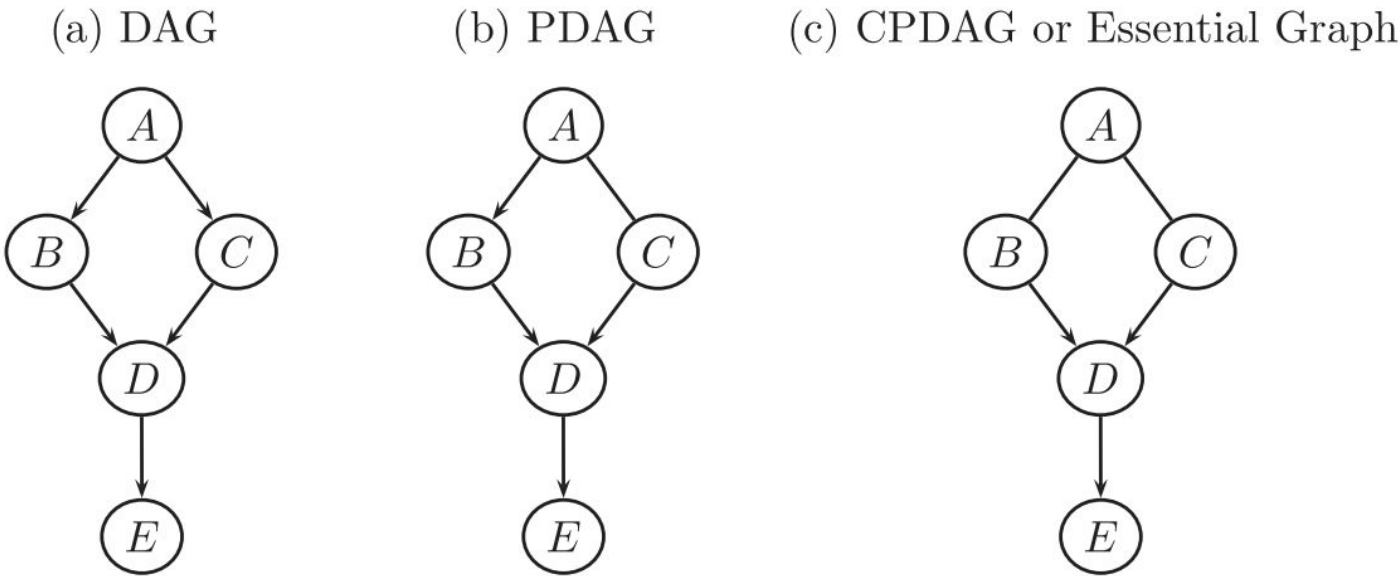


Fig. 1. DAG, PDAG and CPDAD or essential graph.

		Present Relationships	Absent Relationships
1	A → B	A is the cause of B	B is not cause of A
2	A — B	A is the cause of B or B is the cause of A	
3	A ↔ B	This an unmeasured confounder of A and B.	A is not a cause of B B is not a cause of A
4	A ○ → B	Either A is a cause of B or there is an unmeasured confounder of A and B	B is not a cause of A
5	A ○ — ○ B	Exactly one of the following holds: 1. A is a cause of B 2. B is a cause of A 3. There is an unmeasured confounder of A and B 4. Both 1 and 3 5. Both 2 and 3	Note that causal discovery algorithms do not necessarily provide complete causal information

3. Methods

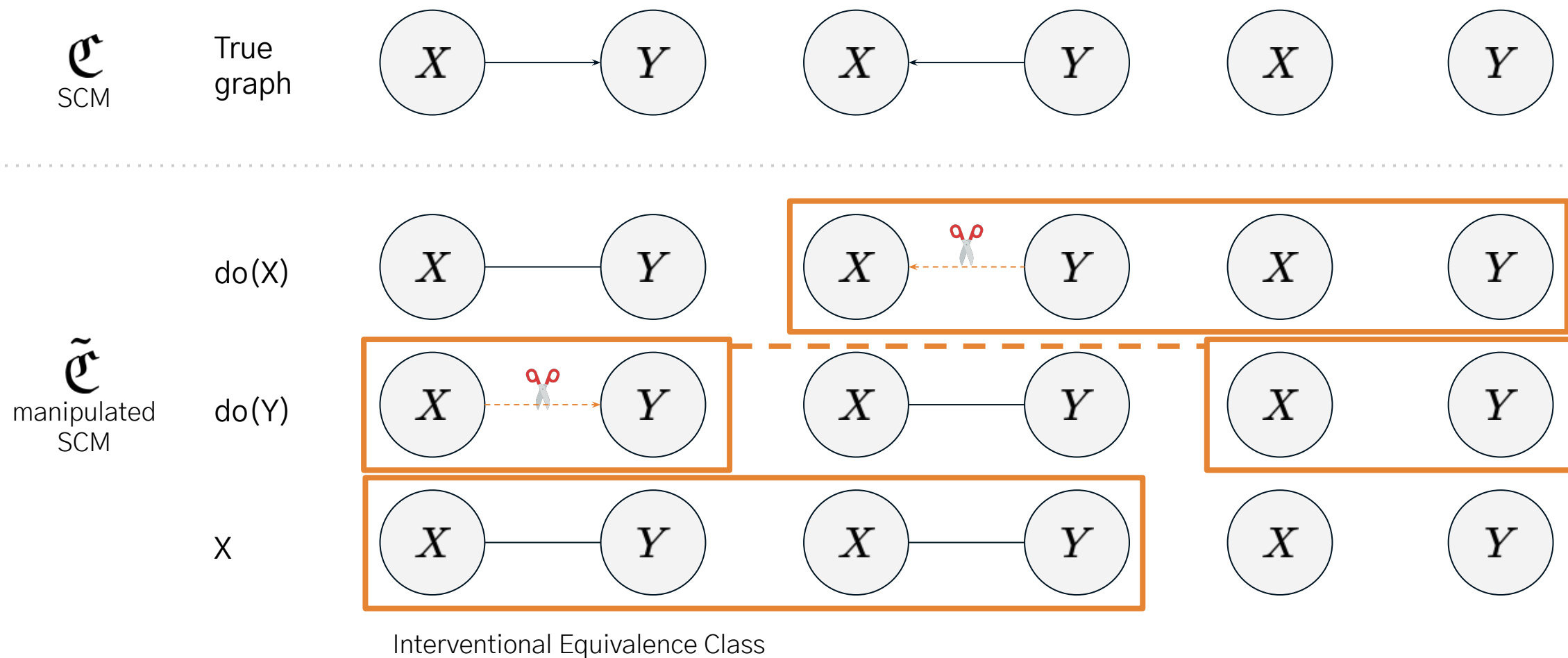
Causal Discovery 방법론

Approach		Description	Algorithms	Assumptions
Causal Discovery with Interventions		Intervention을 이용하여 인과 그래프 식별	(Hard) Intervention, Soft Intervention, single-node intervention, multi-node intervention, ...	- (모두 필요)
Causal Discovery with Observations	Constraint-based	변수 쌍에 대한 조건부 독립성 검정을 통해 fully connected graph로부터 반복적으로 엣지를 제거함	PC, FCI, CCD, ...	- (모두 필요)
	Score-based	데이터에 대한 모델의 fitting score를 이용해 undirected graph로부터 반복적으로 엣지를 더함	GES, (GIES), FGES, ...	데이터 분포 가정 필요
	Functional	함수 클래스에 제약을 가한 뒤 조건부 독립성 검정을 통해 식별	ANM, LiNGAM, PNL, ...	○ 데이터 분포 및 함수 클래스에 대한 가정 필요 ○ faithfulness 가정 불필요

Note: Gradient-based, llm-driven 등 최근 다른 관점의 방법들도 연구/개발되고 있다.

Causal Discovery w/ Interventions

(이변량의 경우)



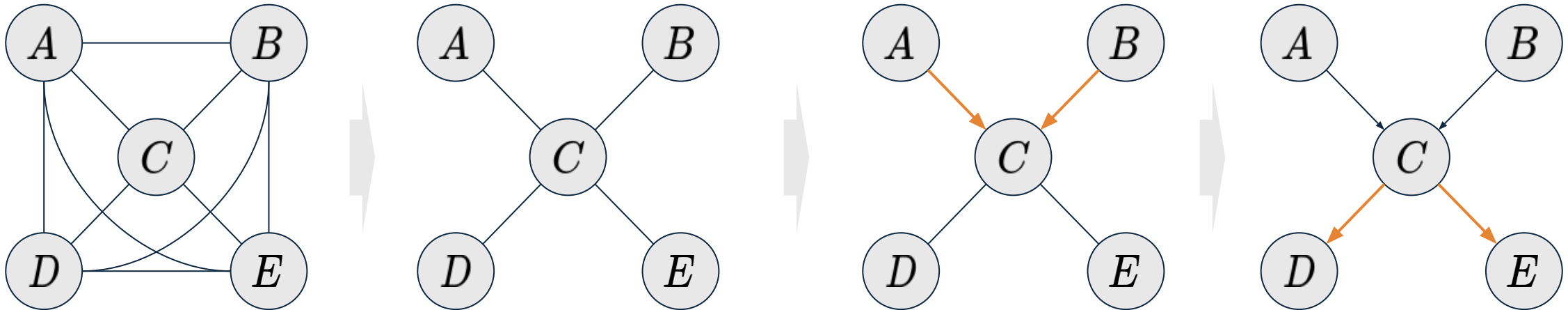
=> 한 번의 intervention으로는 구별할 수 없다.

Causal Discovery w/ Interventions

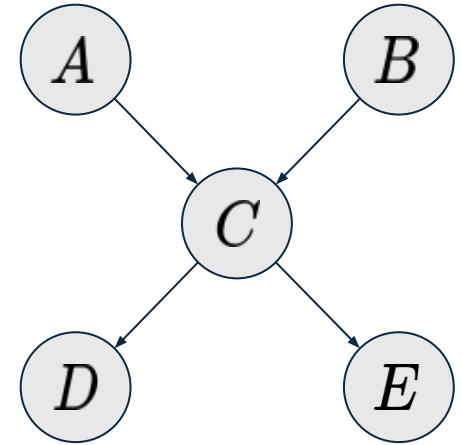
- **두 변수**의 경우 그래프 식별을 위해 **두 번의 intervention**이 필수적으로 필요하다.
- N개의 변수의 경우 최소 N-1번의 intervention 필요 (변수 1개만 intervention 하는 경우)
- 여러 변수에 한 번에 intervention을 한다면? ...
- 특정 값으로 intervention하는 hard intervention이 아닌, 특정 확률 분포 등으로 intervention하는 soft intervention으로 한다면? ...
- 위는 worst case를 고려한 경우이고, Intervention은 일반적으로 costly하므로 **필요한 intervention을 최소한으로 줄여나가면서** 인과 그래프를 찾는 것이 좋다.

Constraint-based CD: PC Algorithm

- PC (Peter-Clark) Algorithm: 변수 (쌍) 간 독립성 제약 기반 방법
- Method
 1. complete undirected graph에서 시작
 2. 조건부 독립성을 이용하여 skeleton(방향성 없이 엣지만 있는 그래프) 식별
 - 조건부 set을 바꿔가며 test: $I=\{\}, \{A\}, \{B\}, \{A, B\}, \dots$
 3. v-structure 식별 & orient: 조건부 독립으로 그래프 구조를 특정시킬 수 있는 형태
 4. v-structure 내 적절한 엣지 방향 맞추기(orient)

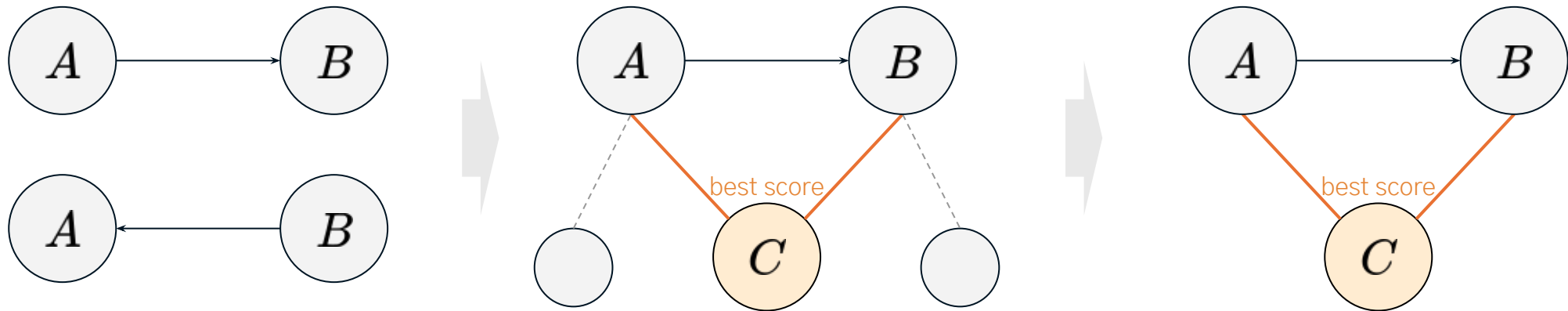


True Graph



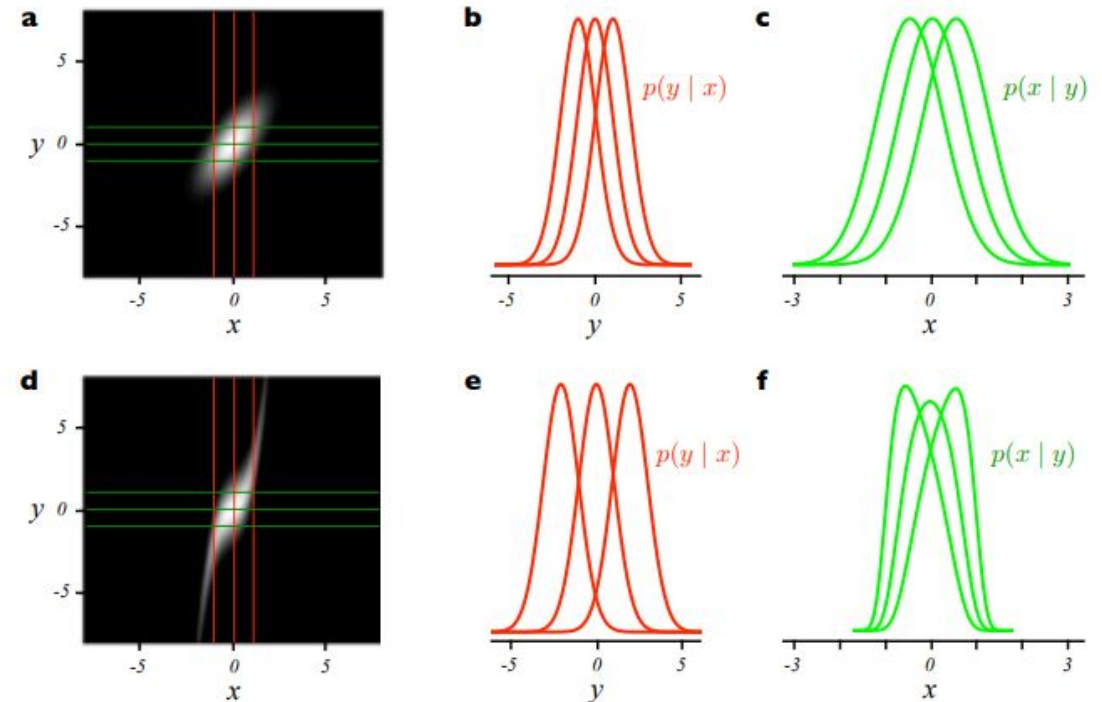
Score-based CD: GES Algorithm

- GES (Greedy Equivalent Search): 관측 데이터의 모델 fitting에 대한 BIC score를 최대화하는 그래프를 찾아나가는 방법
- Method
 1. undirected graph에서 시작
 2. edge를 더했을 때 score가 가장 높을 경우 해당 edge를 더하고 그에 해당하는 MEC까지 ($A \rightarrow B$ 와 $B \rightarrow A$) 고려하여, score를 더이상 향상시킬 수 없을 때까지 반복
 - BIC (Bayesian Information Criterion): $k \log N - 2 \log L$ (낮을수록 모델 fitting score가 높음)
 3. edge를 뺐을 때 score가 가장 높을 경우 해당 edge를 제거하는 것을 score를 더이상 향상시킬 수 없을 때까지 반복



Functional CD: ANM

- ANM (nonlinear causal discovery with Additive Noise Models): 비선형인 인과관계 메커니즘과 additive 노이즈 이용
- 방법: 변수 간 비선형 관계(f)와 Gaussian additive noise를 가정하면, **비대칭성**으로 인해 식별 가능
- 과정 (이변량의 경우):
 1. X,Y간 독립성 검정
 - 독립인 경우 ‘인과관계 없음’으로 결론
 2. 모델 $Y=f(X)+N$ 이 데이터에 consistent한지 검정
 - Y에 대한 X의 비선형 회귀 모델 fitting
 - 잔차 $N=Y-f(X)$ 의 독립성 검정
 - 독립인 경우 현재 모델 채택
 3. 모델 $X=g(Y)+N$ 에 대하여 2. 과정 수행
 4. 결과:
 - 두 모델 다 consistent하면 둘 중 하나 선택 (score 등)
 - 모델 다 inconsistent하면 더 복잡한 인과 모델 혹은 데이터가 필요하다고 결정



4. Applications

Causal Discovery 실무 적용 케이스

- (1) [Databricks] Personalized Incentive Recommender
- (2) [CausaLens] Manufacturing Root Cause Analysis
- (3) [DoorDash] Personalized Restaurant Recommendation

(1) Databricks의 Personalized Incentive Recommender

- **문제:** 프로모션 오퍼(인센티브)가 고객의 구매 행동에 미치는 영향 분석하고 개인화된 오퍼 추천하기
- **접근법:**
 - 인과 그래프를 찾고,
 - 이로부터 인과효과 추정하고,
 - 이익을 최대화하는 고객 인센티브 추천 전략을 개발.



Personalized Incentive Recommender 과정

1. 인과 그래프 찾기 (Causal Discovery): 데이터와 도메인 지식을 이용해 인과 그래프 찾기
2. 식별 및 추정 (Identification and Estimation): 변수 control을 통한 인센티브의 영향 분리
3. Personalized Promotional Offer Recommender: 이익을 최대화하는 인센티브(조합) 추천 전략 제안
4. Robustness Test: Estimator가 얼마나 robust한지 테스트

Personalized Incentive Recommender 과정

데이터와 변수

Outcome

Feature Name	Type	Details
Revenue	continuous	\$ Annual revenue from customer given by the amount of software purchased

We consider three possible treatments, the interventions, whose impact we wish to measure:

Possible Treatment

Feature Name	Type	Details
Tech Support	binary	whether the customer received free tech support during the year
Discount	binary	whether the customer was given a discount during the year
New Engagement Strategy	binary	whether the customer was targeted for a new engagement strategy with different outreach behaviors

Also, a variety of additional customer characteristics that may affect revenue are considered:

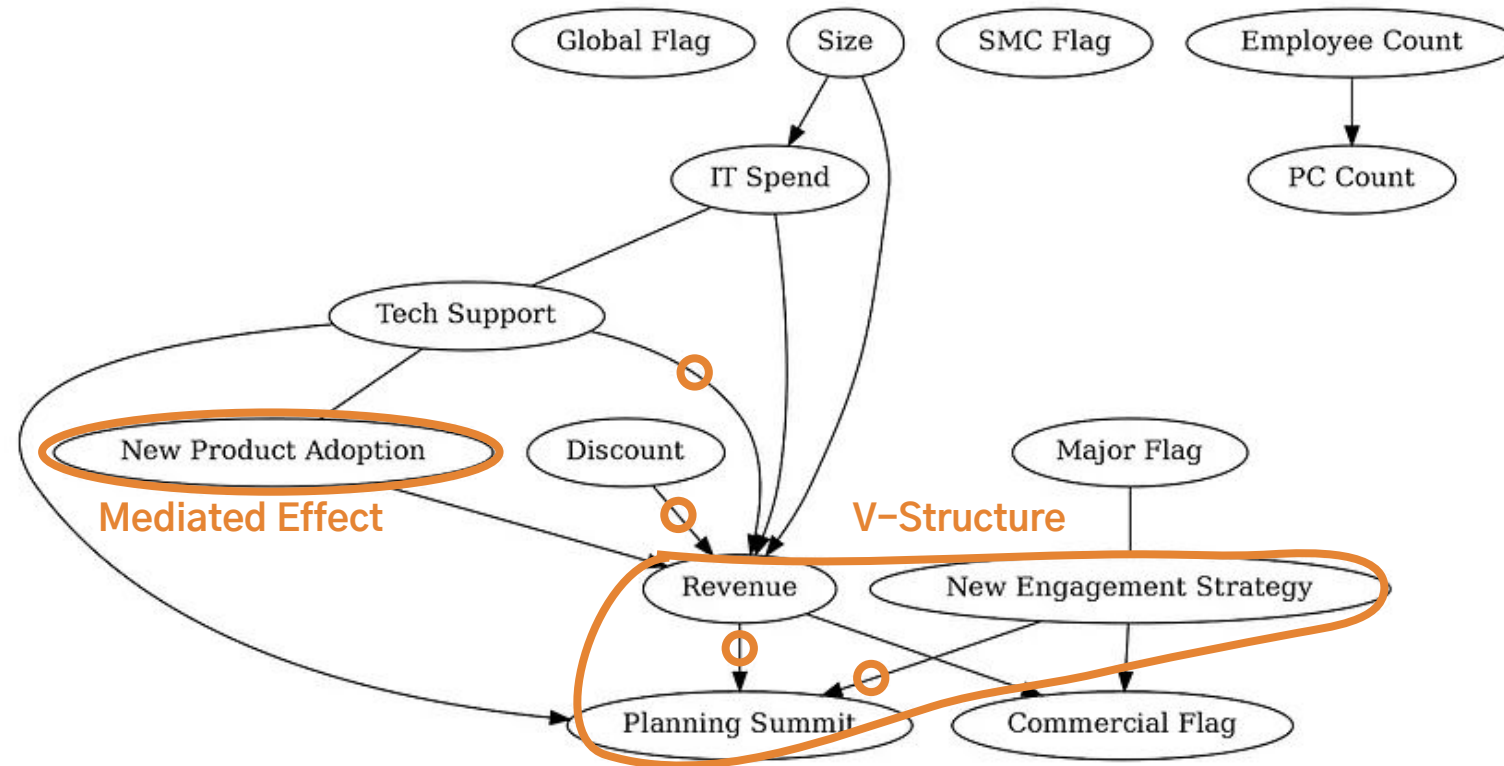
Other Variables

Feature Name	Type	Details
Global Flag	binary	whether the customer has global offices
Major Flag	binary	whether the customer is a large consumer in its industry
SMC Flag	binary	whether the customer is a Small or Medium Corporation (as opposed to large corporation)
Commercial Flag	binary	whether the customer's business is commercial (as opposed to public sector)
Planning Summit	binary	whether a sales team member held an outreach event with the customer during the year
New Product Adoption	binary	whether the customer signed a contract for any new products during the year
IT Spend	continuous	\$ spent on IT-related purchases
Employee Count	continuous	number of employees
PC Count	continuous	number of PCs used by the customer
Size	continuous	customer's total revenue in the previous calendar year

Note: 여기서 데이터는 시뮬레이션 데이터이고, 실제 인과관계는 도메인 지식 등으로 알고 있다고 전제.

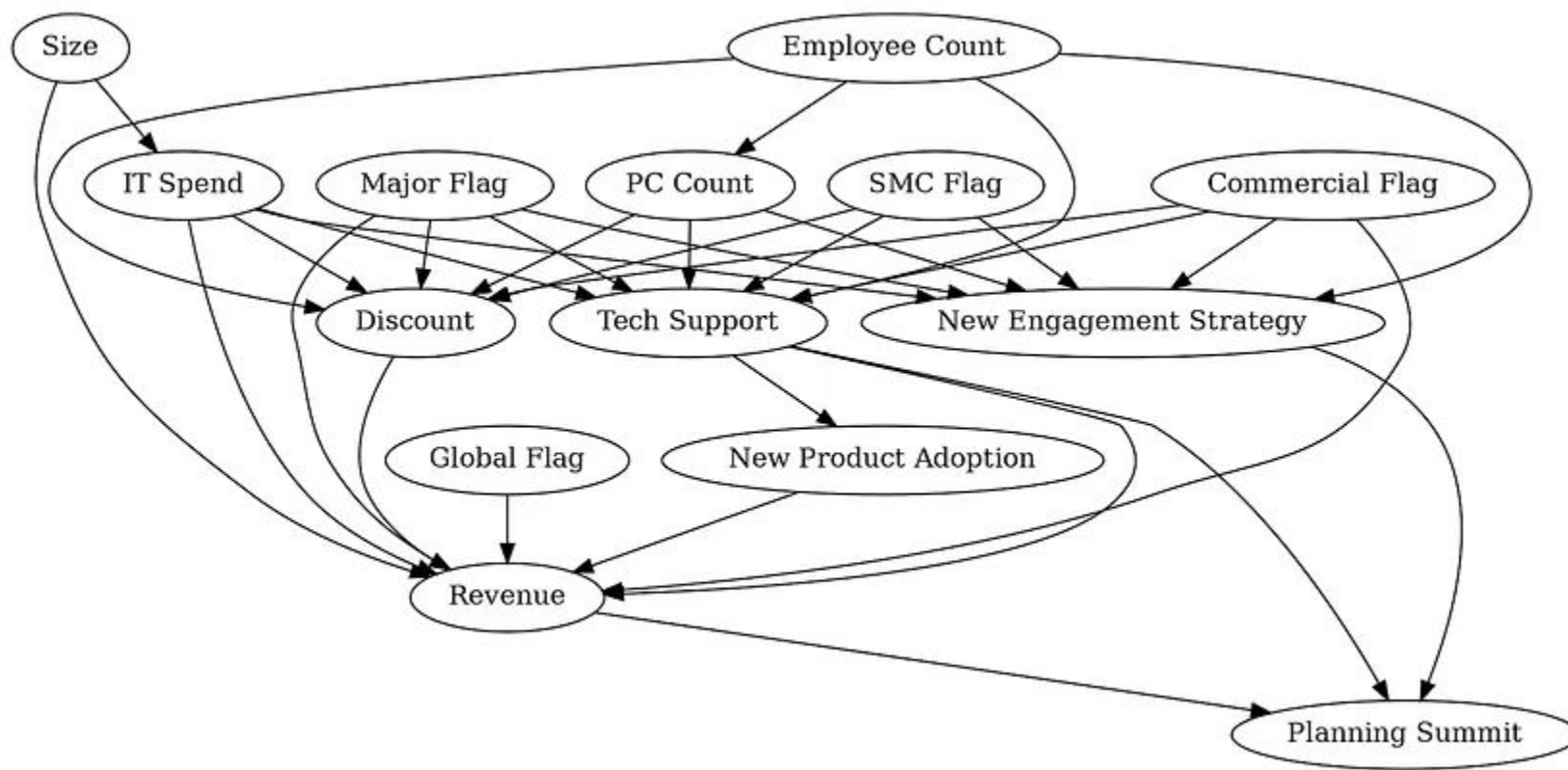
인과 그래프 찾기

- PC Algorithm을 이용하여 인과 그래프를 찾음
- Collider 변수인 Planning Summit이 conditioning되면, New Engagement Strategy와 Revenue에 spurious correlation 발생하므로 Treatment Effect 추정 시 유의해야 함.



인과 그래프 찾기

- 도메인 지식을 이용해 최종 그래프 생성



Note: 사실 Causal Discovery에서 도메인 지식은 매우 중요하다... 어쩌면 가장...

식별 (Identification)

- Backdoor Criterion 하에서 Do-Calculus를 수행하여 Identification(효과를 추정할 수 있는지 확인)

```
# Instantiate a model object to estimate the tech support
tech_support_effect_model = dowhy.CausalModel(
    data=input_df, graph=graph, treatment="Tech Support",
)

# Identify methods we can use to estimate the tech support
tech_support_total_effect_identified_estimand = (
    tech_support_effect_model.identify_effect(
        estimand_type="nonparametric-ate",
        method_name="maximal-adjustment",
    )
)

# Print out all identified estimands
print(tech_support_total_effect_identified_estimand)
```

```
### Estimand : 1
Estimand name: backdoor
Estimand expression:
    d
    -----(E[Revenue|Employee Count,Global Flag,PC Count,Major Flag,SMC F
d[Tech Support]

lag,Size,Commercial Flag,IT Spend,New Engagement Strategy,Discount])

Estimand assumption 1, Unconfoundedness: If U-{Tech Support} and U-Revenue then P(Revenue|Tech Support,Employee Co
e|Tech Support,Employee Count,Global Flag,PC Count,Major Flag,SMC Flag,Size,Commercial Flag,IT Spend,New Engagemen

### Estimand : 2
Estimand name: iv
No such variable(s) found!

### Estimand : 3
Estimand name: frontdoor
No such variable(s) found!
```

* Backdoor Criterion, Adjustment and Do-Calculus

- 인과 그래프를 알고 있을 때, 주어진 조건 하에서 관측 데이터(L1)만으로도 처치 효과를 추정(L2)할 수 있다.

- Backdoor Criterion

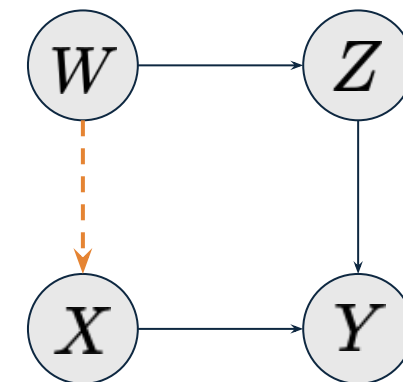
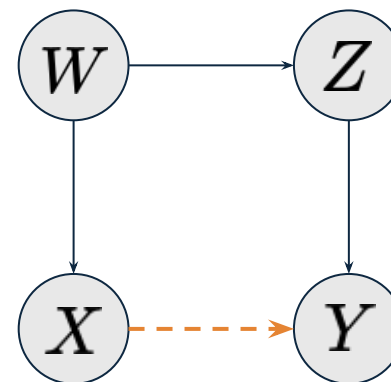
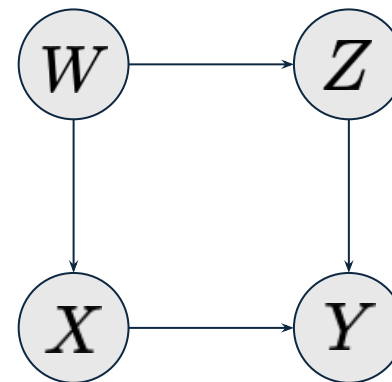
“backdoor criterion”: Any $\mathbf{Z} \subseteq \mathbf{X} \setminus \{X, Y\}$ with

- \mathbf{Z} contains no descendant of X AND
- \mathbf{Z} blocks all paths from X to Y entering X through the backdoor
($X \leftarrow \dots$, see Figure 6.5)

is a valid adjustment set for (X, Y) .

- Backdoor Adjustment

$$\begin{aligned}
 P^{\tilde{c}}(y) &= P(y|do(X := x)) \\
 &= \sum_z P(y|do(X := x), z)P(z|do(x)) \quad \text{Marginalize} \\
 &= \sum_z P(y|x, z)P(z|do(x)) \quad \text{Rule 2} \\
 &= \sum_z p(y|x, z)p(z) \quad \text{Rule 3}
 \end{aligned}$$



추정 (Estimation)

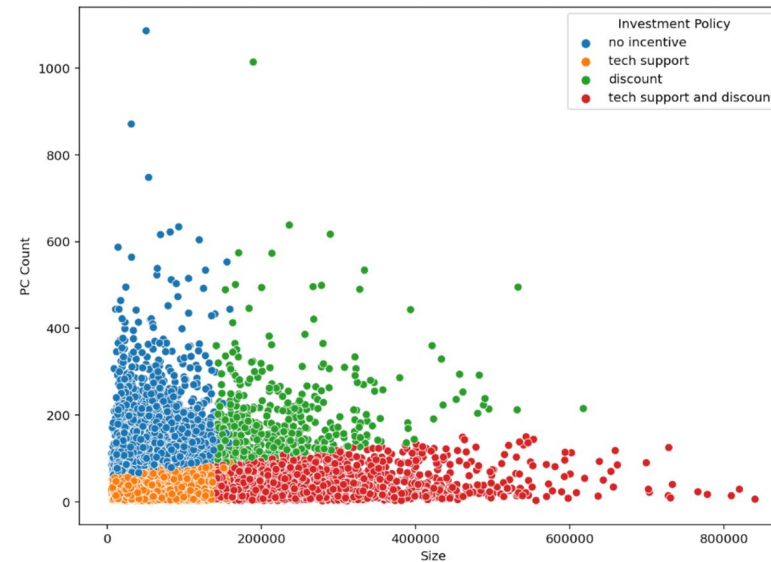
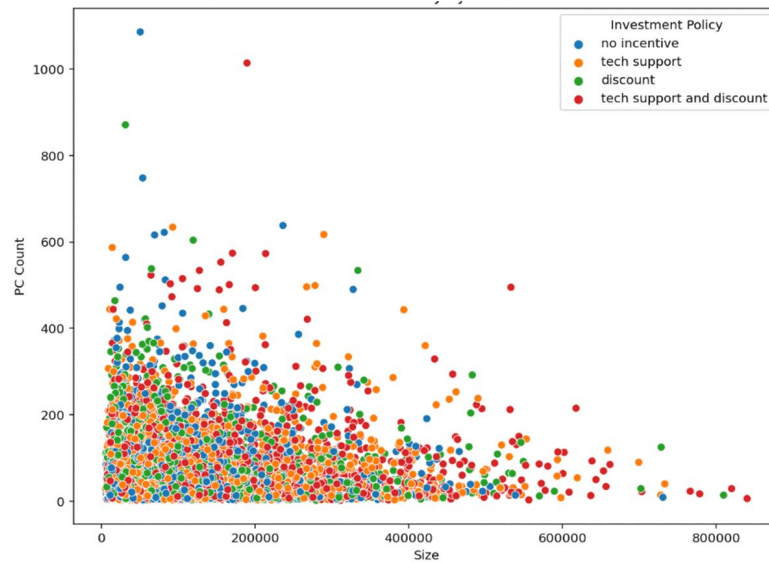
- Double Machine Learning을 이용해 추정
- 각 변수가 Revenue에 미치는 Average (Total) Effect를 추정
 - Tech Support의 경우 Mediator인 New Product Adoption의 영향을 뺀 Direct Effect도 추정
- 프로모션 오퍼(로지스틱 회귀 모델) 및 수익(Lasso 회귀 모델) 추정
- 그 결과, 다른 변수와 달리 New Engagement Strategy는 결과에 미치는 영향이 없음을 확인.

```
1 # Disable the mlflow autolog feature
2 mlflow.autolog(disable=True)
3
4 # Set up the treatment (t) and outcome (y) models for DML.
5 # See notebook-config for detail.
6 model_t, model_y = setup_treatment_and_out_models()
7
8 # Specify the effect modifiers, which are variables
9 # that can change the magnitude of the effect based on the groups.
10 effect_modifiers = ["Size", "Global Flag"]
11
12 # Specify the estimand recommended in the previous cell
13 method_name = "backdoor.econml.dml.LinearDML"
14
15 init_params = {
16     "model_t": model_t,
17     "model_y": model_y,
18     "linear_first_stages": True,
19     "discrete_treatment": True,
20     "cv": 3,
21     "mc_iters": 10,
22 }
23
24 # Estimate the effect of tech support
25 tech_support_total_effect_estimate = tech_support_effect_model.estimate_effect(
26     tech_support_total_effect_identified_estimand,
27     effect_modifiers=effect_modifiers,
28     method_name=method_name,
29     method_params={"init_params": init_params},
30 )
31
32 # Extract the interpretation of the estimate
33 tech_support_total_effect_estimate.interpret()
```

글에 결과는 안나와있네요..

Personalized Promotional Offer Recommender

- ‘Discount’와 ‘Tech Support’를 고려한 4가지 프로모션 오퍼 전략
 - 없음
 - Discount 제공
 - Tech Support 제공
 - Discount와 Tech Support 모두 제공
- 전략 유무에 따라 고객당 평균 marginal profit을 비교하였더니 차이가 꽤 크다.



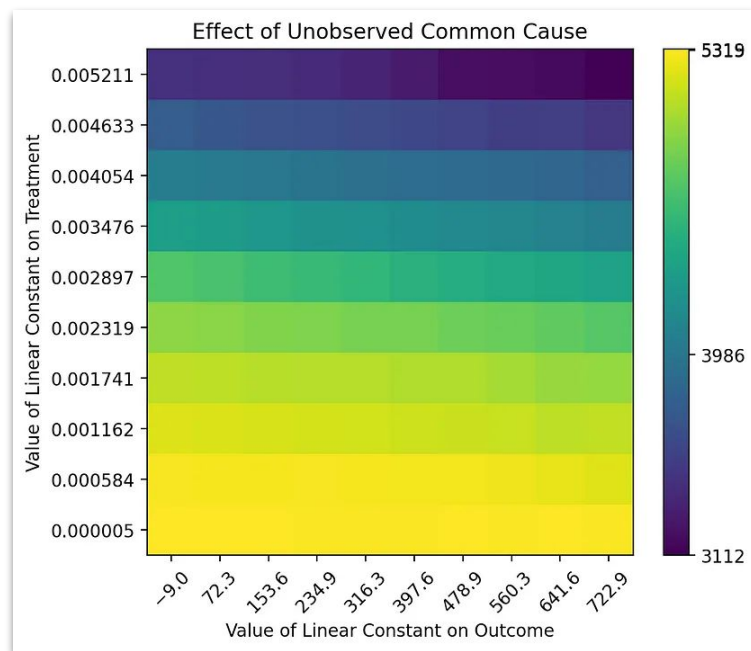
Average marginal profit per account	
Always giving only 'tech support'	1814.528839
Always giving only 'discount'	-1685.011662
Always giving 'tech support' and 'discount'	129.517176
Giving no incentive	0.000000
Giving recommended incentive	4959.563942
No Policy	311.483533

Robustness Test

- Estimator가 얼마나 robust한지를 많은 가정들을 파괴하는 시뮬레이션 데이터로 테스트
- 테스트 방법:
 - placebo treatment refuter: treatment를 무작위로 섞어 treatment와 outcome의 causality 제거하고, 실제로 잘 추정되지 않은지 확인
 - 그래프에 confounder를 추가하여 이에 얼마나 민감한지 테스트: 하나는 measured, 하나는 unmeasured confounder를 추가하여서 비교해도 여전히 효과가 존재하는지 확인
 - 그외 [2] 참조

Stale widget: cannot display widget because the python repl changed. Please rerun the notebook

	Refutation Type	Estimated Effect	New Effect	Refutation Result (p value)
0	Refute: Use a Placebo Treatment	5314.988338	2.410651	0.96



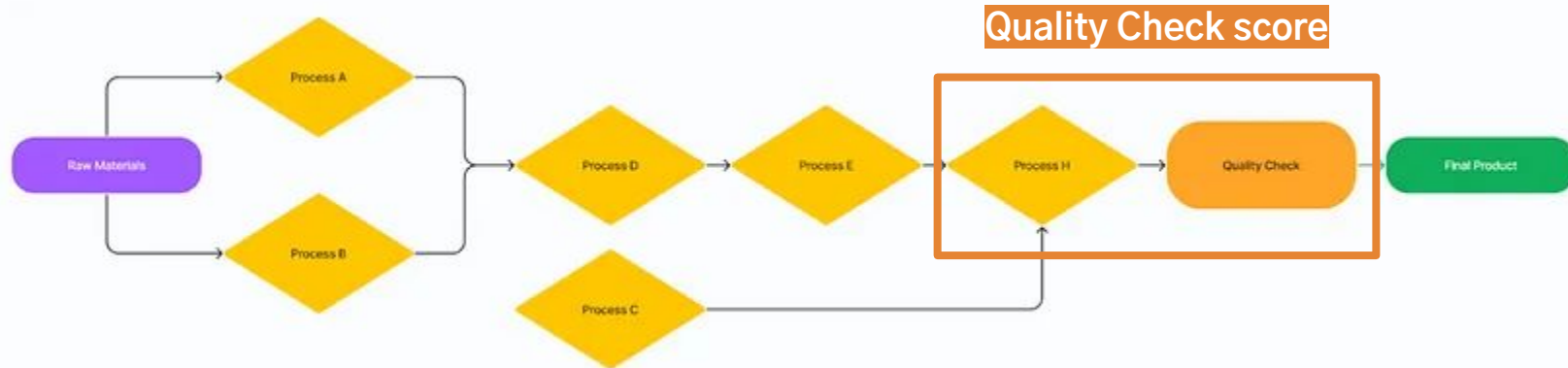
[1] <https://www.databricks.com/blog/optimizing-promotional-offers-using-causal-machine-learning>

[2] Microsoft Research, Foundations of causal inference and its impacts on machine learning webinar, <https://www.youtube.com/watch?v=LALfQStONEc>

(2) 제조업에서의 Root Cause Analysis

- 문제: 결함에 대한 근본 원인을 찾아내기!
 - + 빠른 속도 및 설명가능성을 포함하는 방법
- 접근법:

도메인 지식과 결합(Human-Guided Causal Discovery)하여 인과 관계를 찾은(Causal Modeling) 후, 근본 원인 분석(Root cause analysis decision intelligence)하고, 제조 공정을 최적화(Process optimization)함.



생산라인의 개략적인 구조

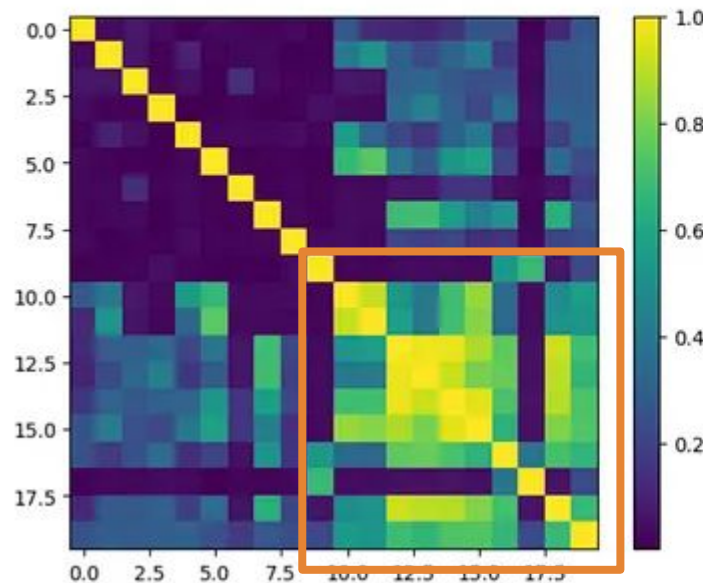
Note: CausaLens의 Causal AI 솔루션인 decisionOS를 이용함.

Root Cause Analysis 과정

1. 데이터 설정: 데이터 로드 및 EDA
2. 이상치 탐지: 프로세스 H에서 수행한 측정을 기반으로 Quality Check 점수를 내어 threshold 기반으로 결함 파악
3. 인과모델 Specification: 도메인 지식 정의, 인과그래프 및 SCM 찾기
4. RCA (Root Cause Analysis): 제조 공정 내 근본 원인 식별 및 경로 시각화

데이터 전처리

1. 데이터 설정: 데이터 로드 및 EDA
2. 이상치 탐지: 프로세스 H에서 수행한 측정을 기반으로 Quality Check 점수를 내어 threshold 기반으로 결함 파악



[1. 데이터 설정] EDA 결과: 변수 간 상관행렬

```
count    10200.00
mean         0.02
std         0.12
min          0.00
25%          0.00
50%          0.00
75%          0.00
max          1.00
Name: Quality Check, dtype: float64

Proportion of Failures: 0.017
```

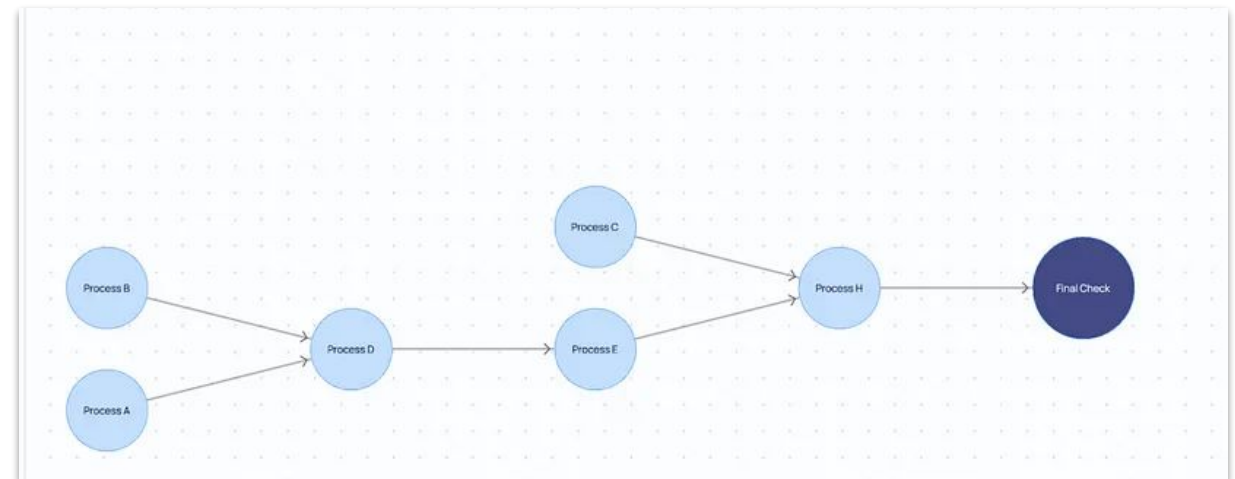
[2. 이상치 탐지] 데이터 내 결함의 비율

인과모델 정의: 도메인 지식 반영

- 도메인 지식으로부터 각 프로세스를 grouping하고 각 계층(tier) 별 상호작용을 그래프로 표현
- 엣지 제약사항 반영 (프로세스 H에서만 품질 검사)
- => 도메인 지식을 반영한 제조 공정 그래프 생성

```
# First we separate each group of variables out into their subprocesses
tiers = {
    'Process A':['A_01', 'A_02', 'A_06', 'A_07'],
    'Process B':['B_02', 'B_03', 'B_06', 'B_07', 'B_08'],
    'Process C':['C_08'],
    'Process D':['D_04', 'D_05'],
    'Process E':['E_04', 'E_05'],
    'Process H':['H_15', 'H_17', 'H_28', 'H_30', 'H_38'],
    'Final Check':['Quality Check']
}
```

```
# Enforce that each H measure has a direct causal impact on Quality Check and nothing else contributes
domain_knowledge = {}
for i in tiers['Process H']:
    domain_knowledge[(i, 'Quality Check')] = EdgeConstraint.FORWARD_DIRECTED_EDGE
for tier in tiers:
    if tier not in ['Process H', 'Final Check']:
        for i in tiers[tier]:
            domain_knowledge[(i, 'Quality Check')] = EdgeConstraint.FORBIDDEN_EDGE
```



인과모델 정의: 인과 그래프 찾기

- FCI Tiers 알고리즘 이용

```
from causal_discovery import FCITiers

# Initialise our Causal Discovery object
cd = FCITiers(causal_knowledge=ck, oracle='monotonic', max_sepset_size=5)

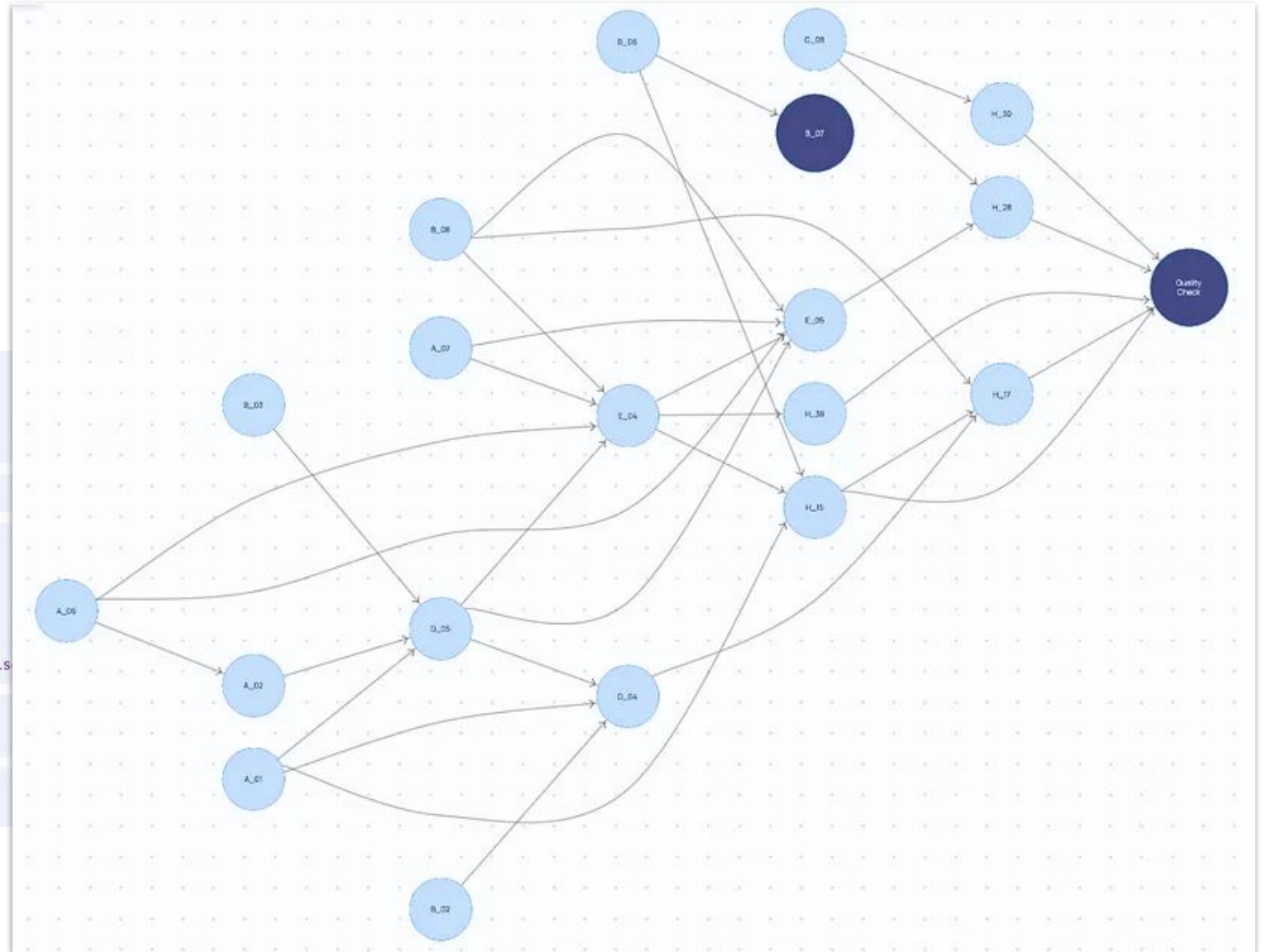
cd_result = cd.fit_predict(data)

causal_discovery.scores.bic import BICScore
causal_discovery.utils.dag_sampling import DAGSampling

sal Discovery outputs a Causal Graph object but this is only a Markov Equivalence Class
need to sample DAGs from this MEC and score them to select the best Causal Graph for our problem.
DAGSampling(graph=cd_result, causal_knowledge=ck, add_latent_confounders=False, include_selection_bias=False)

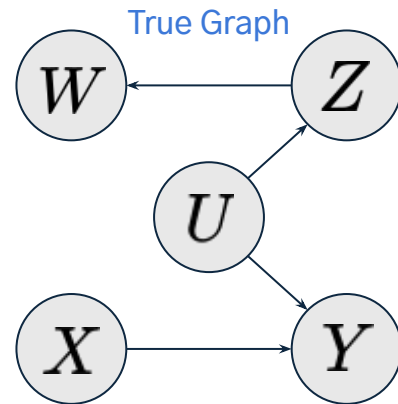
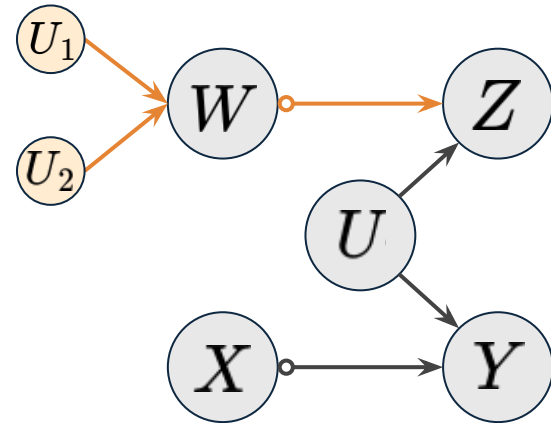
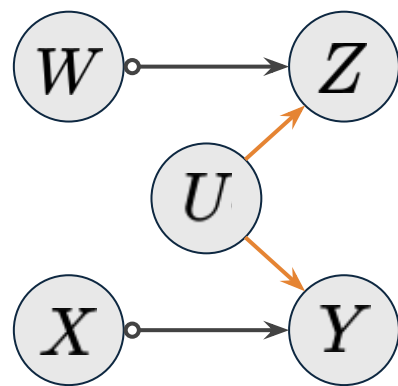
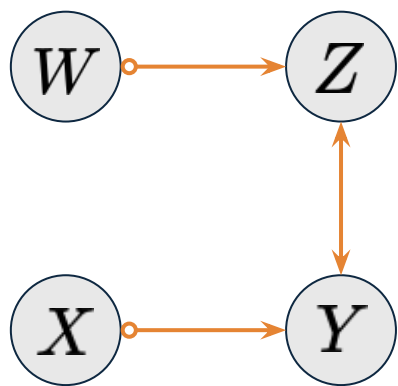
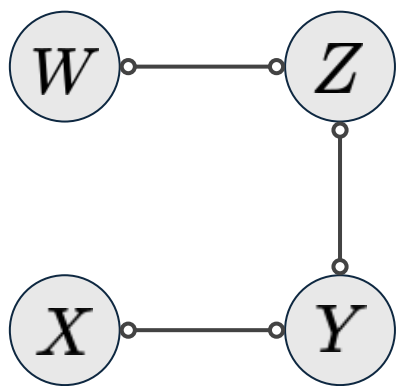
scorer = BICScore()
scores = [scorer.score_full_graph(data=data, dag=i.to_networkx())[0] for i in cgs]

# We select the graph with the best score
best_graph = cgs[numpy.argmin(scores)]
```



* FCI Algorithm

- Fast Causal Inference (FCI) Algorithm: confounder를 허용하는 PC 알고리즘 (constraints-based CD)
 - 즉, causal sufficiency 가정 불필요
- 과정:
 - undirected graph(confounder 가능)에서 시작
 - 조건부 독립성을 이용하여 skeleton 식별 (PC Algorithm과 동일)
 - V-structure 찾기: V-structure가 양 옆으로 있으면 unmeasured confounder가 있음
 - Y-structure 찾기: Y-structure가 있으면 unmeasured confounder가 없음



인과모델 정의: SCM 학습

- CausalNet 모델을 사용하여 SCM 찾기
 - functional relation까지 찾는 방법
 - (..회사 confidential이라 공개가 안된듯)

```
# Define a CausalNet instance from the CausalGraph instance
net = CausalNet.from_graph(best_graph)

# Our Quality Check score is normalised between 0 and 1
# We can use an inverse sigmoid to transform this into a continuous score more suitable for regression
# When analysing our results we can easily apply a sigmoid to then convert the output back into the original scale

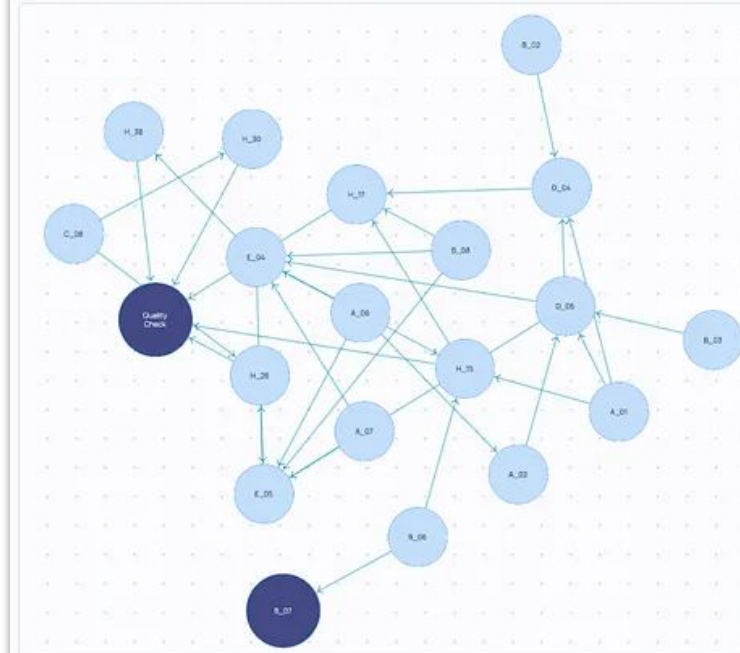
rca_data = data.copy()
rca_data['Quality Check'] = numpy.log(1/(1/(rca_data['Quality Check']-1e-8)-1))

# Train the CausalNet instance using the available data
# Through our domain knowledge of the problem we know that all the relationships are positive
net.node['Quality Check'].aggregation = 'sum_with_bias'
for edge in net.edges:
    edge.activations = 'positive_linear'

# We then train our SCM on the data using the cvxpy engine
net.train(
    inputs=rca_data,
    engine='cvxpy',
    l1=0.0,
    l2=0.0
)
```

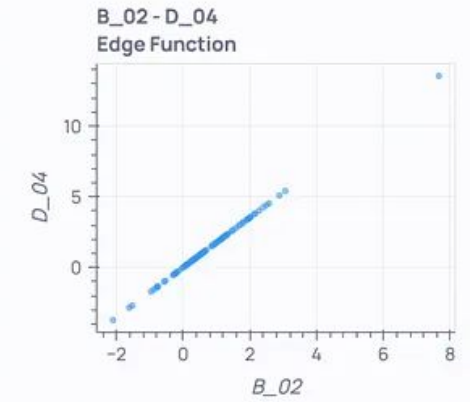
Causal Graph Explorer - Edge Explanations

Understand the causal relationship of the different variables on the target.



Edge Functional Form

Feature-Target Relationship



This function is the learned function for the edge from B_02 to D_04.

More Info

RCA, 근본 원인 분석

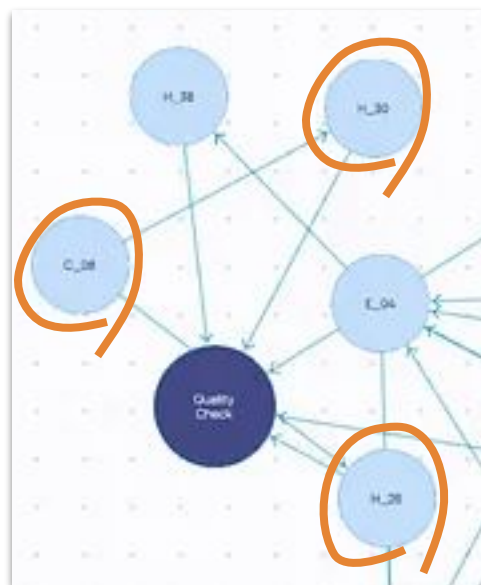
- 어떤 변수가 특정 값으로 통제했을 때 결과를 바꾸는지 확인하여 근본 원인을 식별
- 두 가지 방법: InterventionalRCA, **CounterfactualRCA**
- CounterfactualRCA: 불리한 목표 결과에 대한 관찰을 사용하여 데이터의 노이즈를 추정하여 예측에 도움.
 - 여기서, 변수 간 path가 있으면 이 영향을 **분리**시켜야 하는데, 여기선 유사도 거리 기반 방법으로 해결

```
# instantiate an RCA object
rca = CounterfactualRCA(causal_model=rca_net)

res = rca.get_root_causes(
    data=normal_events,
    target='Quality Check',
    event = outlier_events.iloc[0].to_dict(),
    control_method='median'
)

res.ranking[:5]

['H_28', 'C_08', 'H_30', 'A_06', 'A_02']
```



```
# Set the similarity distance to be higher to catch t
res = rca.get_root_causes(
    data=normal_events,
    target='Quality Check',
    event = outlier_events.iloc[0].to_dict(),
    control_method='median',
    similarity_distance=2
)

# We can view the scores for each node as
# Due to the sigmoid saturating for high v
pandas.DataFrame({
    'Pre-Sigmoid Score': pandas.Series(res),
    'Sigmoid Score': 1/(1+numpy.exp(-panda

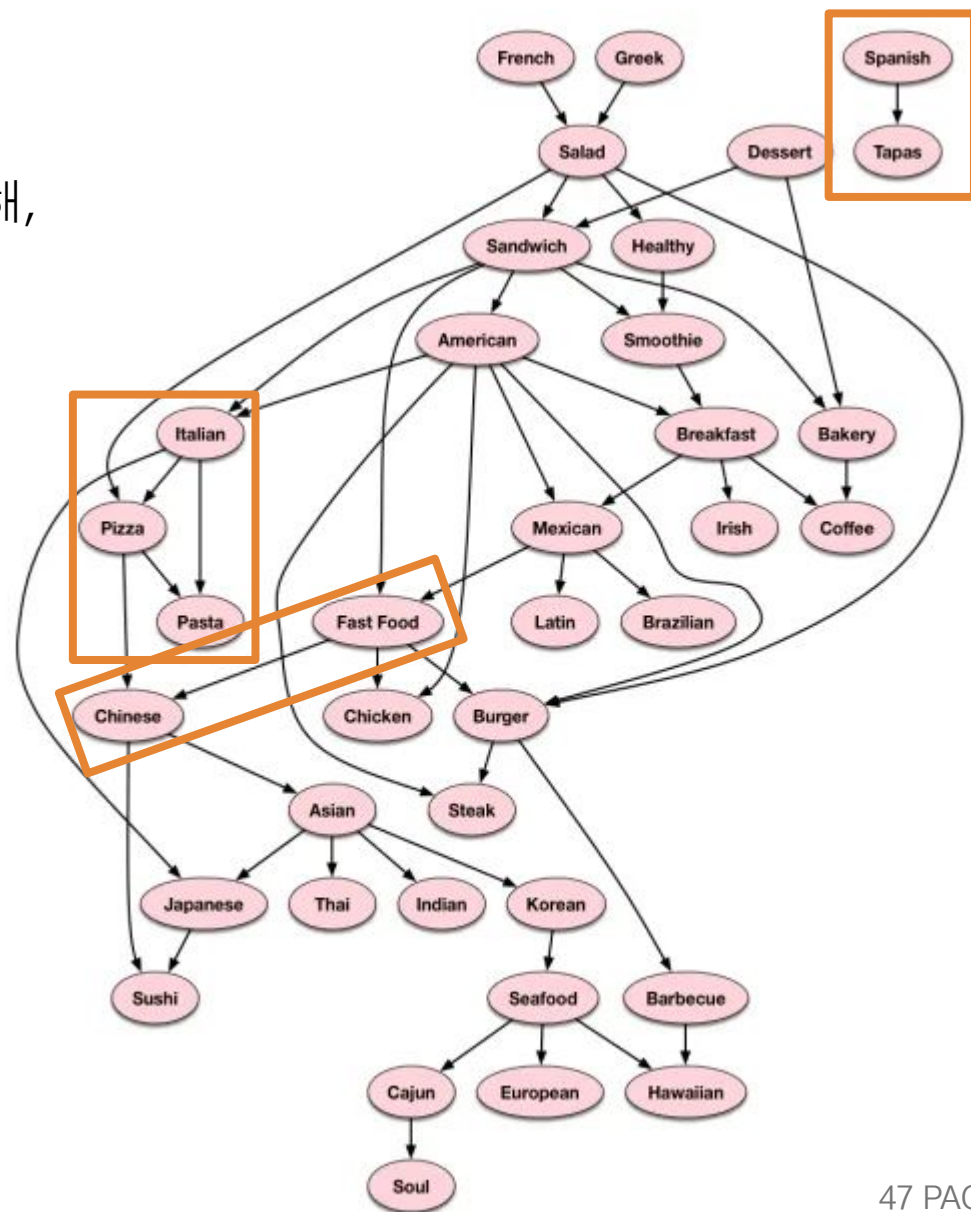
# We have now removed the impact of 'H_28' as it is
# 'H_30' has not been removed as its score is still b
res.ranking[:5]

['C_08', 'H_30', 'A_06', 'A_02', 'B_03']
```

	Pre-Sigmoid Score	Sigmoid Score
C_08	1.97	0.877611
H_30	6.19	0.997954
A_02	9.42	0.999919
A_06	9.42	0.999919
B_03	9.43	0.999920
B_06	9.43	0.999920
B_07	9.45	0.999921

(3) DoorDash의 레스토랑 개인화 추천

- **문제:** 배달앱 사용자에게 레스토랑을 개인화하여 추천하기!
- **접근법:** 사용자의 선호 요리를 기반으로 다른 요리를 추천하기 위해, Bayesian network structure learning 방법 이용
- **방법:** hill climbing approach : 엣지를 더하고/빼고/방향을 바꾸는 방법으로 greedy하게 최적 그래프(BN)를 찾아가는 방법
- **예시:** ‘스페인 요리’를 주문한 이력이 있을 때, 한 가지 유형의 요리(Tapas)를 주문할 확률을 계산
- **결과:**
 - 이탈리아 음식 → 피자나 파스타
 - 패스트푸드 → 중국 음식



Causal Discovery 라이브러리...

- <https://github.com/py-why/causal-learn>
- <https://www.bnlearn.com/>
- <http://www.dagitty.net/>
- <https://github.com/py-why/dowhy>
- <https://github.com/FenTechSolutions/CausalDiscoveryToolbox>

Reference

- [1] Pearl, J., Glymour, M., & Jewell, N. P. (2016). Causal inference in statistics: A primer. John Wiley & Sons.
- [2] Peters, J., Janzing, D., & Schölkopf, B. (2017). Elements of causal inference: foundations and learning algorithms (p. 288). The MIT Press.
- [3] Glymour, C., Zhang, K., & Spirtes, P. (2019). Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10, 524.
- [4] Spirtes, P., Glymour, C., & Scheines, R. (2001). Causation, prediction, and search. MIT press.
- [5] Koller, D., & Friedman, N. (2009). Probabilistic graphical models: principles and techniques. MIT press.
- [6] Vonk, M. C., Malekovic, N., Bäck, T., & Kononova, A. V. (2023). Disentangling causality: assumptions in causal discovery and inference. *Artificial Intelligence Review*, 56(9), 10613–10649.
- [7] De Lara, L. (2023). The difference between structural counterfactuals and potential outcomes. *arXiv preprint arXiv:2309.05997*.
- [8] Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov), 507–554.
- [9] Hoyer, P., Janzing, D., Mooij, J. M., Peters, J., & Schölkopf, B. (2008). Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21.
- [10] Brady Neal, Causal Inference course, <https://www.bradyneal.com/causal-inference-course>
- [11] Jiyong Park, Session 18–3 데이터 기반의 인과관계 발견 (Causal Discovery), Korea Summer Session on Causal Inference 2021, <https://www.youtube.com/playlist?list=PLKKkeayRo4PWYV8Gr-RcbWcis26ltlyMN>
- [12] <https://www.databricks.com/blog/optimizing-promotional-offers-using-causal-machine-learning>
- [13] <https://causalens.com/manufacturing-rca-for-data-scientists/>
- [14] <https://medium.com/@ryutayoshimatsu/causal-machine-learning-a3dc79205674>
- [15] https://medium.com/@cL_RnD/manufacturing-root-cause-analysis-for-data-scientists-with-causal-ai-ac8cce21ab57
- [16] <https://doordash.engineering/2016/12/21/how-to-get-from-salad-to-sushi-in-3-moves/>
- [17] <https://medium.com/@DoorDash/how-to-get-from-salad-to-sushi-in-3-moves-629ff161d88f>
- [18] Microsoft Research, Foundations of causal inference and its impacts on machine learning webinar, <https://www.youtube.com/watch?v=LALfQStONEc>
- [19] https://en.wikipedia.org/wiki/Hill_climbing

감사합니다
Q&A