Amortized Inference for Causal Structure Learning

因果構造学習のための償却推論

内田研究室 M1
中本 一輝

ULab Uchida

読んだ理由

- 因果探索が気になっていたから
- NeurIPSの新しい論文
- （ちょっと関連がある）VAEの論文を読んだことがあった

研究概要

- 事後分布を近似する方法である償却推論を因果探索と組み合わせた
- 償却推論で用いるNNを、因果探索の問題設定に合わせて構成している
- シミュレーションで作成したデータセットで有効性を検証

背景：因果探索

例：チョコレートとノーベル賞

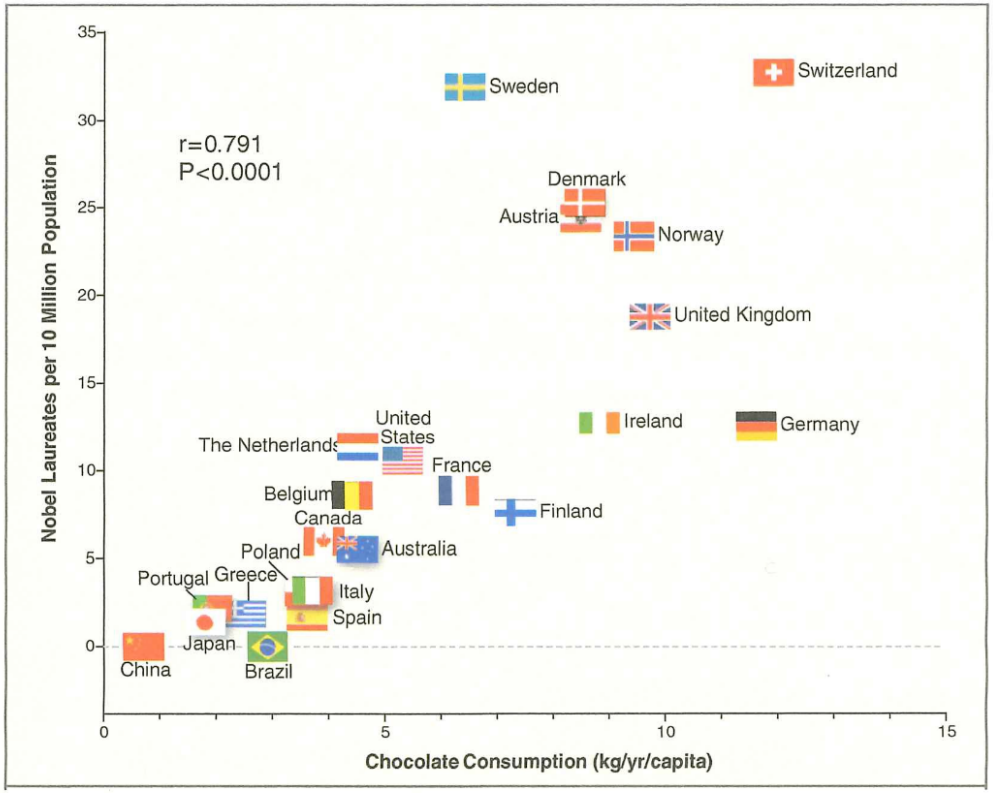


図 1.1　チョコレートの消費量とノーベル賞の受賞者数の散布図．出典：Franz H. Messerli, Chocolate Consumption, Cognitive Function, and Nobel Laureates. *New England Journal of Medicine* (367), 1563, Figure 1, Massachusetts Medical Society, 2012.

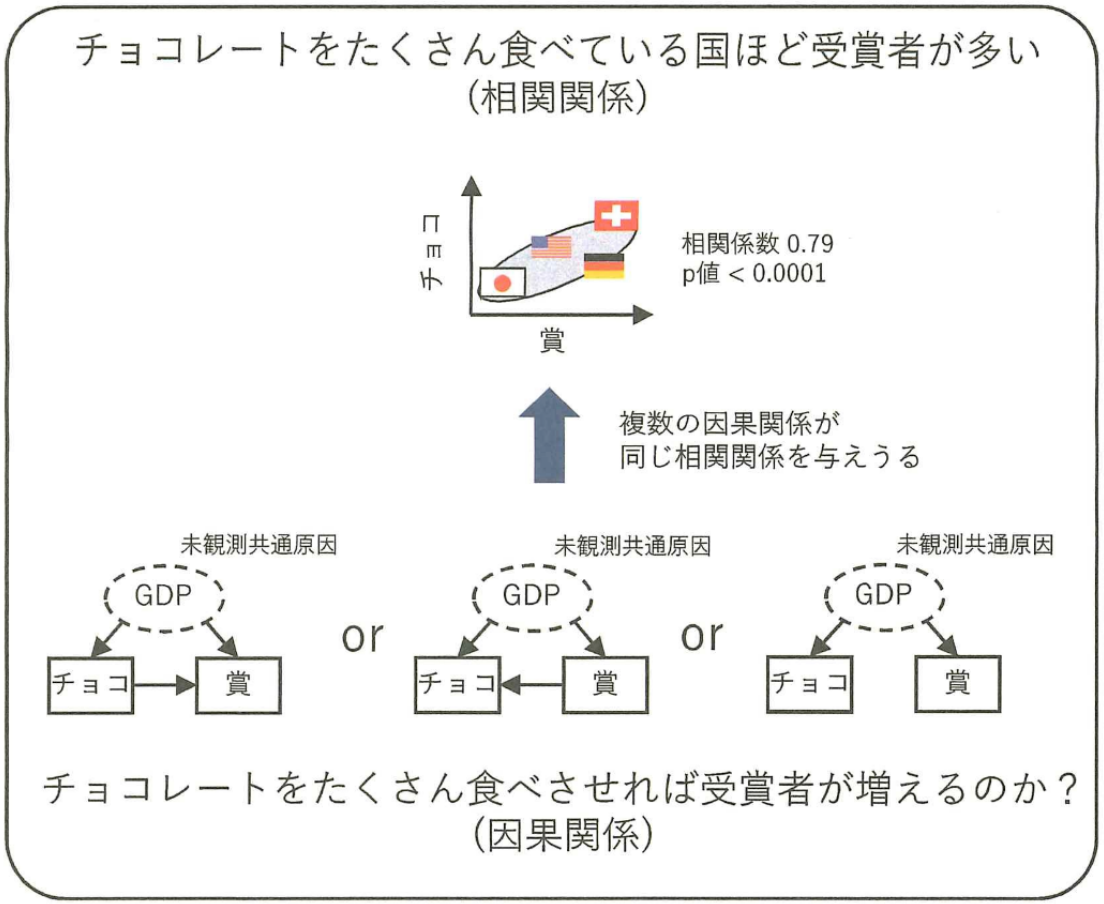
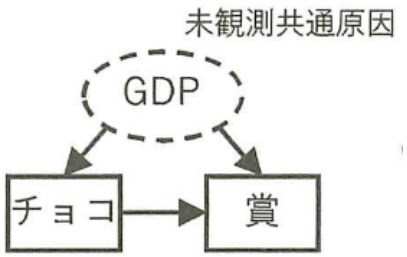


図 1.2　複数の因果関係が同じ相関関係を与える可能性があります.

因果グラフ



- **観測変数（observed variable）**
  データが収集される変数。四角で囲まれる。
- **未観測変数（unobserved variable）**
  データが収集されない変数。点線の楕円で囲まれる。

矢印の始点が原因の変数で、終点が結果の変数となっている。
このような定性的な因果関係を表す図を、**因果グラフ（causal graph）**という。
※因果効果の大きさなどの定量的な情報は含まれない。

では、そもそもどのようなときに因果関係があるといえるのか？

→ 反事実モデルによって定義

反事実モデル（個体レベルの因果）

個体Aはある病気にかかっている。我々は、ある薬が個体Aの病気を治すかどうか知りたい。
→ 個体Aの2つの行動の結果を比較する。

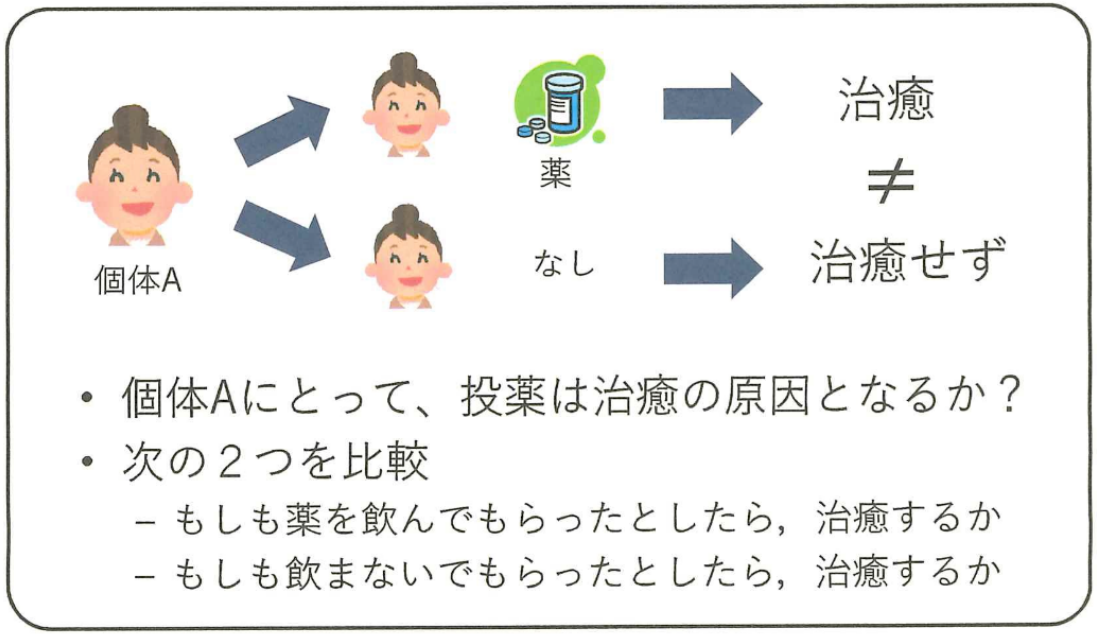


図 2.1　反事実モデル：個体レベルの因果.

- 片方を観測してしまうと、もう片方は観測できない（**因果推論の根本問題**）

反事実モデル（集団レベルの因果） 9

ある集団のすべての個体がある病気にかかっていたとする。

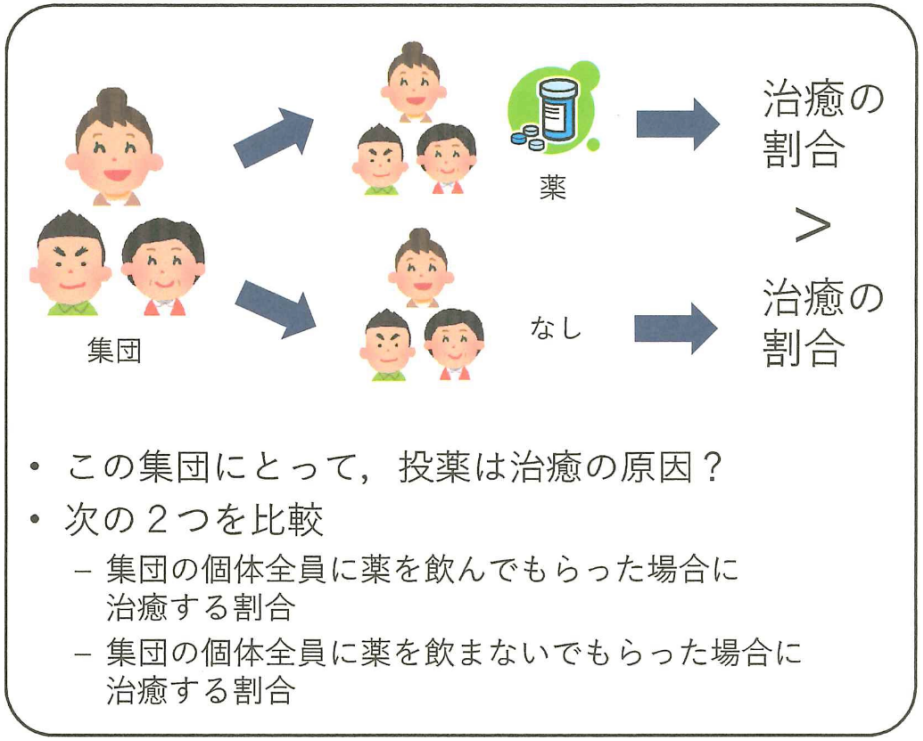


図 2.3　反事実モデル：集団レベルの因果.

集団について考える因果関係を、**集団レベルの因果**（**population-level causation**）という。

構造的因果モデル（SCM）

構造的因果モデル（structual causal model）

＝　反事実モデル　＋　構造方程式モデル

（構造方程式モデル：データ生成過程を決定的に表したもの）

定義　介入

ある変数 x に**介入する**とは、「他の変数がどんな値をとろうとも、変数 x の値を定数 c にとる」ことを意味する。

他の変数とは、観測される変数も、されない変数も含めたすべての変数。

このような介入を、 do という記号を用いて $\mathrm{do}(x = c)$ と表す。

- 病気の薬の例で言えば、 x に介入するとは、「年齢や性別、重症度に関わらず必ず薬を飲んでもらう」ということ。

構造的因果モデルでの因果の表現

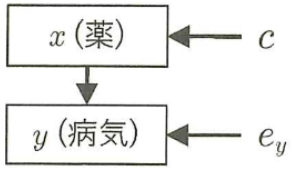
- 介入後のyの分布 := 介入後のモデル$M_{x=c}$におけるyの分布

$$p(y|\mathrm{do}(x=c)) := p_{M_{x=c}}(y)$$

- 介入後のモデル$M_{x=c}$

$$x = c$$
$$y = f_y(x, e_y)$$

構造方程式



因果グラフ

- 薬を飲むかどうかが病気が治るかどうかの原因となる.

$$p(y|\mathrm{do}(x=1)) \neq p(y|\mathrm{do}(x=0))$$

とりあえず、この論文では…

d 個の変数 $\mathbf{x} = (x_1, \ldots, x_d)$ の因果構造 G とは、それぞれの辺が $\mathbf{x}$ の変数間の因果効果を表した有向グラフ。

変数 x_i が変数 x_j に対して因果効果を持つとは、変数 x_i への介入が変数 x_j に、他の変数 $\mathbf{x}_{\backslash ij} := \mathbf{x} \backslash \{x_i, x_j\}$ とは独立に影響することである。つまり、

$$p(x_j | \mathrm{do}(x_i = a, \mathbf{x}_{\backslash ij} = \mathbf{c})) \neq p(x_j | \mathrm{do}(x_i = a', \mathbf{x}_{\backslash ij} = \mathbf{c}))$$

を満たす $a \neq a'$ が存在することである。

- 因果構造 G は非巡回だと嬉しい

背景：償却（変分）推論

償却推論とは

確率モデルにおける推論 = 事後分布の計算

でもモデルが複雑だと（解析的には）計算できない → **事後分布を近似**

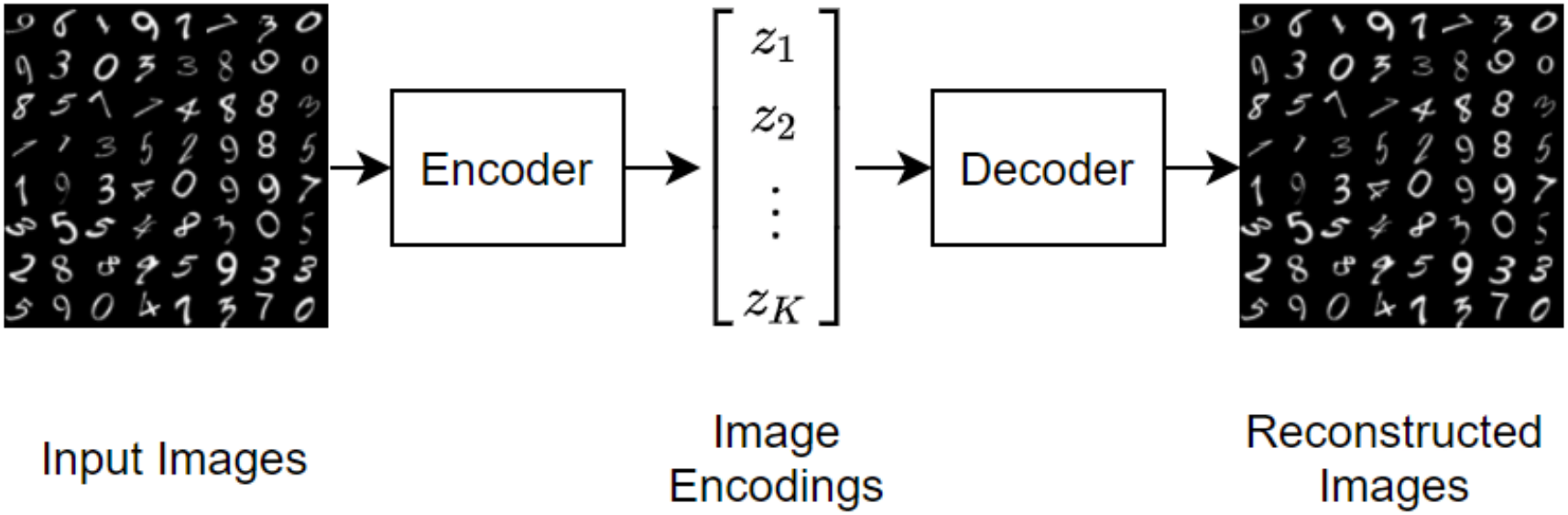
1. ギブスサンプリング
2. 変分推論
   やっぱり計算が大変

償却推論では…

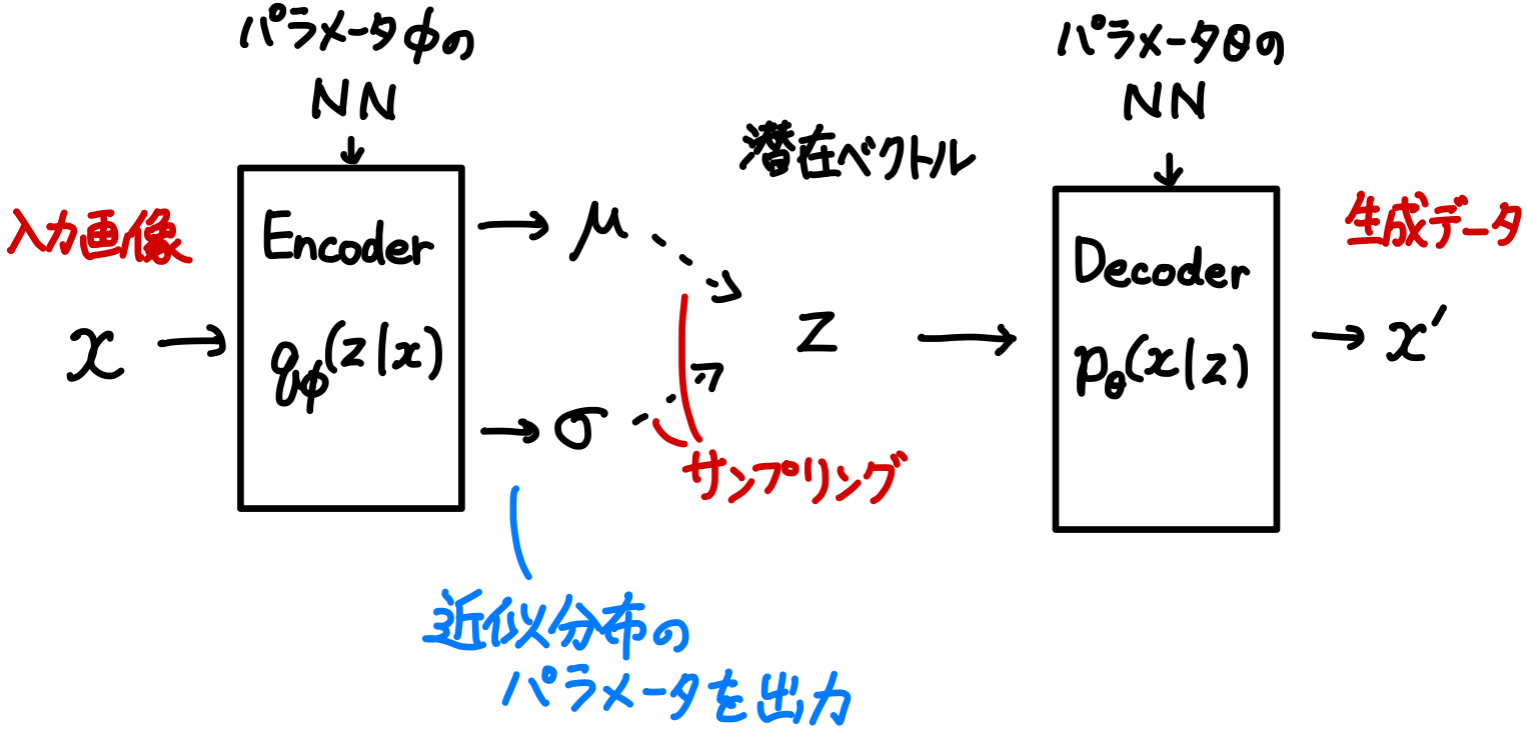
- 近似事後分布にパラメトリックな分布を仮定
- 近似分布のパラメータを、データから直接予測するモデル（関数）を作る
- 結局、この関数を最適化するという問題になる

償却推論の例：VAE

- 訓練データと似たような画像を作る生成モデルとして有名



VAEの仕組み（ざっくり）



$D_{KL}(q_\phi(z|x) \| p_\theta(z|x))$ が小さくなるように

ϕ, θ を学習

提案手法

AVICI: Amortized Variational Inference for Causal Discovery

問題設定

- データ生成分布： $p(D)$
- 観測データ： $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \sim p(D)$
- データ生成過程： $p(D|G)$

目標：観測データ D から因果構造の事後分布 $p(G|D)$ を $q(G;\theta)$ で近似する

そこで、近似分布のパラメータ θ を推論モデル f_ϕ で予測する（償却推論）。
事後分布と近似分布のforward KLを最小化するように、 f_ϕ を学習する。

$$\min_{\phi} \mathbb{E}_{p(D)} D_{\mathrm{KL}}(p(G|D) \| q(G; f_\phi(D)))$$

目的関数

$$
\begin{aligned}
&\mathbb{E}_{p(D)} D_{\mathrm{KL}}(p(G|D) \| q(G; f_\phi(D))) \\
&\quad = \mathbb{E}_{p(D)} \mathbb{E}_{p(G|D)} [\log p(G|D) - \log q(G; f_\phi(D))] \\
&\quad = -\mathbb{E}_{p(G)} \mathbb{E}_{p(D|G)} [\log q(G; f_\phi(D))] + \mathrm{const.}
\end{aligned}
$$

定数の部分は ϕ に依存しないので、 $\mathcal{L}(\phi) := \mathbb{E}_{p(G)} \mathbb{E}_{p(D|G)} [\log q(G; f_\phi(D))]$ を最大化すれば良い。

- 真のデータ生成分布 $p(G, D)$ からサンプルして、 $q(G; \theta)$ を予測する
- reverse KLの分散を小さく見積もってしまうという問題が起きないらしい
  - 変分推論ではよくreverse KLで最適化が行われる

---

reverse KL $D_{\mathrm{KL}}(q \| p)$ には再構成誤差の項 $\mathbb{E}_{q(G;\theta)}[\log p(D|G)]$ が含まれる。しかし、周辺尤度 $p(D|G)$ の計算をするために、モデルの制約が増やす必要がある。その必要のない今回のモデルをLikelihood-Free Inferenceと呼んでいる。

近似分布

近似分布 $q(G;\theta)$ は次のようにベルヌーイ分布を用いる。

$$q(G;\theta) = \prod_{i,j} q(g_{i,j};\theta_{i,j}) \quad \text{with} \quad g_{i,j} \sim \mathrm{Bern}(\theta_{i,j})$$

推論モデル f_ϕ は、 n 個のサンプル $\{\mathbf{o}^1, \ldots, \mathbf{o}^n\}$ に対応するデータセット D に、 $d \times d$ 行列を対応させる写像となる。

- それぞれのサンプル $\mathbf{o}^i = (o^i_1, \ldots, o^i_d)$ には、観測された値 $\mathbf{x}^i = (x^i_1, \ldots, x^i_d)$ に加えて、その変数が介入を受けたかの情報も含まれる。
- 具体的には、 $o^i_j = (x^i_j, u^i_j)$ として、 $u^i_j \in \{0, 1\}$ はサンプル i において変数 j が介入を受けたかを表す。

推論モデル f_ϕ

推論モデル f_ϕ には８層のニューラルネットワークを用いている。
具体的なNNの構成についてはよく理解できなかったので省略する。

- Transformerなどで用いられているmulti-head self-attentionが利用されているらしい

f_ϕ はサンプルの順番や特徴量の並び順によって予測が変わってほしくない。
→ Max Poolingによってこの条件を達成している。

ネットワークの出力を $\mathbf{u}^i, \mathbf{v}^i \in \mathbb{R}^k$ とすると、因果グラフの辺が存在する確率 $\theta_{i,j}$ は次のように決まる。

$$\theta_{i,j} = \sigma(\tau \mathbf{u}^i \cdot \mathbf{v}^i + b)$$

ここで、 σ はロジスティック関数。

非巡回の制約

特定のドメインでは因果グラフが非巡回という仮定をつけるとよく予測できる。

このような制約は、 ϕ の制約として次のように書くことができる。

$$\mathcal{F}(\phi) := \mathbb{E}_{p(D)}[h(f_\phi(D))] = 0$$

※ h の中身については省略

$\mathcal{F}(\phi) = 0$ のもとで $\mathcal{L}(\phi)$ を最大化するので、次のような問題を解けば良い。

$$\min_{\lambda} \max_{\phi} \mathcal{L}(\phi) - \lambda \mathcal{F}(\phi)$$

最適化アルゴリズム

Algorithm 1 Training the inference model f_ϕ

Parameters: ϕ variational, λ dual, η step size
while not converged **do**
  for l **steps do**
    $\Delta\phi \propto \nabla_\phi\big(\mathcal{L}(\phi) - \lambda\mathcal{F}(\phi)\big)$
  $\lambda \leftarrow \lambda + \eta\mathcal{F}(\phi)$

- 非巡回の制約をつけない場合は $\lambda = 0$ で開始する

実験

データセット

真の因果構造がわかっている現実のデータセットが存在しない！

この論文では３つの人工データを使って検証している。それぞれ、o.o.dやノイズが乗っているデータの場合も調べている

- SCM with Linear functions (LINEAR)
- SCM with nonlinear functions of random Fourier feature (RFF)

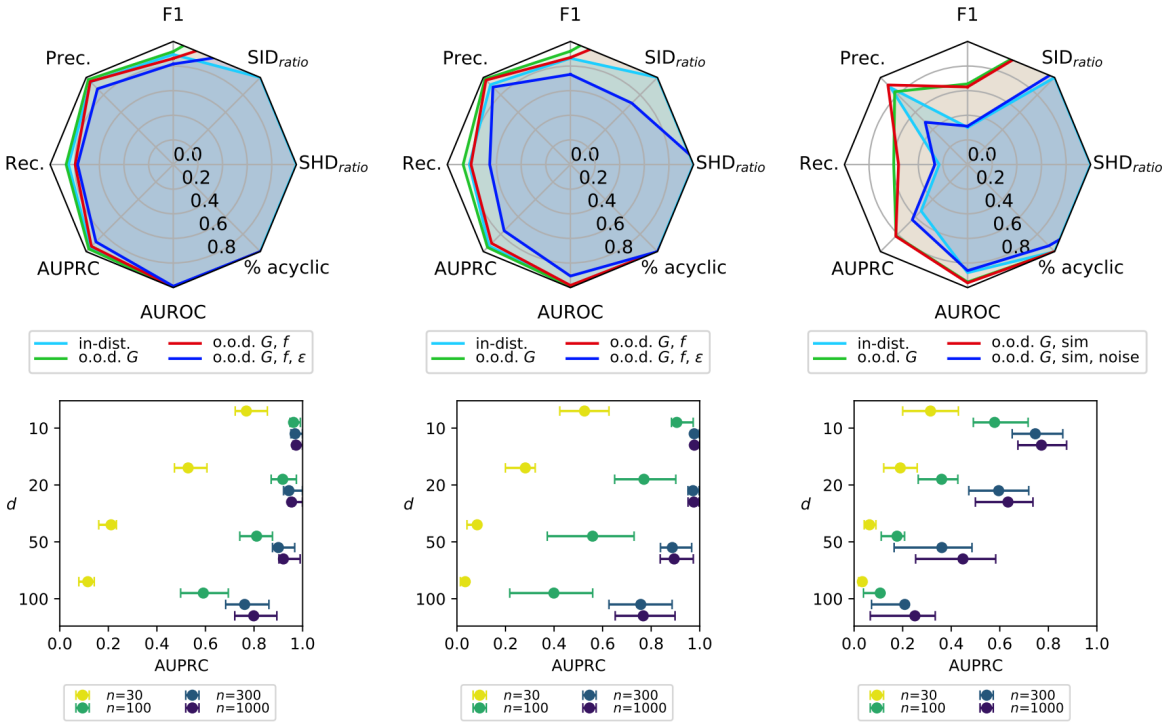
↑このふたつは構造因果モデルをもとにしたシミュレーション。
因果構造にはスケールフリーネットワークを利用している。

- semisyntheticsingle-cell expression data of gene regulatory networks (GRNs)
  - 細胞内の確率的な遺伝子発現のシミュレータらしい（よくわからん）

評価指標

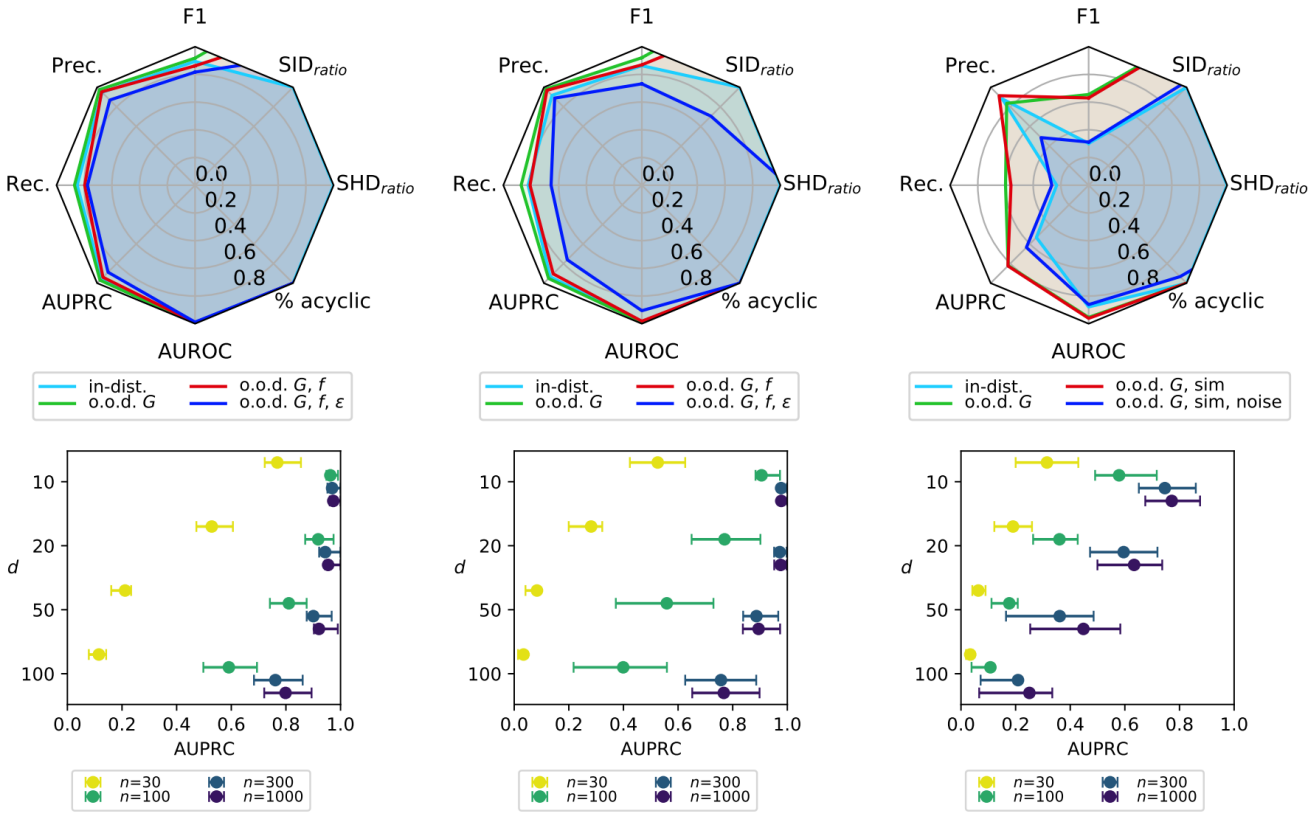
- 構造ハミング距離（SHD）：グラフ間の編集距離
- 構造介入距離（SID）：グラフの近さの定量化
- single-edge precision, recall, and F1 score
- AUPRC, AUROC

結果：OODデータに対する性能



(a) LINEAR **(b) RFF** **(c) GRN**

- LEINEARとRFFでは提案手法は良い性能を発揮できる
- GRNではむしろOODデータのほうがうまく予測できる



(a) LINEAR **(b)** RFF **(c)** GRN

下段：

- データ数を増やすと性能は上がるが、限界がある
- 変数が増えてタスクの難易度が上がると、なめらかに性能が下がる

結果：他の手法との比較

	LINEAR		RFF		GRN	
Algorithm	SID	F1	SID	F1	SID	F1
GES	**215.6** (35.0)	0.548 (0.03)	**346.3** (44.4)	0.285 (0.03)	**573.6** (29.2)	**0.058** (0.01)
LiNGAM	413.4 (48.4)	0.369 (0.04)	410.3 (47.6)	0.238 (0.02)	**617.5** (31.7)	**0.044** (0.01)
PC	400.5 (53.7)	0.338 (0.03)	**370.1** (51.2)	0.421 (0.03)	**594.0** (30.0)	**0.061** (0.01)
DAG-GNN	474.5 (50.8)	0.154 (0.01)	425.3 (50.2)	0.221 (0.03)	**588.7** (36.6)	**0.078** (0.02)
GraN-DAG	466.0 (54.3)	0.200 (0.03)	**328.6** (48.4)	**0.476** (0.05)	**582.4** (33.4)	**0.073** (0.02)
AVICI (ours)	**145.6** (21.5)	**0.672** (0.04)	**255.1** (48.2)	**0.618** (0.06)	**641.7** (34.7)	0.000 (0.00)
GIES	**120.8** (26.2)	0.736 (0.03)	**304.8** (44.0)	0.338 (0.04)	545.5 (26.9)	0.092 (0.01)
IGSP	244.0 (34.4)	0.559 (0.02)	374.1 (45.0)	0.407 (0.04)	597.4 (31.7)	0.057 (0.01)
DCDI	383.5 (45.1)	0.327 (0.03)	**282.8** (46.3)	0.409 (0.04)	590.9 (30.6)	0.075 (0.02)
AVICI (ours)	**110.9** (19.3)	**0.819** (0.02)	**192.7** (44.8)	**0.707** (0.06)	**416.9** (47.1)	**0.338** (0.06)

- 多くの場合に既存のモデルよりも良い性能を達成している

所感

- VAEの仕組みは知っていたけど、償却推論は知らなかった
  - VAEなどでうまくいっている方法だけあって強い
- 検証できるデータセットが現実にないのは難しい
  - 生物分野のシミュレータが使われていて面白い
- グラフを直接NNに予想させるというのが新鮮だった
  - GNNなど、グラフを学習するだけだと思っていた