DL連絡会輪読会

NTTコミュニケーションズ イノベーションセンター
藤原大悟

PC Algorithm と 因果の色々

参考文献：

“Causation, prediction, and search”, P Spirtes, C Glymour, R Scheines - 2000
“An Algorithm for Fast Recovery of Sparse Causal Graphs”, P Spirtes, C Glymour
「統計的因果探索」、清水昌平
「Pythonによる因果分析」、小川雄太郎
「Causal Inference In Statistics: A Primer」、Judea Pearl, et.al
その他

今回の目的

・因果探索(データから因果的構造を見つけるタスク)におけるデファクトスタンダード手法であるPCアルゴリズムを紹介

・ついでに、「因果って最近流行ってるけどなんなん？」と馴染みがなくてとっつきにくいと感じてる人にお気持ちや感覚的なところを持って帰ってもらう

因果とは？

(私が思うに)データの生成過程のモデリング方法の一つ

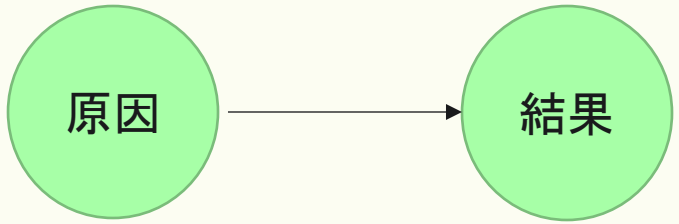
・確率分布でモデリング
・微分方程式でモデリング
・決定木やルールベースでモデリング

etc...

世の中にはさまざまなモデリングがある。どの考え方(近似)を採用するかは自由。

私たちには”因果(原因と結果)”みたいな概念や感覚がある、それを定式化したい

「Bが起こったのはAが原因である」

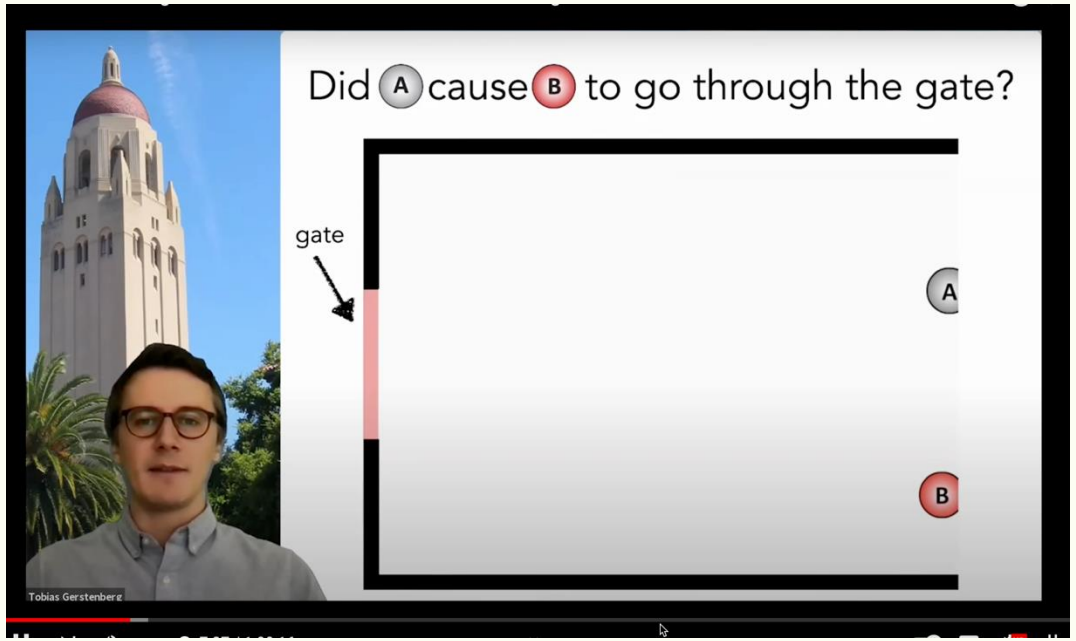


因果とは？：基礎概念

・**Counterfuctual (反事実)**

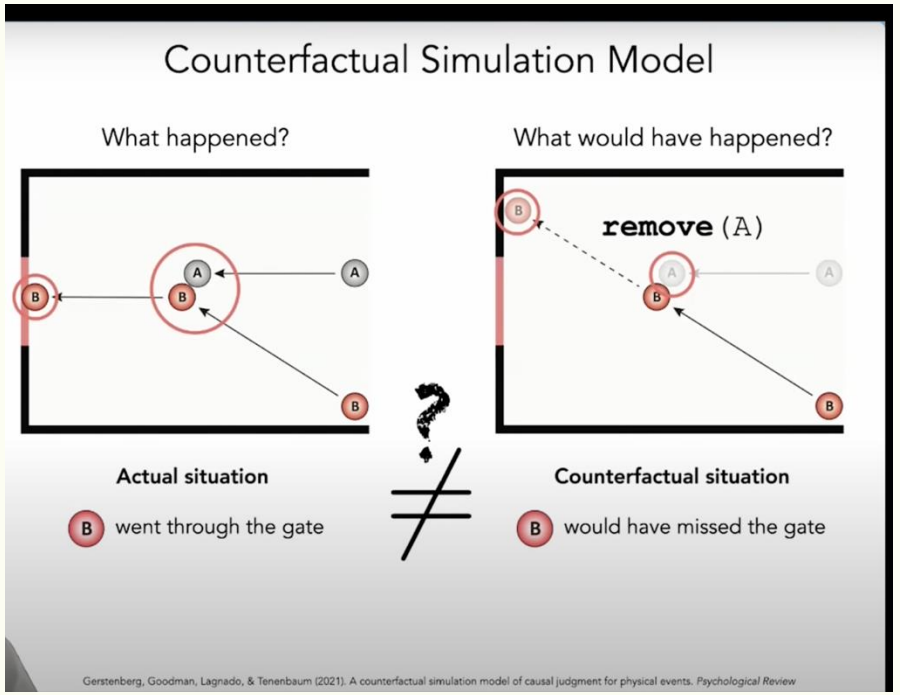
「もし、(実際にはあった)事象Aがなかったら」、「もし、変数Xがもっと小さかったら」

「因果とは何か？」を事実に反する「もし」で定義する



動画

From: [Tobias Gerstenberg: Counterfactual Simulation in Human Cognition]



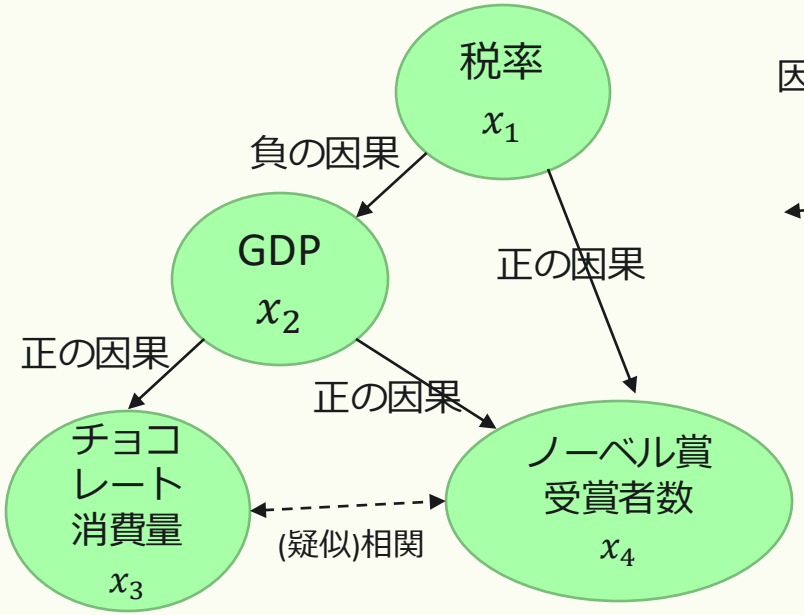
AがなければGoalしなかった
→ **AはGoalの原因**

(蛇足：もちろんこの現象は微分方程式でモデリングすることも可能、何を選択するかの問題)

因果とは？：基礎概念

・DAG (Directed Acyclic Graph)

仮定：
因果関係は**有向非巡回なグラフ**に埋め込み可能



因果構造が対応

・SCM (Structural Causal Model)

仮定：
各ノードの値は、**親変数と外乱を引数とする関数から生成**される

$$x_1 = f_1(e_1)$$
$$x_2 = f_2(x_1, e_2)$$
$$x_3 = f_3(x_2, e_3)$$
$$x_4 = f_4(x_1, x_2, e_4)$$

蛇足：SCMは(Rubin系の因果など)採用しない時もある

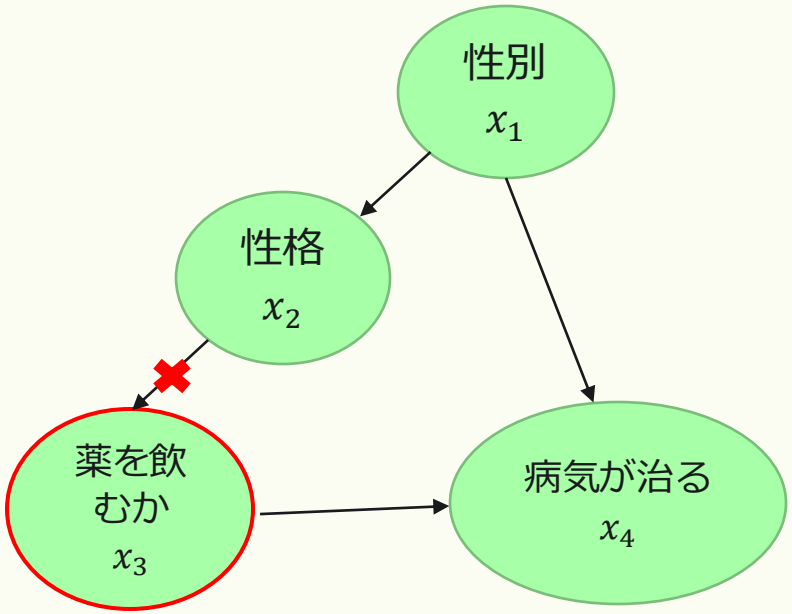
因果とは？：基礎概念

・**Intervention (介入)**

平たく言うと、恣意的にある変数を動かすこと。

本来確率的に値が決まる変数を、その関係を無視して動かすこと。

・薬を飲む群と飲まない群にランダムに振り分けて実験すれば、病気に効くかわかる
(A/Bテスト,RCTs, 対照実験)



・本来(自然な統計データだと)性格によって決まるところをそれに寄らず強制的に決める

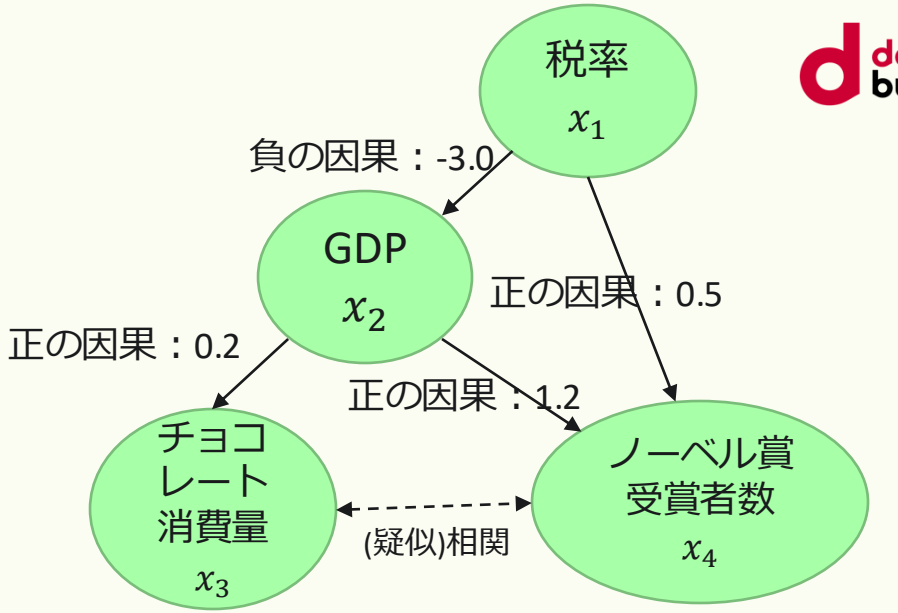
$do(x_3 = 1)$などと書く。
例：薬を介入で飲ませた上で病気が治る確率 $\mathrm{P}(x_4|do(x_3 = 1))$

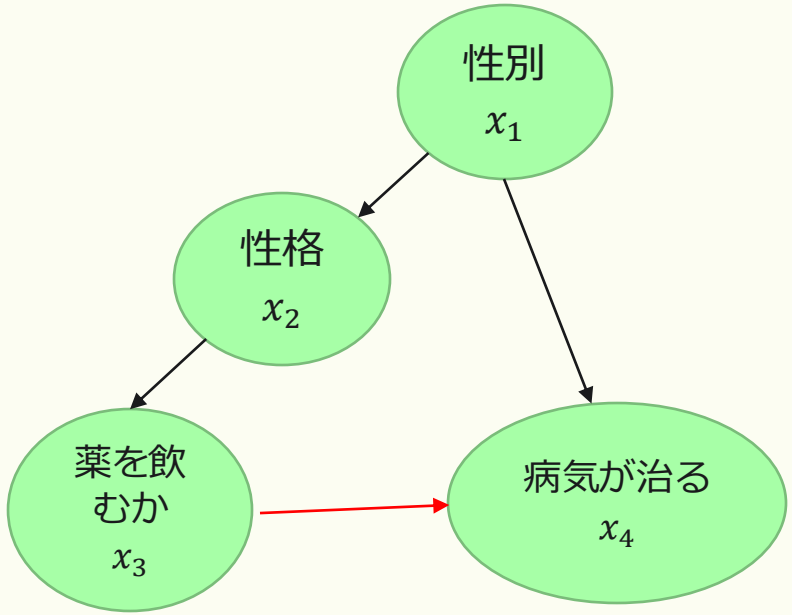
なぜ因果が重要なのか？1

・疑似相関：同時分布モデリングの危険性
政策「チョコレートを販売しよう！」は適切か？
チョコレート消費量とノーベル賞受賞者は共起する可能性が高いが因果ではない。

・介入効果推定：アクションの適切な評価
我々は税率を上げ下げしてどれだけノーベル賞受賞数が増えるのか知りたい。
薬を飲ませてどれくらいの確率で病気が治るのか知りたい。
これらはただの回帰や相関係数ではわからない。

・解釈性/最適化への利用：介入先、介入量の選定など
政策で「何に」「どれだけ」介入すればいいのか知りたい





ここが(ここだけが)知りたい！

なぜ因果が重要なのか?2：やりたいこと

一般論

システム

例：金融市場

データ

ドル

A社株

B国債

AI

ドルが高い！
→A社株買い
介入

何が起こる？？

AIによる予測 (同時分布モデリング)

A社株価を予測するモデル
ドルが高いときA社株も高い

実際のデータ
生成過程

予測モデル
の方向

A社株価
x_1

ドル
x_3

B国債値
x_2

円
x_4

$$x_1 = f_1(pa(x_1), e_1)$$
$$x_2 = f_2(pa(x_2), e_2)$$
$$x_3 = f_3(pa(x_3), e_3)$$
$$x_4 = f_4(pa(x_4), e_4)$$

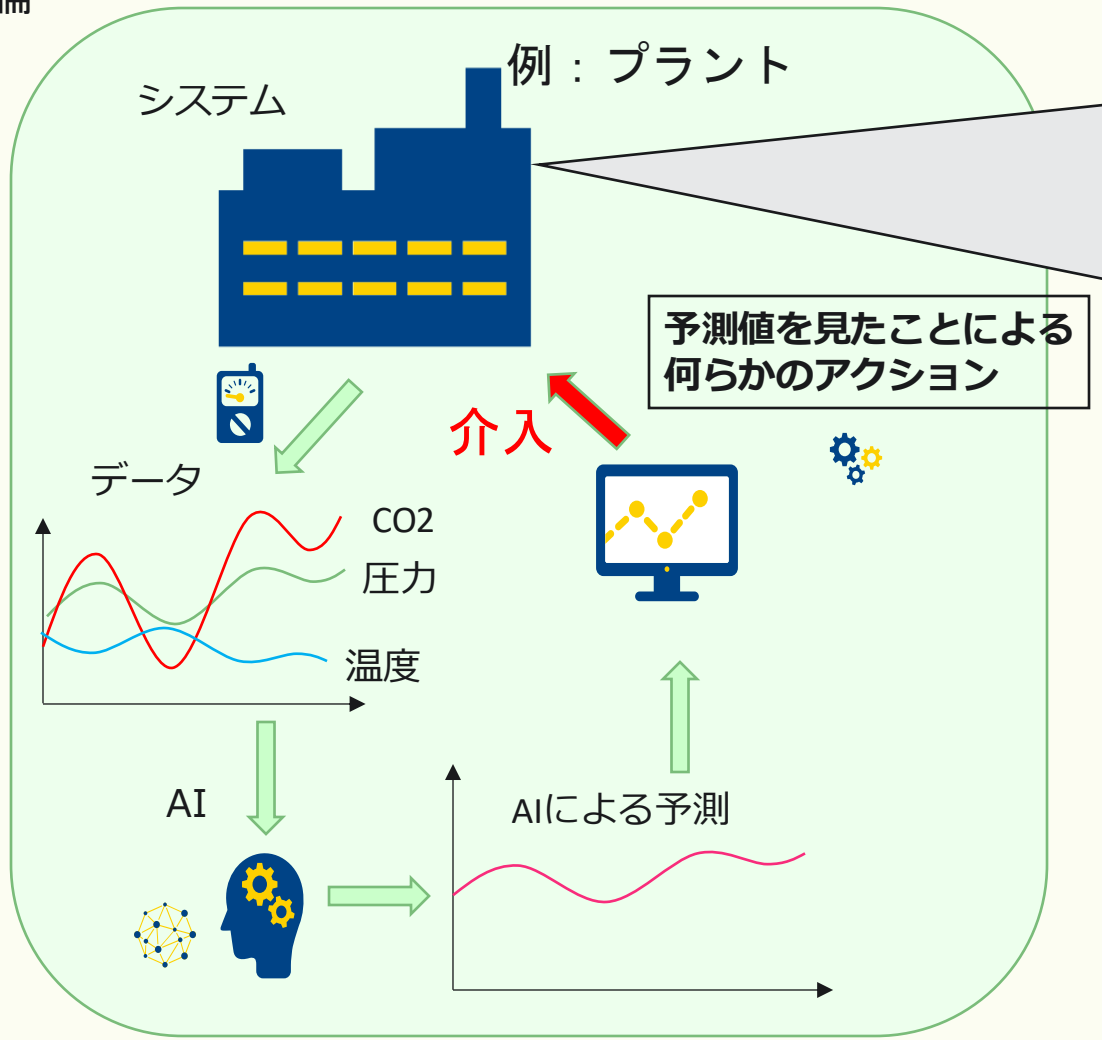
$pa(x_i)$ is set of parental variables of x_i

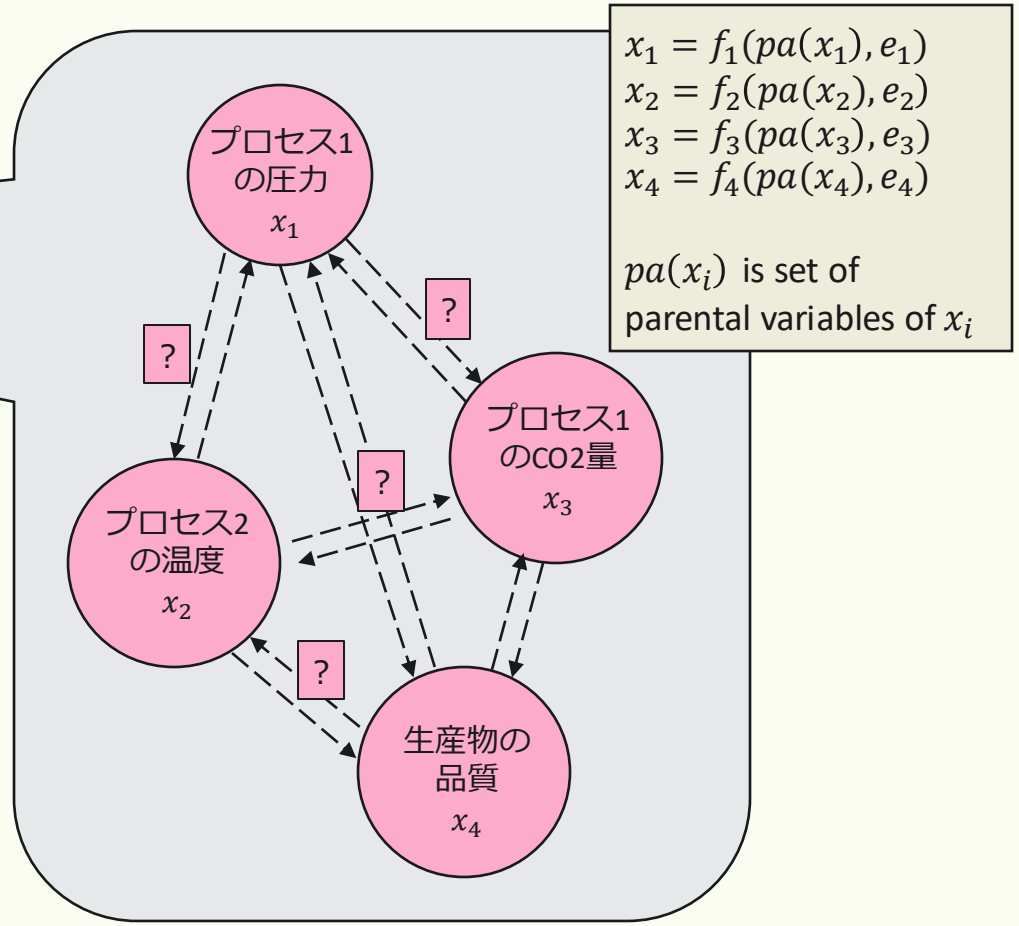
因果モデリング

(相対的に)システムに介入効果が無視できるときは気にしなくていい
→もし影響が無視できないほど市場が小さかったら？

なぜ因果が重要なのか?2：やりたいこと

一般論



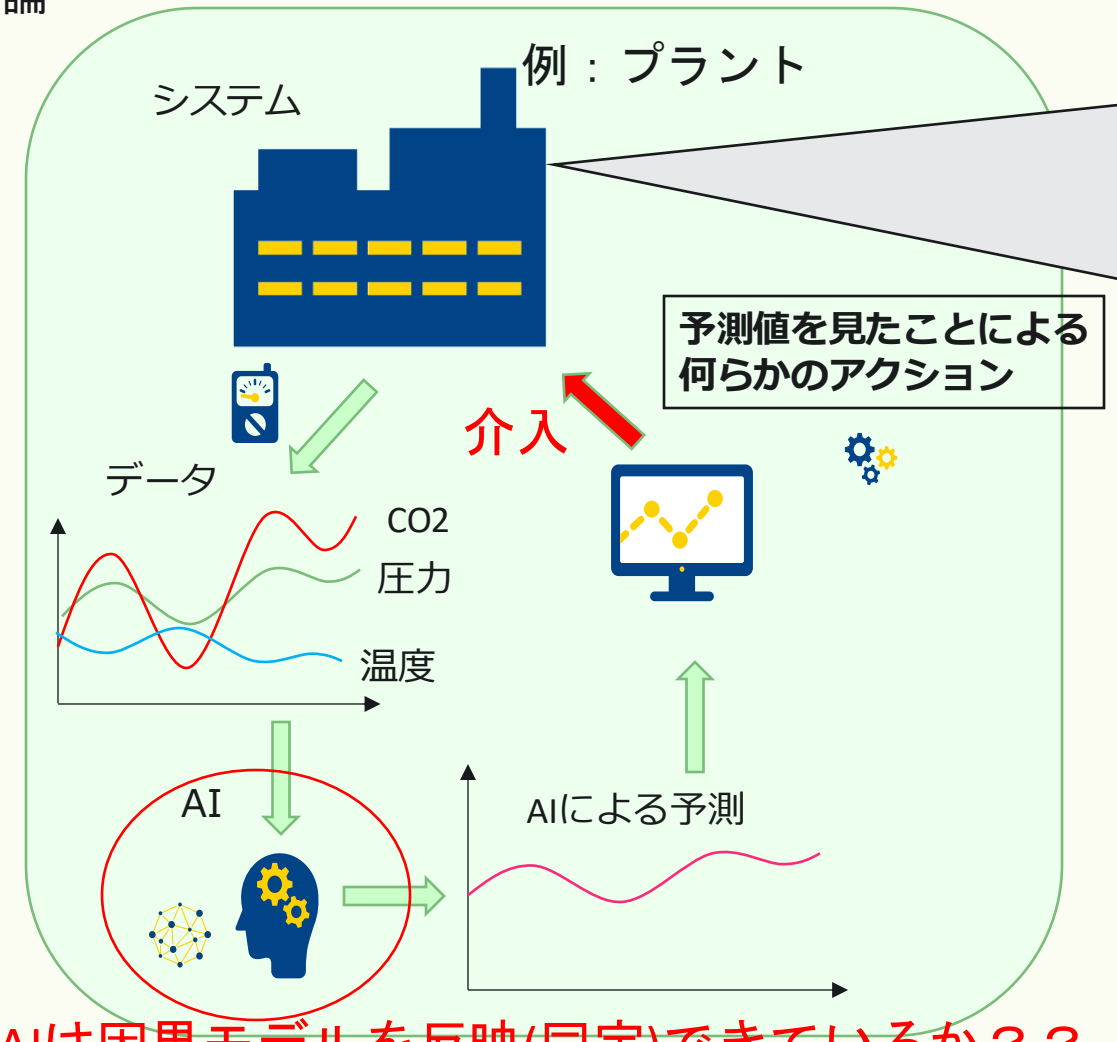


因果モデリング

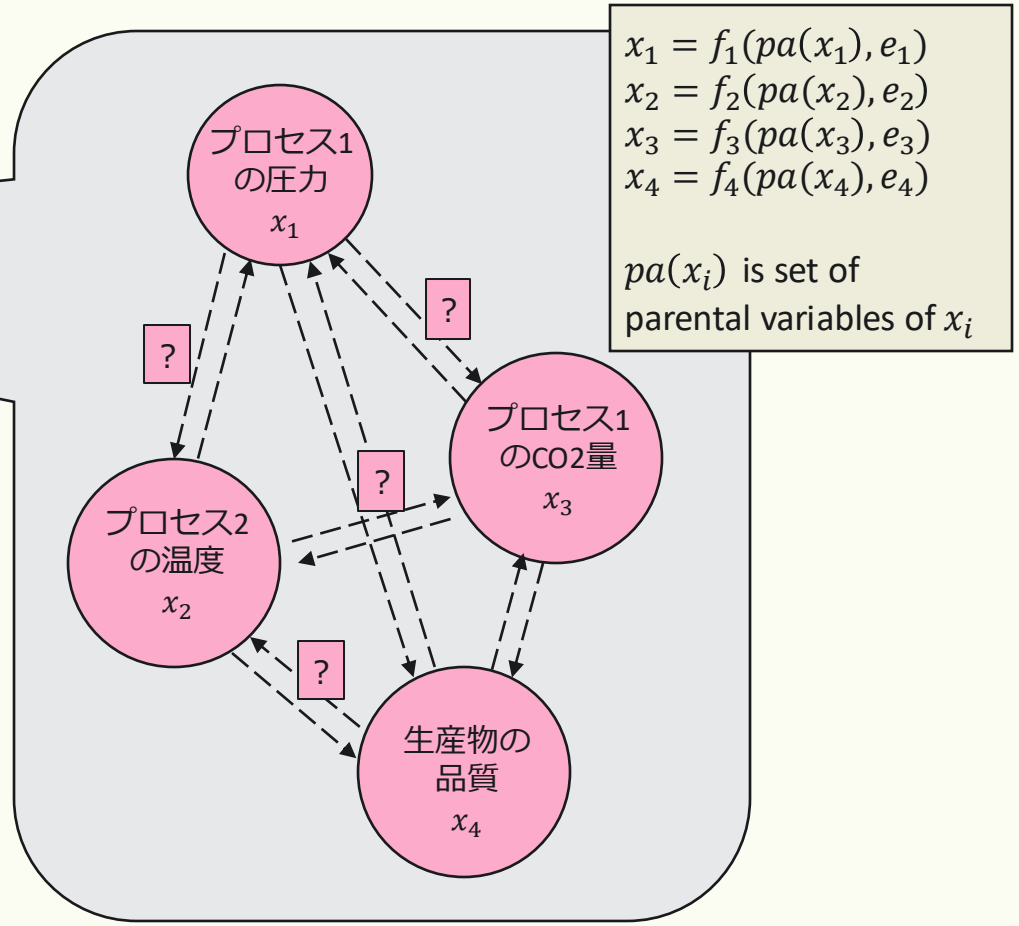
介入操作が入る場合、因果が重要

なぜ因果が重要なのか?2：やりたいこと

一般論



AIは因果モデルを反映(同定)できているか？？



因果モデリング

介入操作が入る場合、因果が重要

因果ロードマップ

要望：
・「Xを動かした介入効果が知りたい！」、「Yを改善する介入先/量)を知りたい！」とか

1.因果関係がわかっているとき

例：

時間的な前後関係「薬を飲む→病気が治る」、一般常識「親の年収→子供の学力」

a.実験できるとき

→ RCTs (対照実験)

b.実験できないとき(予算なし、倫理的課題)

→ 介入効果推定

(データから、統計や機械学習をもとに)

疑似相関、交絡など推定を邪魔する要素を適切に考慮

2.因果関係がわからないとき

a.実験できるとき

→ やはりRCTsなど

b.実験できないとき(予算なし、倫理的課題)

→ **因果探索 (因果関係の同定)**

(データから、統計や機械学習をもとに)

そのあとわかった因果で介入効果推定や介入先/量を選定

12

因果探索

大きく分けて3つのアプローチ

・Costraint-Based (制約ベース)アルゴリズム

PCアルゴリズム、FCI、CCDなど

データが満たす(条件付き)独立性から候補を絞る

・Score-Based (スコアベース)アルゴリズム

GESアルゴリズム、NOTEARSなど

適当な確率モデルを仮定

データに対する尤もらしさ(情報量基準など)からスコアを算出し候補を絞る

・セミパラメトリック

LiNGAMなど

外生ノイズの独立性を利用してICAや独立性検定問題に落とし込む

根本的な困難：

1.グラフの候補の多さ

(雑な見積もり)N変数で

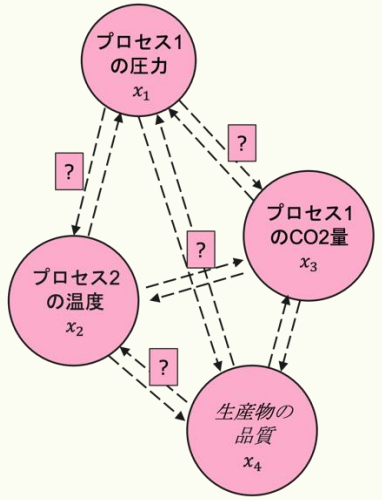
・$\frac{N(N-1)}{2}$ 個のエッジ候補

・矢印の向きが2つとエッジなしの3通り

計 $3^{\frac{N(N-1)}{2}}$ 通り

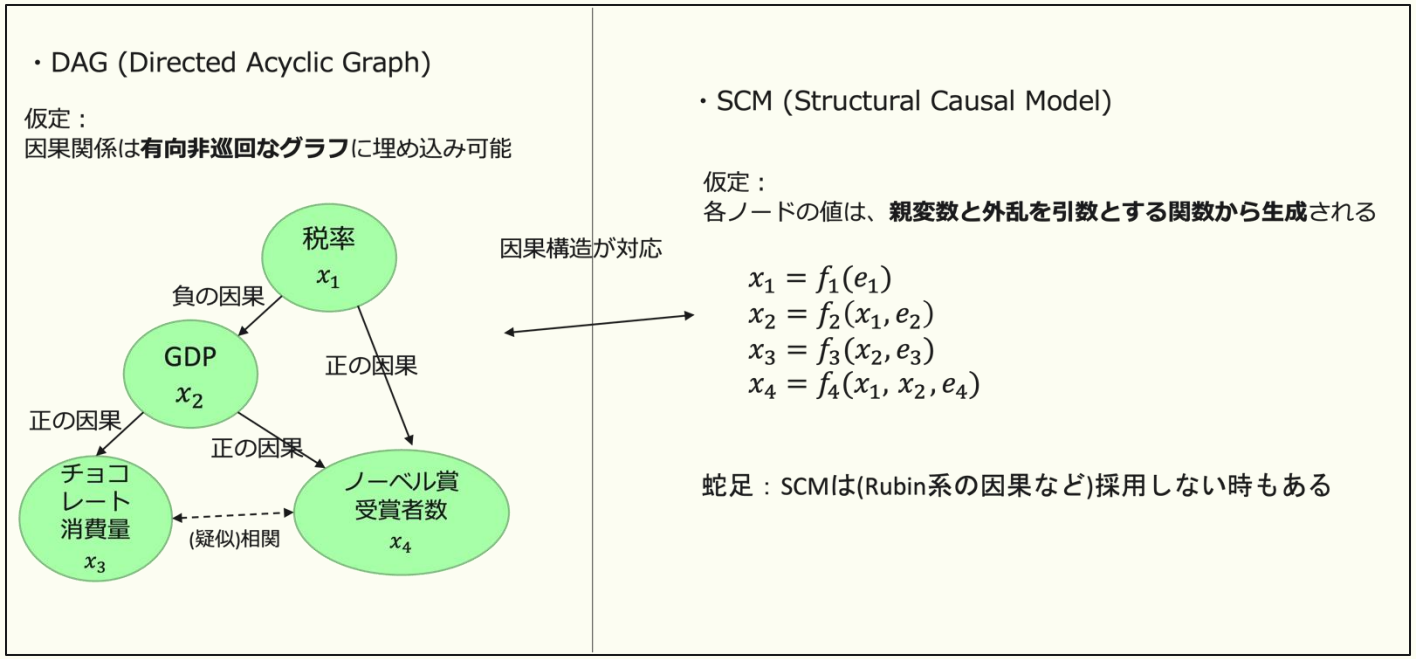
2.独立性の判定困難さ

カイ二乗検定やHSIC、相互情報量、
相関(線形ガウスの場合のみ)など
帰無仮説の積極肯定(誤用)
計算量も多い



PCアルゴリズム：ベイジアンネットと因果

すごくベイジアンネットと似ている



違いは？

・ベイジアンネット：
条件付き独立性を記述

・因果グラフ：
データの生成過程を記述 (→その結果条件付き独立性も導かれる)

二つにはちょっと齟齬がある時がある

ポイント：Constraint-Based アルゴリズムでは条件付き独立性の情報のみを利用
→実際に絞れるのは ベイジアンネット

PCアルゴリズム：ベイジアンネットと因果

忠実性：(雑に言うと)ベイジアンネットと因果グラフが一致するという仮定

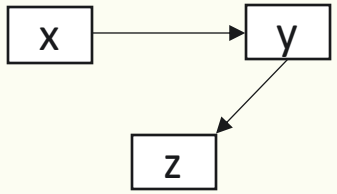
24

追加の仮定：忠実性

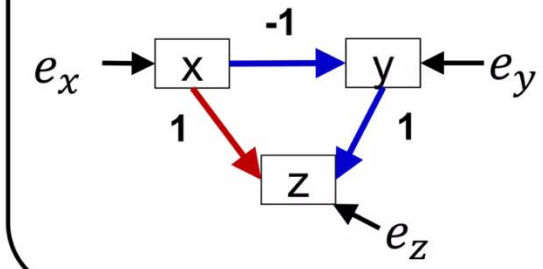
(Spirtes et al., 1993; Pearl, 2000)

- 「観測変数間の独立性・条件付き独立性の有無は、グラフ構造のみによって決まる」
- 特定のパラメータ値には依存しない

このグラフで成り立つ条件付き独立性を表すベイジアンネット



忠実性が崩れている例(e_i がガウスと仮定)

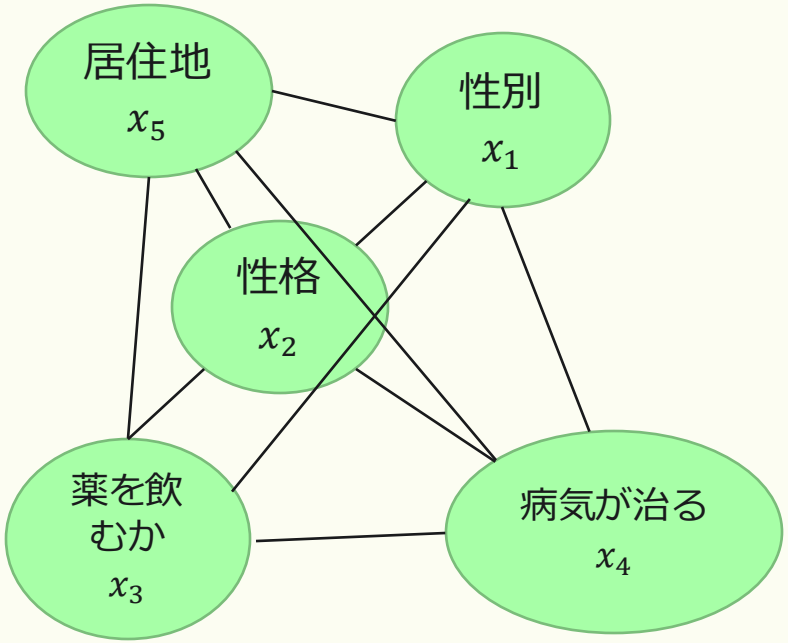


$$x = e_x$$
$$y = -x + e_y$$
$$z = x + y + e_z$$

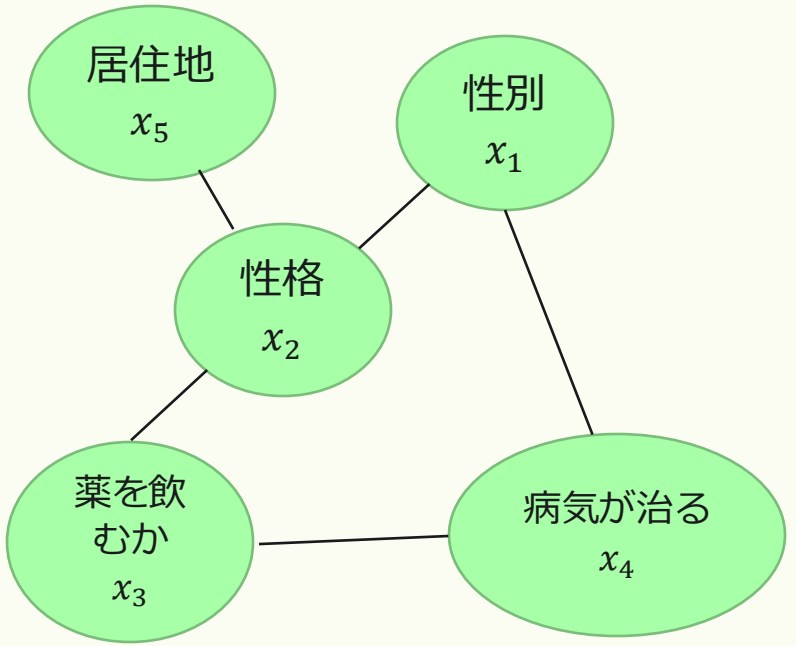
$$\mathrm{cov}(x, z) = 0$$

スライド From: 因果探索: 基本から最近の発展までを概説 (清水昌平)

PCアルゴリズム：基本アイデア

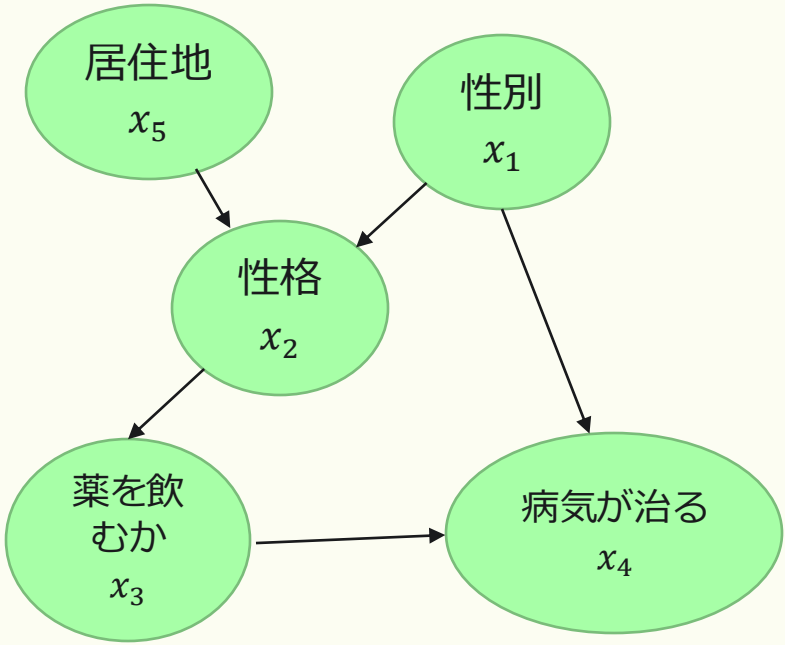


1.無向完全グラフから始める



2.データから読み取れる(条件付き)独立性を満たすようにエッジを取り除く

データから独立性を求める方法：
カイ二乗検定やHSIC、相互情報量、
相関(線形ガウスの場合のみ)など利用



3.決まったルールに沿って矢印を方向づける

DAGであるという条件などから
[Verma92,Meek 95],

PCアルゴリズム：のここがすごい

PCアルゴリズムの先行手法、SGSアルゴリズムとICアルゴリズムは同じような手順だが、
ある二頂点に対して他の全頂点(のある部分集合)で条件付けした独立性を虱潰しに検定するものだった。
(愚直)

先行手法の Pros

・確実。条件付き独立が成り立つような集合があるならエッジはない。

先行手法の Cons

・指数的に検討パターンが増える。(2^N)

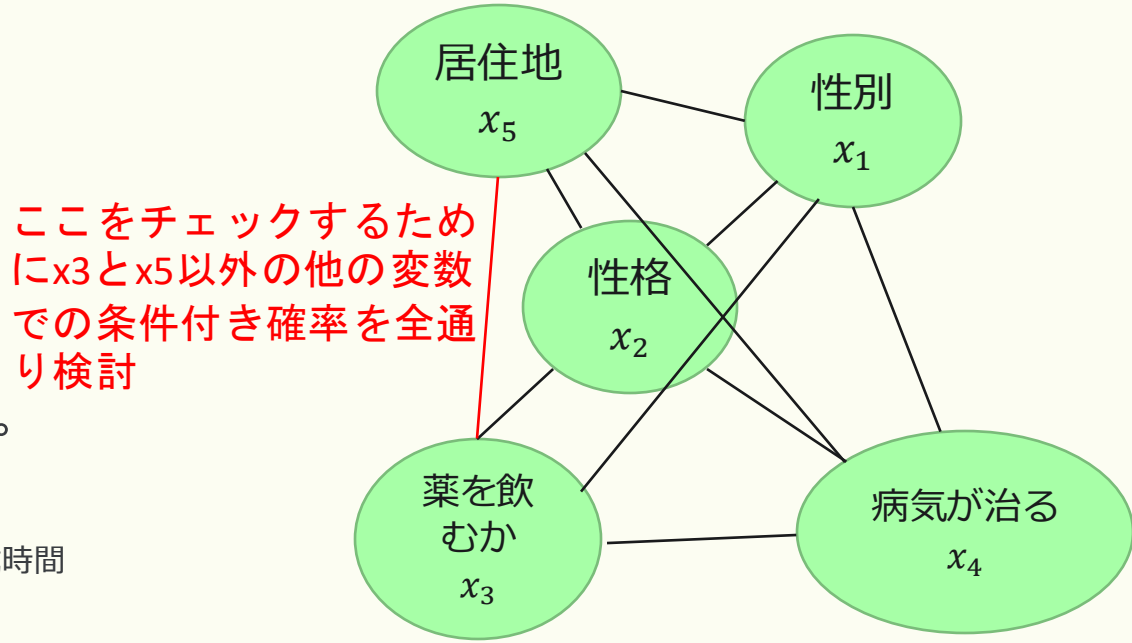
・多変数条件付き独立性のサンプル計算は数値的に不安定

PCアルゴリズムは、

条件付ける変数を一つずつ変数を増やして独立性を検証する。
それによりスパースなグラフに対して特に早く終わる。

※ノード数N, 隣接ノード数kとすると最悪計算量　$N^2(k+1)(N-2)^k \,/\, k!$ ←Nに対して多項式時間

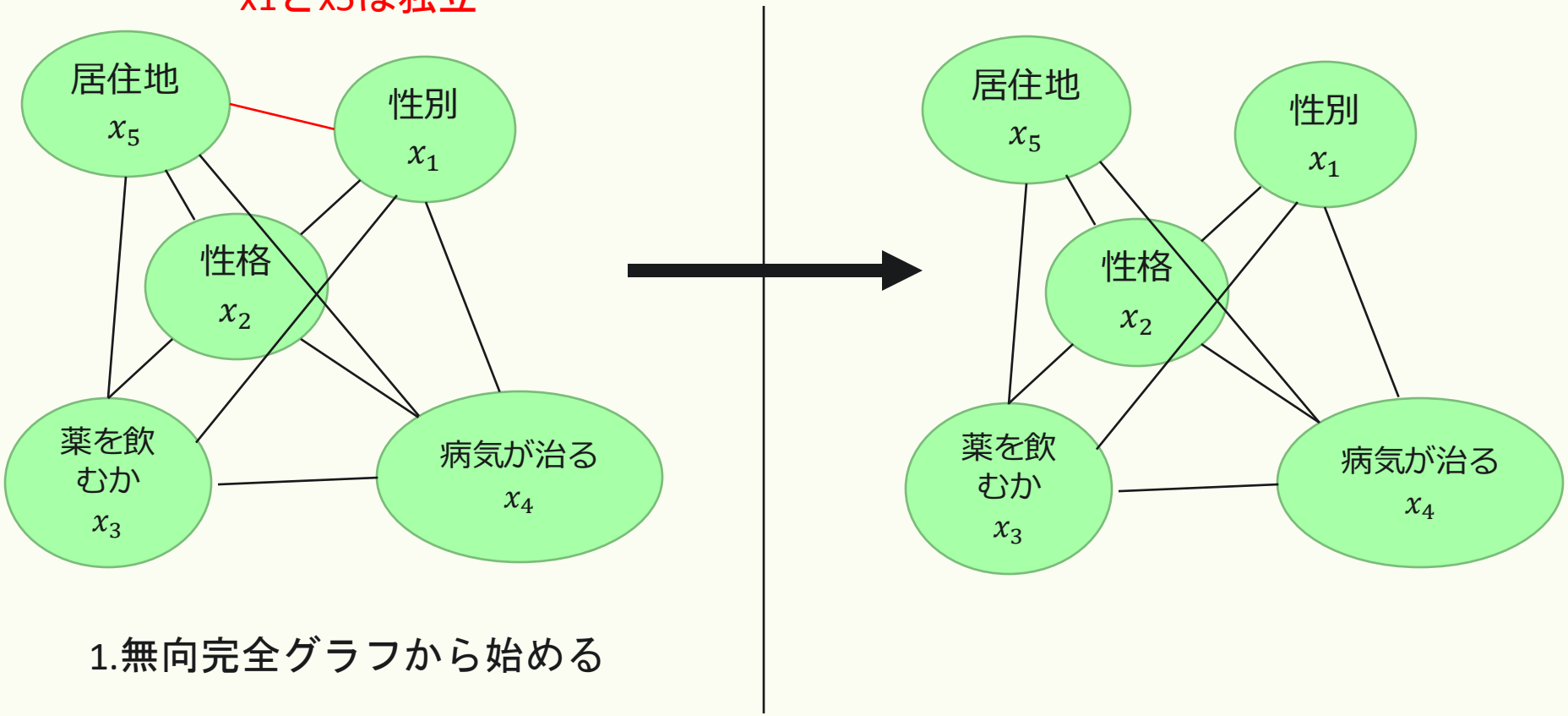
※結果がSGS/ICアルゴリズムと一致することを証明



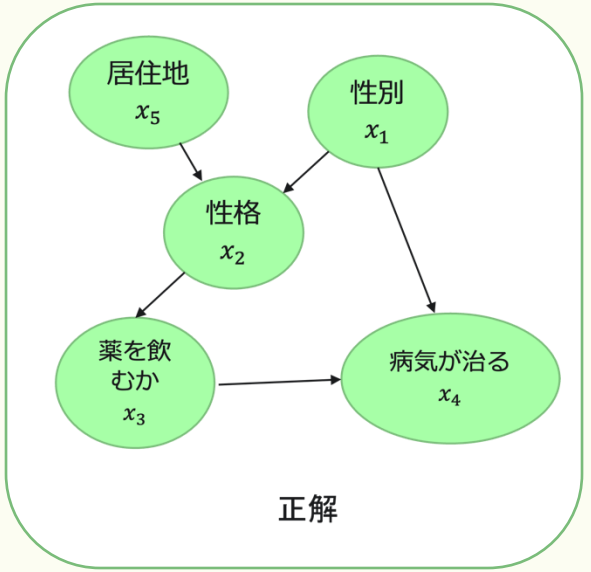
PCアルゴリズム：実際にやってみよう

0次の独立性検定
for i ∈ [1,N]
　for j ∈ {iに隣接するノード}
　　if $X_i \perp X_j | \{\}$, remove the edge

x1とx5は独立



1.無向完全グラフから始める

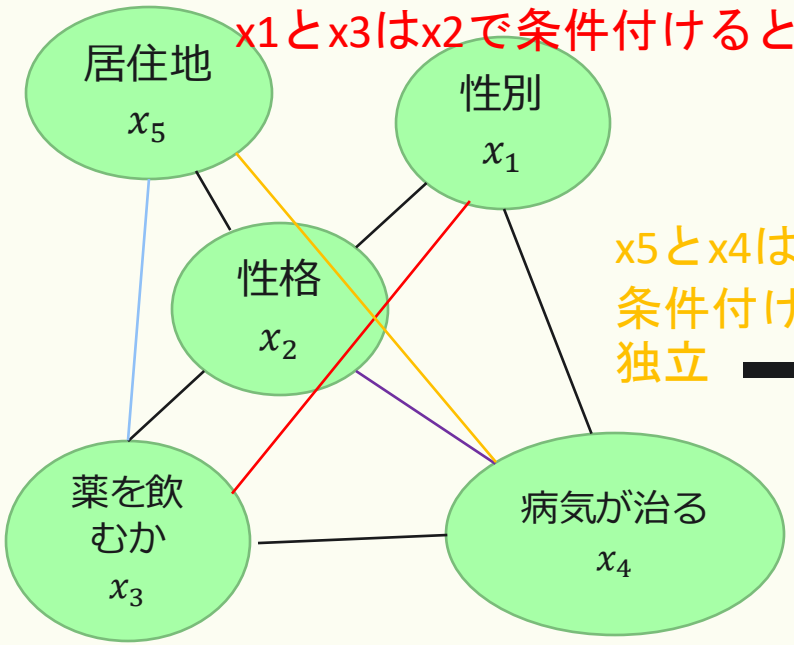


PCアルゴリズム：実際にやってみよう

1次の独立性検定

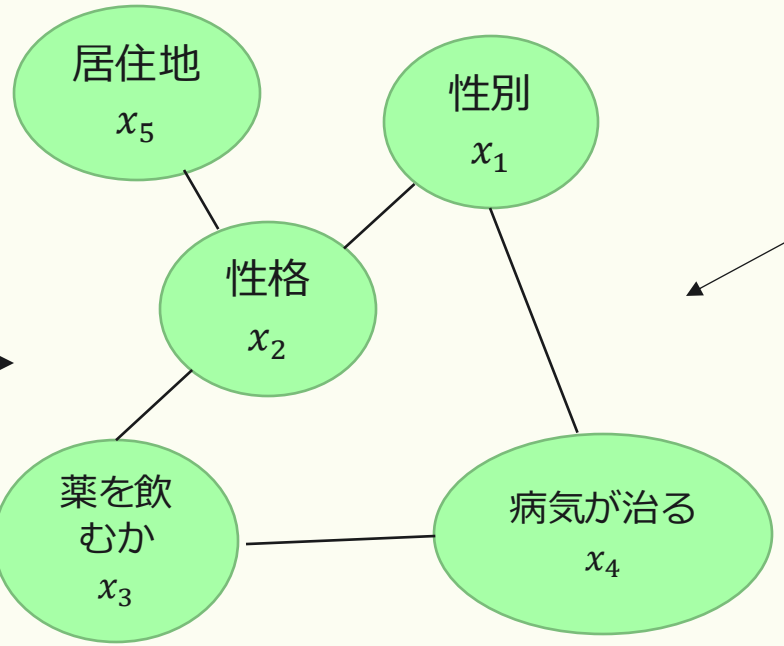
for i ∈ [1,N]
　for j ∈ {iに隣接するノード}
　　for k ∈{j以外のiに隣接するノード}
　　　if $X_i \perp X_j | \{X_k\}$, remove the edge

x5とx3はx2で条件付けると独立

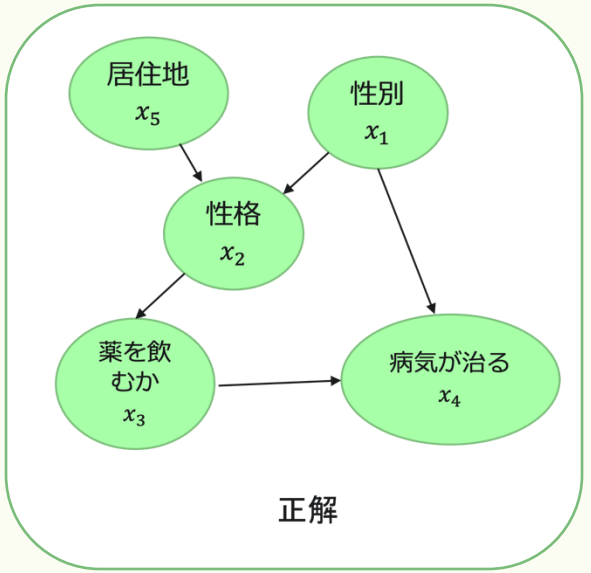


x1とx3はx2で条件付けると独立

x5とx4はx2で条件付けると独立



x2とx4はx3で条件付けると独立

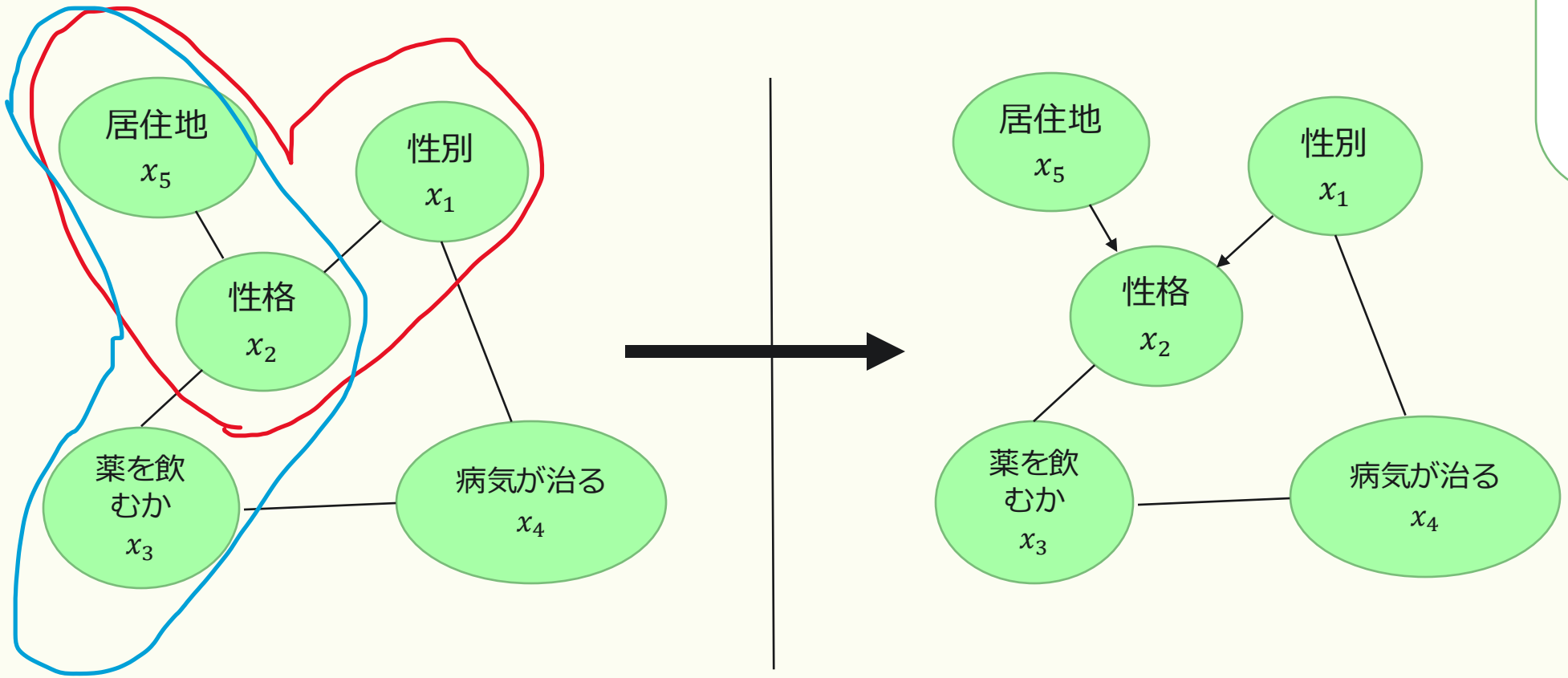


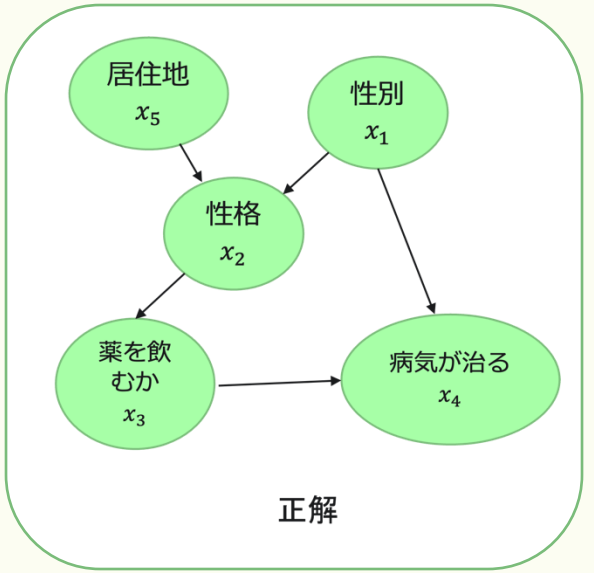
骨格(skelton)が一致した

場合によっては2次、3次以上の独立性検定 $X_i \perp X_j | \{X_k, X_l\}$ が必要である(残っている最大隣接エッジ数で判断)

PCアルゴリズム：実際にやってみよう

ルールによる矢印つけ1
V字構造 $X_i - X_k - X_j$について、
if **not** $X_i \perp X_j | \{X_k\}$, direct like $X_i \rightarrow X_k \leftarrow X_j$





Collider (合流点)で条件付けるとセレクションバイアスが乗って疑似相関が生じるのを利用

PCアルゴリズム：実際にやってみよう

ルールによる矢印つけ2

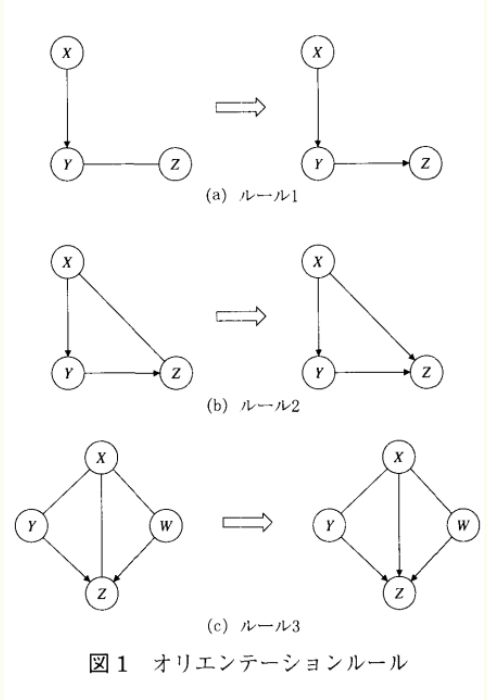
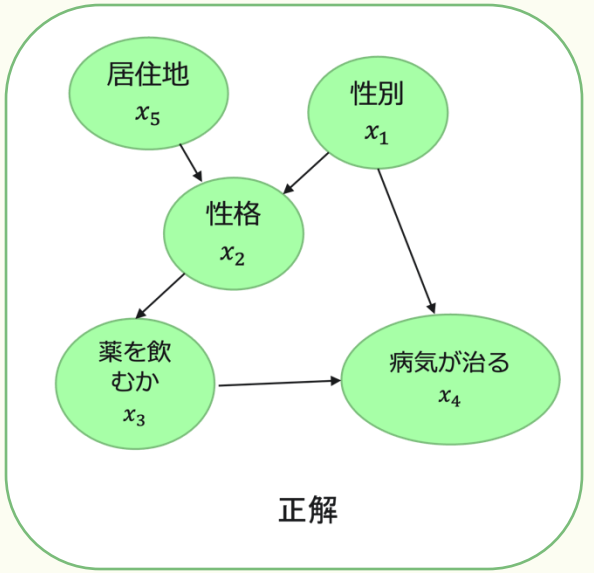
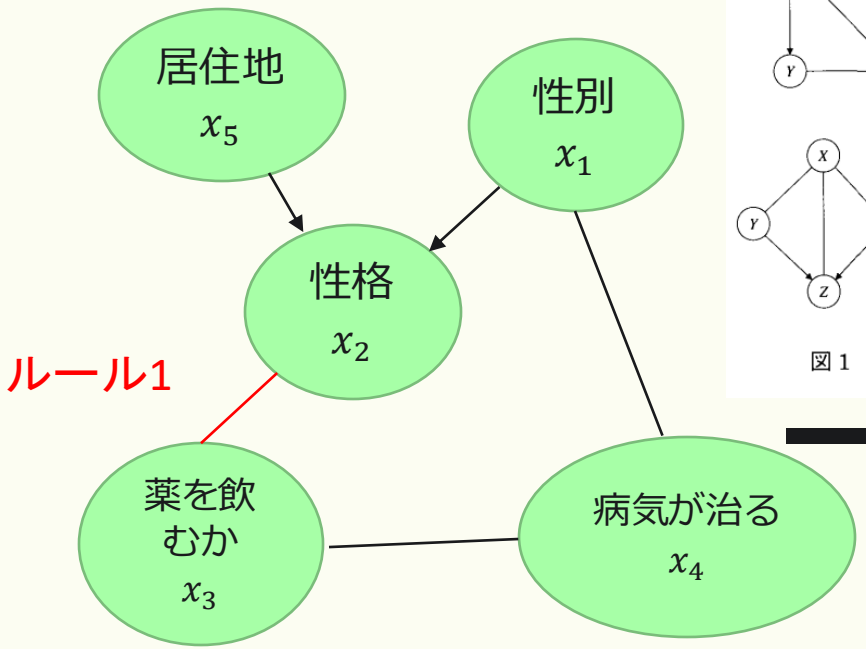
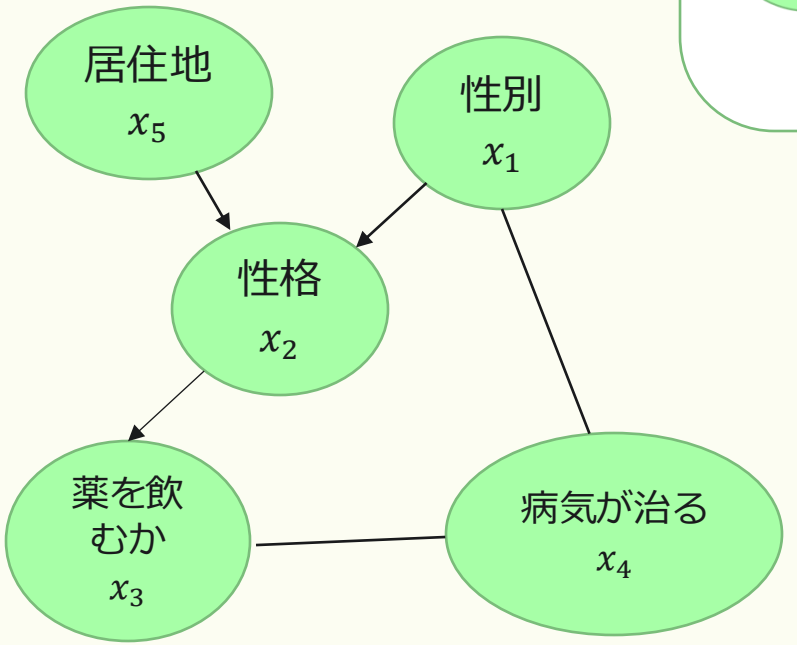


図1　オリエンテーションルール





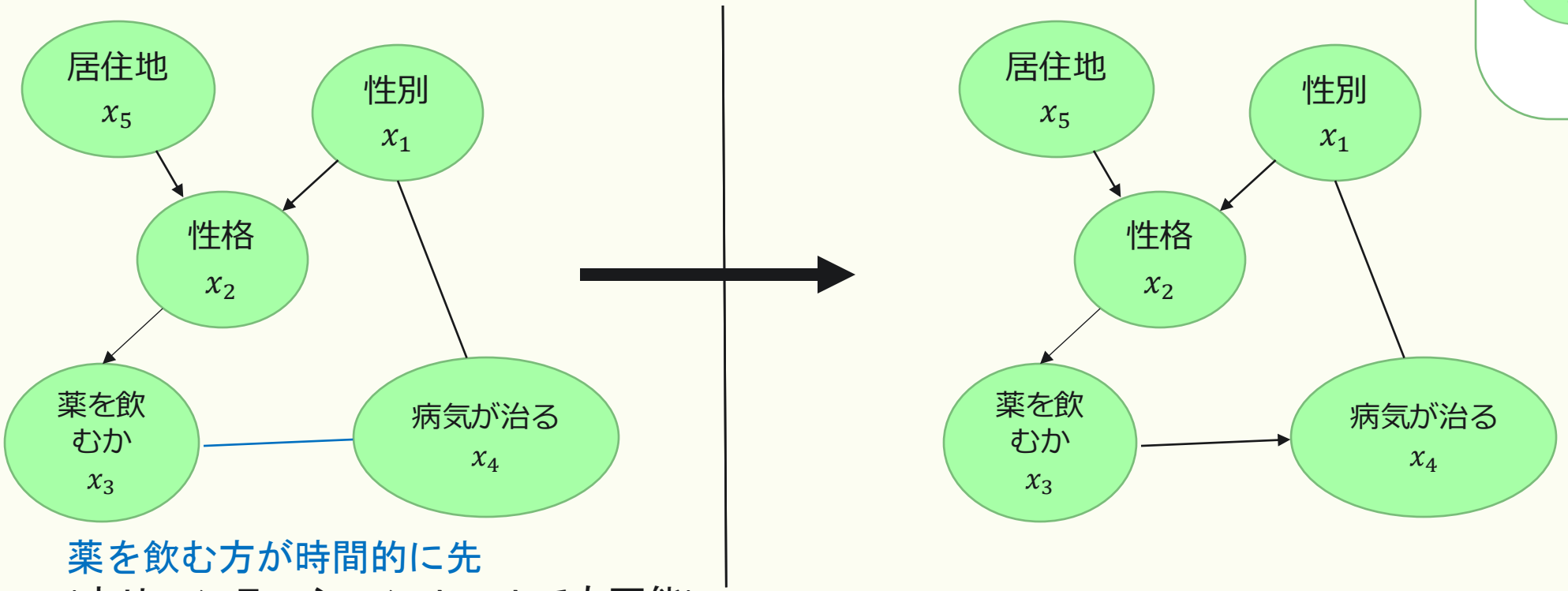
ルール1

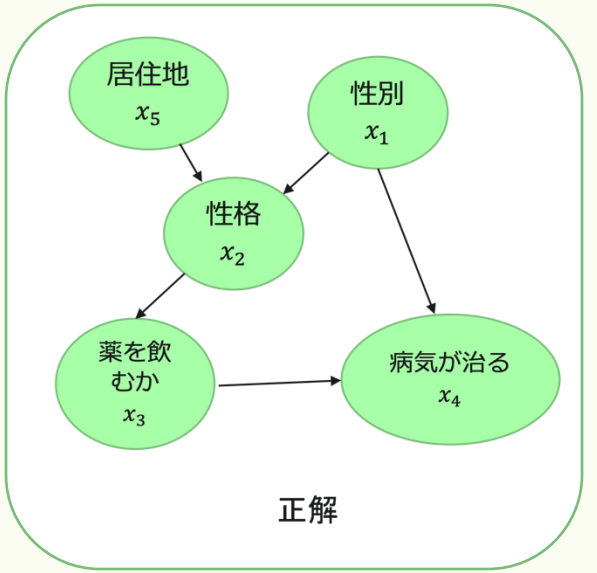


※逆だとしたら条件付き独立性が成り立ってないとおかしい

PCアルゴリズム：実際にやってみよう

ルールによる矢印つけ3
時間的な前後関係、事前知識の利用

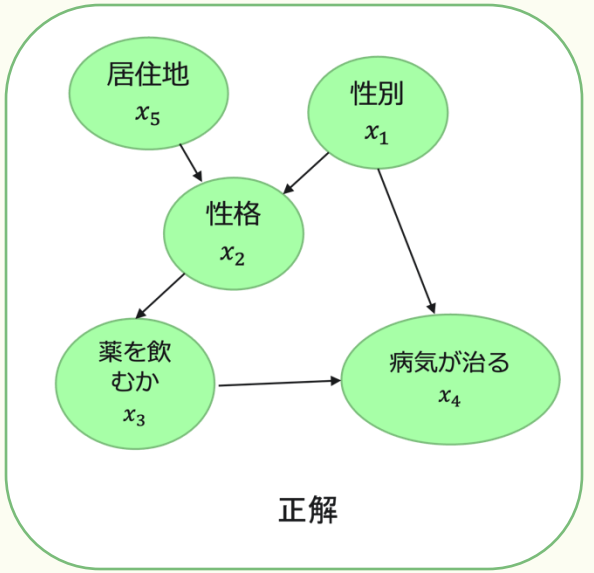


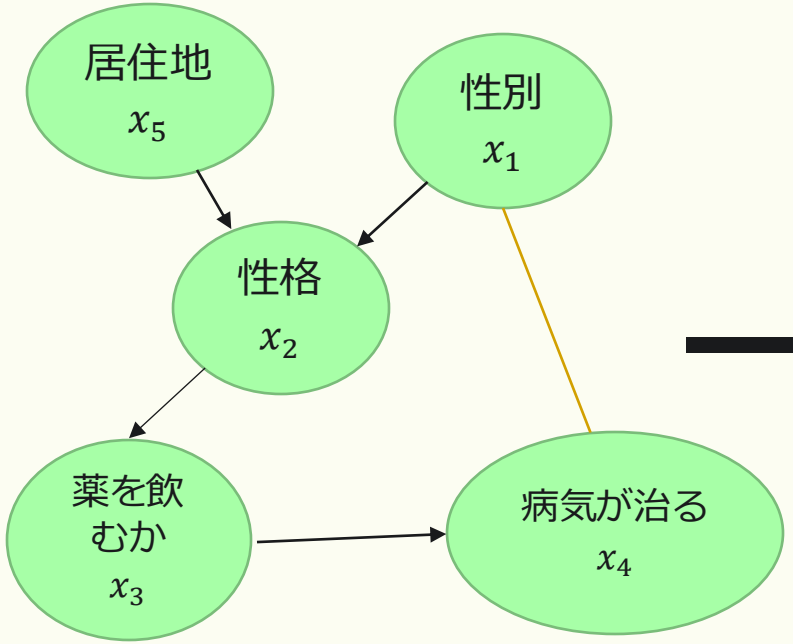


薬を飲む方が時間的に先
(オリエンテーションルールでも可能)

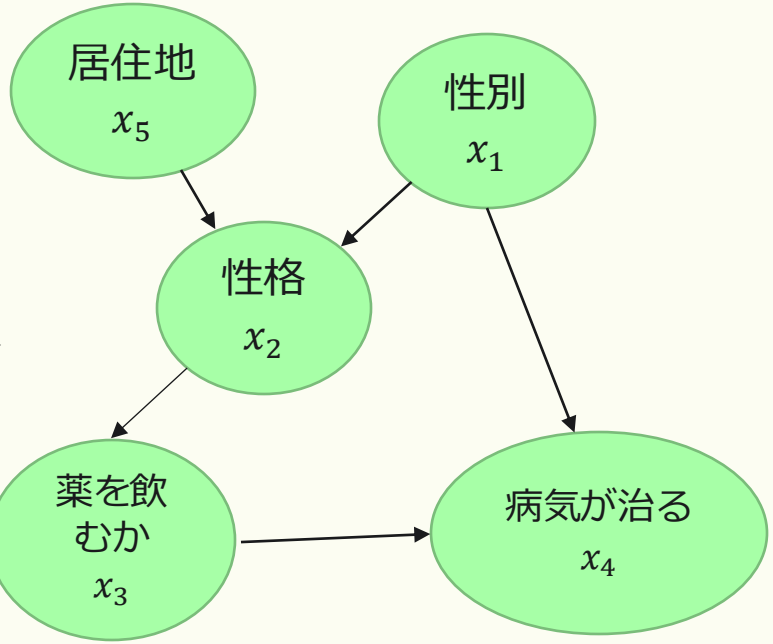
PCアルゴリズム：実際にやってみよう

ルールによる矢印つけ4
DAG制約の利用など





DAG制約から決まる
逆だとしたら巡回ができる



これだけやっても全部は向きが決まらないこともある

PCアルゴリズム：評価

・37の変数と46の辺を持つ疎なグラフ-ALARMネットワーク（Beinlich, Suermondt, Chavez, & Cooper, 1989）

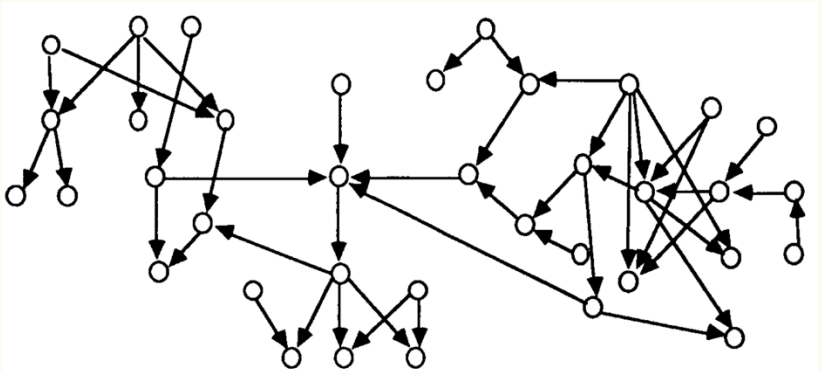


Figure 1 ALARM network

従来手法1：因果順序が与えられた有利な手法
・サンプル数10000
・Macintosh II で22時間ほど
・エッジFN2つとFP2つ

従来手法2：SGSアルゴリズム
・20MBのRAMを搭載したDECワークステーション
・約17個の変数でメモリオーバーフロー
・そもそも扱えない

提案手法：PCアルゴリズム
・サンプル数20000
・DECstation 3100で10秒ほど
(Macintosh IIより遥かに高速とのこと)
・精度は右の表

Table 1

	Omitted undirected edges	False undirected edges	Orientation errors
Trial 1	3	0	5
Trial 2	3	0	6
Trial 3	3	2	7

PCアルゴリズム：評価

いい点：

・高速でシンプルなアルゴリズム
・(比較的)仮定が少ない。非線形の因果性にも対応(ただし独立性を非線形対応させる必要がある)
・離散、連続変数ともに独立性さえ測れれば使える
・少なくとも、それまでのSGSやICアルゴリズムよりは飛躍的に速い

一般的な批判点：

・独立性検定の多くは帰無仮説が「独立」。が本アルゴリズムは棄却できないとき独立と積極肯定している
・密結合ネットワークに対してはいまだに指数的な計算量を持ち遅い
・独立性検定アルゴリズム自体が計算量が高い傾向にある
・一部分の無向エッジが残る場合がある。その点ではLiNGAMなどの方が優れている。
・忠実性の仮定が必要

まとめ

・因果について、基礎的な事項を直感的に説明してみた
・因果について、大まかなロードマップを書いてみた
・その中で因果探索について、現在でもベースラインアルゴリズムとして生き残っているデファクトスタンダードな手法、PCアルゴリズムを学んだ