TDLS:
Learning Functional Causal Models with Generative Neural Networks

Discussion Lead: Christopher Alert (Bell Canada)

Discussion Facilitators: Rohollah Soltani, Masoud Hashemi

The Paper

Learning Functional Causal Models with Generative Neural Networks

Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, Michèle Sebag

12/3/2018 (v1: 9/15/2017) stat.ML

Explainable and Interpretable Models in Computer Vision and Machine Learning. Springer Series on C...

1709.05321v3 pdf

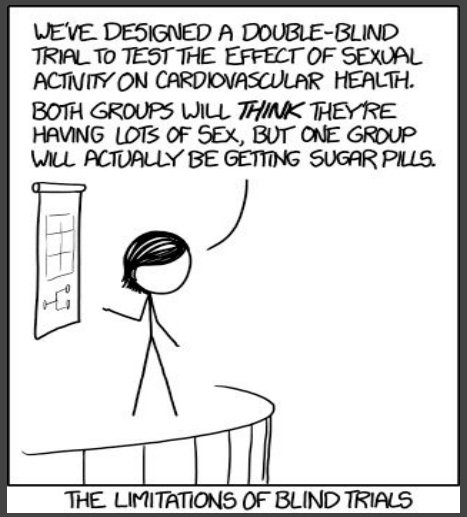
show similar | discuss

Key Contribution: Train generative nn-model to simulate interventions on one or more variables in a system and evaluate their impact on a set of target variables

https://github.com/GoudetOlivier/CGNN

TOC

Causal Inference and Functional Causal Models (FCM)

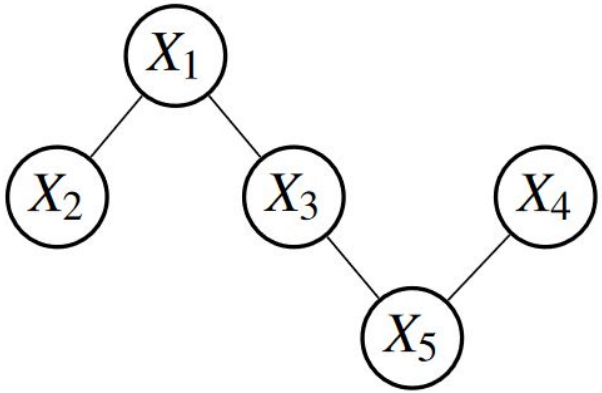


THE LIMITATIONS OF BLIND TRIALS

source: xkcd.com

Causal Inference: Definitions

X=[X_1, X_2, … X_d]: observed variables of interest w/ joint distribution P(**X**)

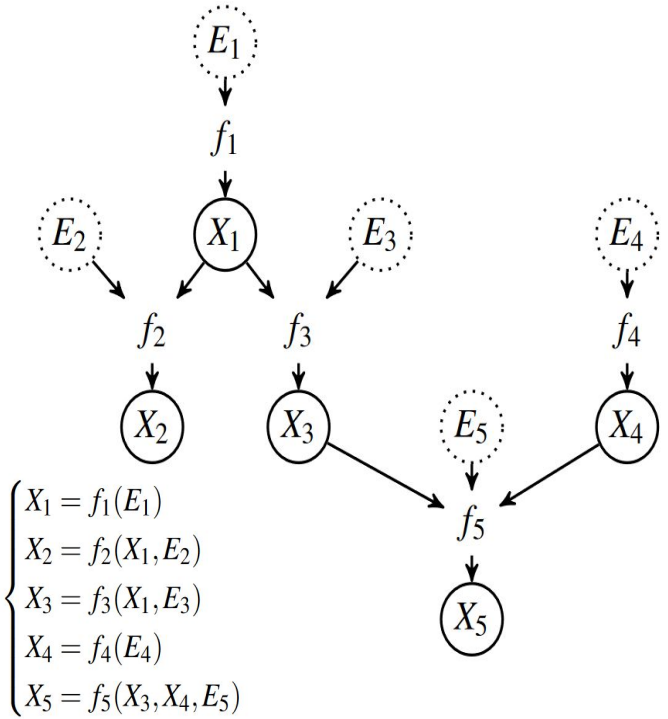


skeleton (S): an undirected graph where every path in S denotes some relationship of dependence between variables on the path

Causal Inference: Definitions

Functional Causal Model: ($\mathcal{G}, f, \varepsilon$)

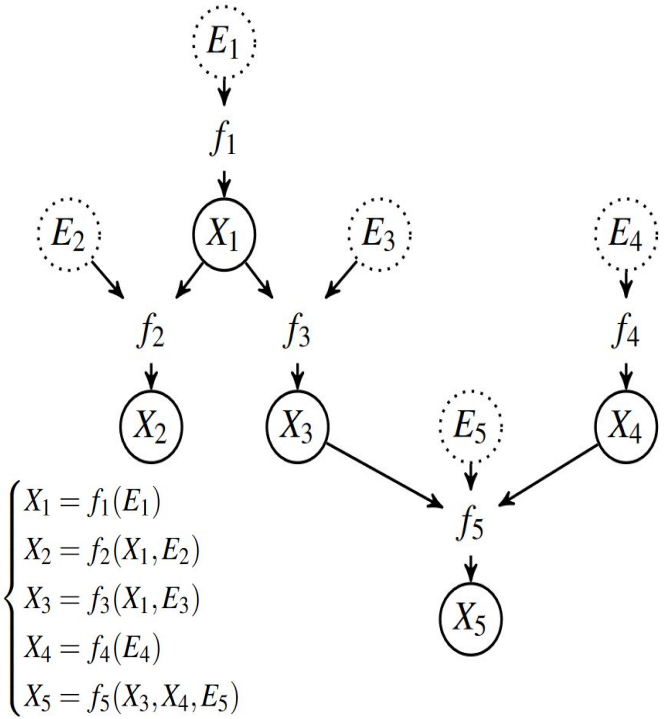
$\mathcal{G}$: directed acyclic graph (DAG) drawn by orienting edges from $X_{Pa(i;G)}$, E_i towards their effects X_i



Causal Inference: Definitions

Functional Causal Model: triplet ($\mathcal{G}, f, \varepsilon$)

f=(f_1,...,f_n):causal mechanisms f_i defining a mapping from causes X_j and unobserved variables E_i to effects X_i

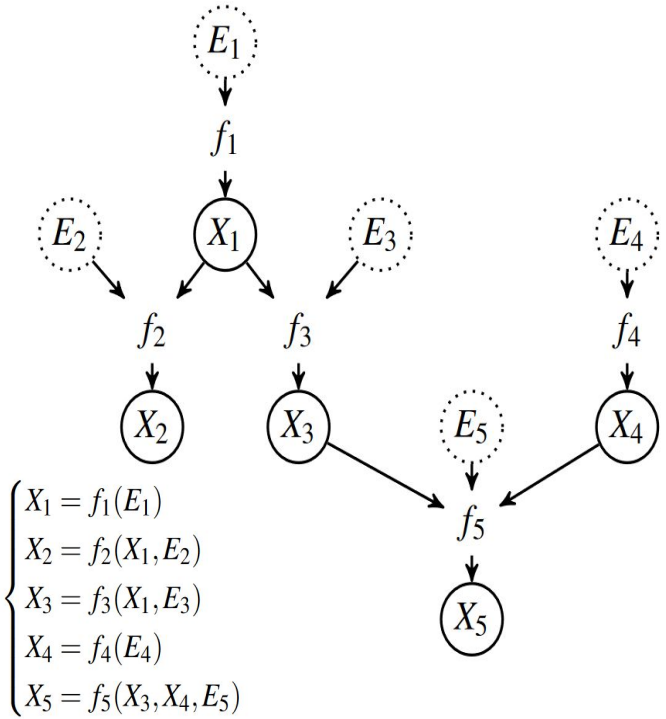


Causal Inference: Definitions

Functional Causal Model: triplet ($\mathcal{G}, f, \varepsilon$)

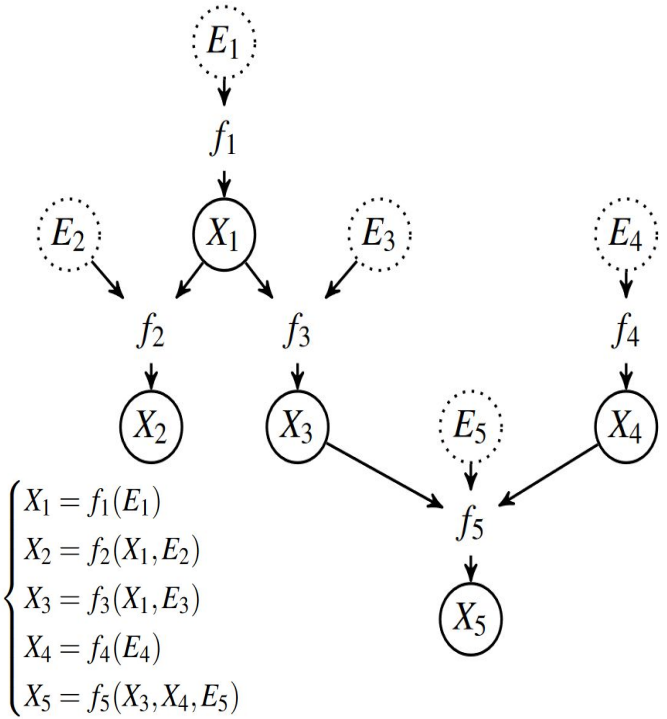
$E=(E_1, \ldots, E_n)$: accounting for noise and unobserved variables

Each E_i assumed independent of other E_j's and all $E_i \sim \varepsilon$



Causal Inference: Definitions

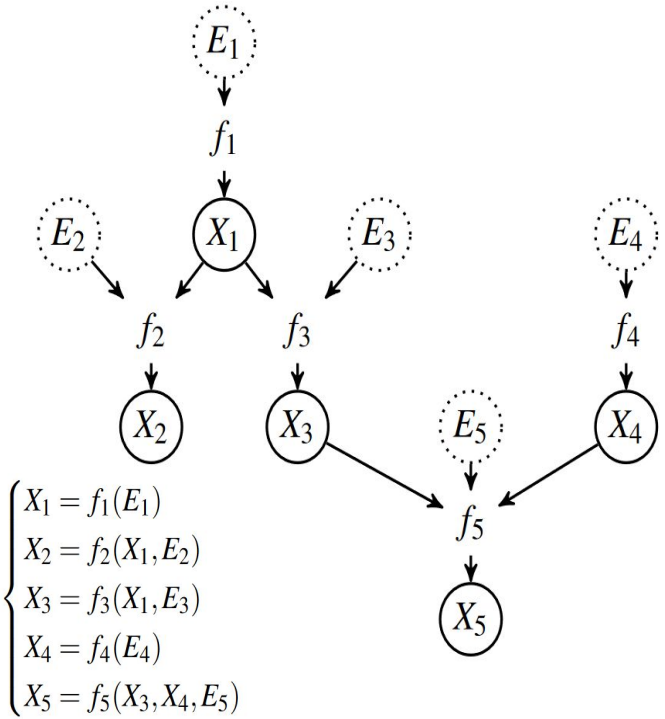
Causal Sufficiency Assumption (CSA): the variables X_i in your observed sample and error terms E_i capture all relevant causal influences on target variables of interest



Causal Inference: Goal

Simulate interventions on one or more variables in a system and evaluate their impact on a set of target variables

interventional distribution $P_{do(Xi= vi)}(X)$: obtained by clamping variable X_i to value v_i



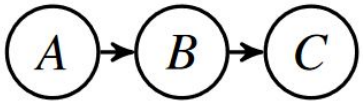
Causal Inference: Bivariate cause-effect problem

Unseen generative process: A: # price of cigarettes,
B: # cigarettes smoked per day , C: # lung cancer cells

$E_B, E_C \sim N(\mu, \sigma)$

$B \leftarrow 0.5\,A + E_B,$

$C \leftarrow B + E_C$



$$\begin{cases} A = E_A \\ B = A + E_B \\ C = B + E_C \end{cases}$$

Causal Inference: Bivariate cause-effect problem

Unseen generative process: A: # price of cigarettes,
B: # cigarettes smoked per day , C: # lung cancer cells

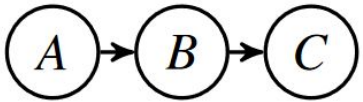
Problem of identifiability: How to distinguish between directions of causal orientation?

$B^* = 0.25\,A + 0.5\,C$

$B \leftarrow \alpha\, C$?

or

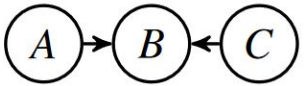
$C \leftarrow \alpha\, B$?



$$\begin{cases} A = E_A \\ B = A + E_B \\ C = B + E_C \end{cases}$$

Causal Inference: v-structure identification

Consider variables: A = student IQ ,B = test score , C = test difficulty



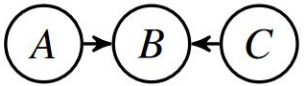
P(A = a| C=c) = P(A = a)

$$\begin{cases} A = E_A \\ C = E_C \\ B = A + C + E_B \end{cases}$$

Test difficulty alone gives you no more information!

Causal Inference: v-structure identification

Consider variables: A = student IQ ,B = test score , C = test difficulty



P(A = a | C=c, B=b) ≠≠≠ P (A=a | C=c) = P(A = a)

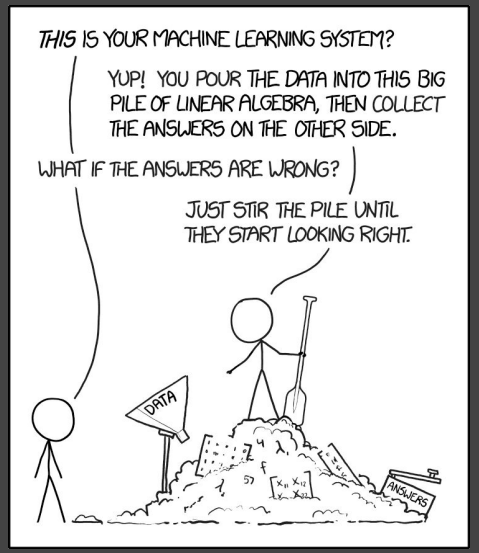
$$\begin{cases} A = E_A \\ C = E_C \\ B = A + C + E_B \end{cases}$$

<u>However</u>, knowing both test difficulty and a student's score on a test , you can make inferences about the student's IQ

Causal Inference: Families of Learning Algorithms

- **Constraint based:** Recover graph structure using tests of conditional independence
- **Score based:** Explore space of graphs while minimizing a global score
- **Hybrid method**: combination of constraint / score based methods
- **Pairwise methods:** restricting the class of functions allowed for causal mechanisms f_i and assuming a functional form
    - Regularize functions f_i with respect to local score and (empirically) helps the problem of identifiability

Causal Generative Neural Networks



source: xkcd.com

Causal Generative Neural Network: Definition

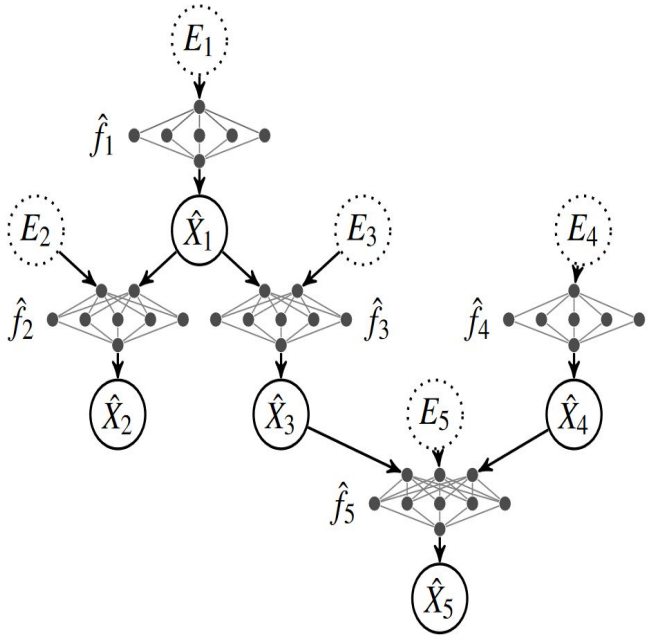
A CGNN over $[\dot{X}_1, \dots \dot{X}_d]$ is a triplet $C_{\hat{G},\hat{f}} = (\hat{G}, \hat{f}, \varepsilon)$ where:

Causal mechanisms $\hat{f}_i$ are 1-hidden layer regression neural networks

n_h: # of hidden neurons in each causal mechanism $\hat{f}_i$

RELU activation units

Each E_i is independent of X_i. Further, all E_i are i.i.d ~ ε

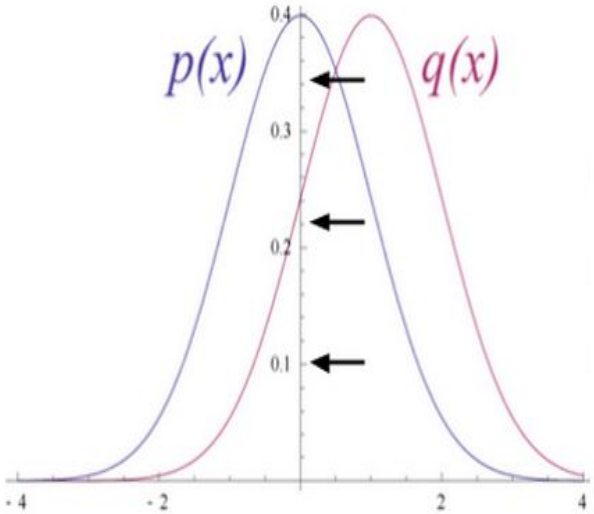


CGNN: Loss Function

Comparing the distribution of the data generated by the CGNN with the distribution of the sample observational data

$\dot{D} \sim p(x) = \{\dot{x}_i\}_{i=1}^{n}$: sampled from the generative model $C_{\hat{G},\hat{f}}$,

$D \sim q(x) = \{x_i\}_{i=1}^{n}$: sample observational data



CGNN: Loss Function

Maximum Mean Discrepancy (MMD) measures the distance between the means of two probability distributions p(x) and q(x) in a kernel embedding space

$$\text{Loss} = S\left(C_{\hat{G},\hat{f}}, D\right) = \text{MMD}_k(D, \dot{D}) + \lambda |\hat{G}|$$

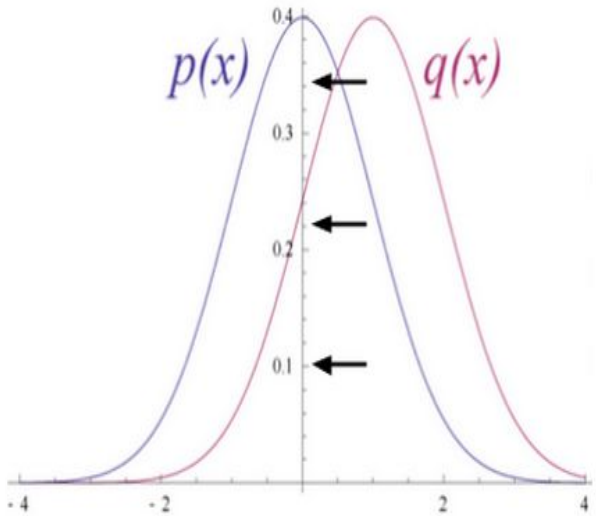
where:

$|\hat{G}|$: # edges in $\hat{G}$

$D = \{x_i\}_{i=1}^{n}$: sample observational data

$\dot{D} = \{\dot{x}_i\}_{i=1}^{n}$: sampled from the generative model $C_{\hat{G},\hat{f}}$,

kernel k is Gaussian kernel (differentiable)



CGNN: Searching Causal Graphs with CGNN

Input(s): S an identified skeleton of causal graph G ; objective function: regularized MMD_k

Initialize: $\hat{G}$ as current Graph, G_{old} = {} as a set of graphs already considered

For n_{train} iterations:

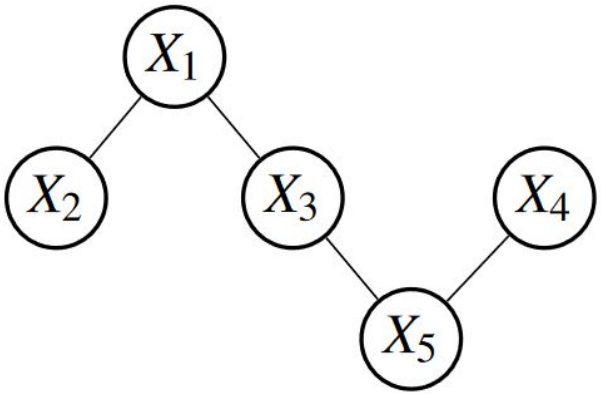
CGNN: Searching Causal Graphs with CGNN

Input(s): S an identified skeleton of causal graph G ; objective function: regularized MMD_k

Initialize: $\hat{G}$ as current Graph, G_{old} = {} as a set of graphs already considered

For n_{train} iterations:

1. Orient each edge X_i - X_j by according to minimum 2-variable CGNN score, producing new $\hat{G}$



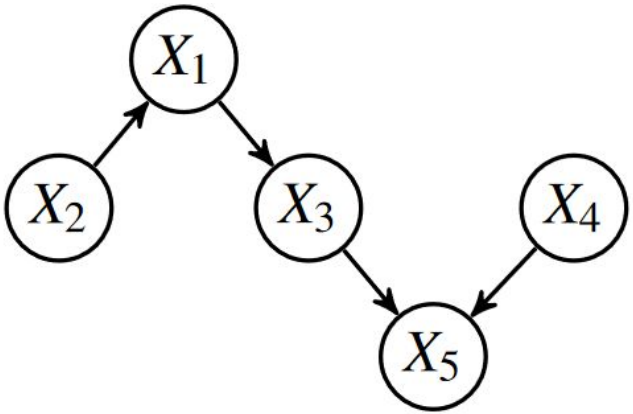
CGNN: Searching Causal Graphs with CGNN

Input(s): S an identified skeleton of causal graph G ; objective function: regularized MMD_k

Initialize: $\hat{G}$ as current Graph, G_{old} = {} as a set of graphs already considered

For n_{train} iterations:

1. Orient each edge X_i - X_j by according to minimum 2-variable CGNN score, producing new $\hat{G}$
2. Traverse $\hat{G}$ and remove cycles by reversing edges until you obtain a DAG



CGNN: Searching Causal Graphs with CGNN

Input(s): S an identified skeleton of causal graph G ; objective function: regularized MMD_k

Initialize: $\hat{G}$ as current Graph, G_{old} = {} as a set of graphs already considered

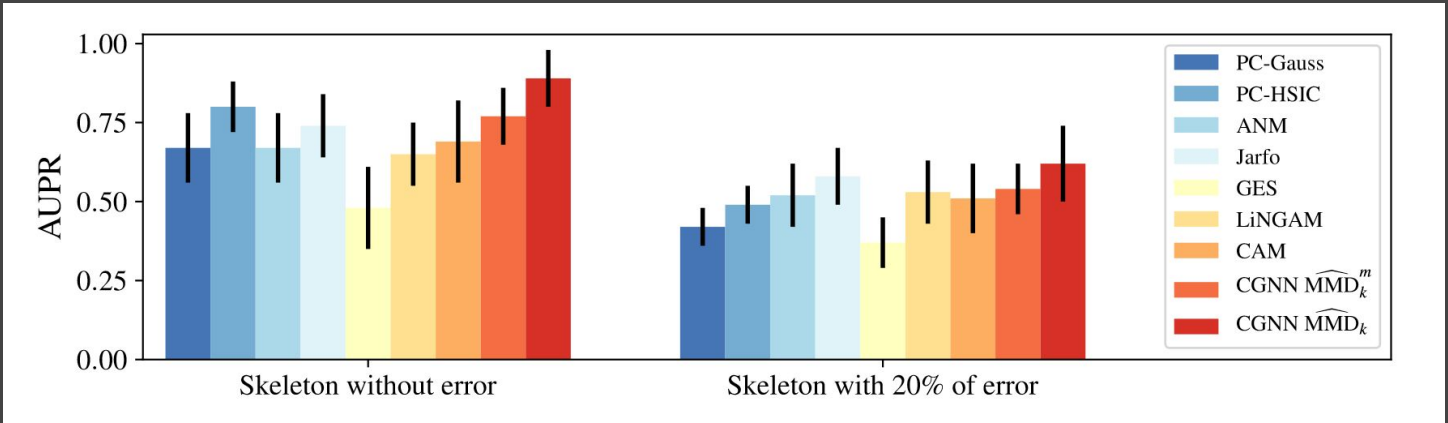
1. Orient each edge X_i - X_j by according to minimum 2-variable CGNN score, producing new $\hat{G}$
2. Traverse $\hat{G}$ and remove cycles by reversing edges until you obtain a DAG
3. **For n_{train} iterations**:
    - Randomly sample an edge in the skeleton
    - Reverse the edge (if still a DAG and $G_{reverse}$ not in G_{old}) and retrain the associated global CGNN
    - If this retrained graph, G', obtains better (lower) global score, then $\hat{G}$ = G '
    - Process repeated until reaching a local optimum

Compute confidence scores $V_{Xi \to Xj}$ = S (G, D) - S(G - $\{X_i \to X_j\}$, D) for each edge $X_i \to X_j$ in $\hat{G}$

Break

- 2 minutes -

Experimental Results

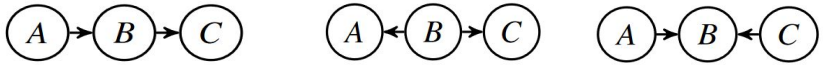


Experimental Setting: V- structure Identification

Artificial data
n = 500
A~N(μ_A, σ_A), B~N(μ_B, σ_B), C ~N(μ_C, σ_C) , ε ~ N(0,1)
Generated from skeleton: A--B--C

Score	non V-structures: Chain str.	non V-structures: Reversed-V str.	V structure: V-structure
C_{ABC}	*0.122 (0.009)*	*0.124 (0.007)*	0.172 (0.005)
C_{CBA}	*0.121 (0.006)*	*0.127 (0.008)*	0.171 (0.004)
$C_{reversedV}$	*0.122 (0.007)*	*0.125 (0.006)*	0.172 (0.004)
$C_{V\,structure}$	0.202 (0.004)	0.180 (0.005)	**0.127** (0.005)

(a) *Chain structure* (c) *reversed-V structur* (d) *V-structure*



$$\begin{cases} A = E_A \\ B = A + E_B \\ C = B + E_C \end{cases} \quad \begin{cases} B = E_B \\ A = B + E_A \\ C = B + E_C \end{cases} \quad \begin{cases} A = E_A \\ C = E_C \\ B = A + C + E_B \end{cases}$$

Let C_{ABC}, C_{CBA}, $C_{v\text{-structure}}$ and $C_{reversed\text{-}V}$ denote scores of CGNN models respectively attached to these structures

(average scores over 64 runs)

Model can clearly distinguish v-structures present in data

Experimental Setting: Cause-effect Inference

Dataset	#Pairs	Description(n ≤1500)
CE- Cha	300	Real and artificial data: A->B, A|B, A-B
CE- Net	300	**X:** random distbn ***f*** : neural networks
CE- Gauss	300	**X**: Gaussian rv ***f***: Gaussian process
CE- Multi	300	Y = f(X)+E / f(X) · E / f(X+E) / f(X · E)
CE- Tüb	99	Finance, climatology, medicine
Training time: 24 mins on GPU CGNN vs 32 mins on CPU for GPI		

Table 1: Cause-effect relations: Area Under the Precision Recall curve on 5 benchmarks for the cause-effect experiments (weighted accuracy in parenthesis for Tüb)

method	Cha	Net	Gauss	Multi	Tüb
Best fit	56.4	77.6	36.3	55.4	58.4 (44.9)
LiNGAM	54.3	43.7	66.5	59.3	39.7 (44.3)
CDS	55.4	89.5	84.3	37.2	59.8 (65.5)
IGCI	54.4	54.7	33.2	80.7	60.7 (62.6)
ANM	66.3	85.1	88.9	35.5	53.7 (59.5)
PNL	73.1	75.5	83.0	49.0	68.1 (66.2)
Jarfo	79.5	92.7	85.3	94.6	54.5 (59.5)
GPI	67.4	88.4	89.1	65.8	66.4 (62.6)
CGNN ($\widehat{\text{MMD}}_k$)	73.6	89.6	82.9	96.6	79.8 (74.4)
CGNN ($\widehat{\text{MMD}}_k^m$)	76.5	87.0	88.3	94.2	76.9 (72.7)

CGNN out-performed competing methods on real world datasets

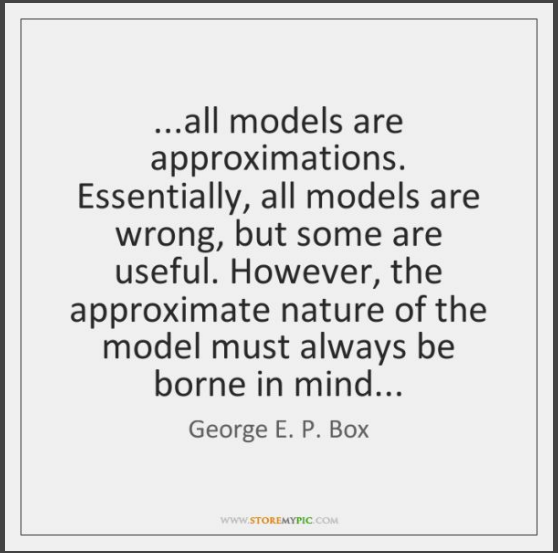
Experimental Setting: Multivariate discovery

Artificial data
20 training graphs ; 20 test graphs
n= 500 ; d = 20
fi: randomly generated polynomials with additive/multiplicative noise $\|Pa(i; \mathcal{G})\| \sim U[0,5]$ for all X_i
CGNN trained on: (1) true skeleton , (2) skeleton w/ 20% edges perturbed
Training time: 4 hrs on GPU for CGNN vs 15 hrs CPU for PC-HSIC (30 GPU hrs for 5 graphs, d=100)

method	Skeleton without error AUPR	SHD	SID	Skeleton with 20% of error AUPR	SHD	SID
Constraints						
PC-Gauss	0.67 (0.11)	9.0 (3.4)	131 (70)	0.42 (0.06)	21.8 (5.5)	191.3 (73)
PC-HSIC	0.80 (0.08)	6.7 (3.2)	80.1 (38)	0.49 (0.06)	19.8 (5.1)	165.1 (67)
Pairwise						
ANM	0.67 (0.11)	7.5 (3.0)	135.4 (63)	0.52 (0.10)	19.2 (5.5)	171.6 (66)
Jarfo	0.74 (0.10)	8.1 (4.7)	147.1 (94)	0.58 (0.09)	20.0 (6.8)	184.8 (88)
Score-based						
GES	0.48 (0.13)	14.1 (5.8)	186.4 (86)	0.37 (0.08)	20.9 (5.5)	209 (83)
LiNGAM	0.65 (0.10)	9.6 (3.8)	171 (86)	0.53 (0.10)	20.9 (6.8)	196 (83)
CAM	0.69 (0.13)	7.0 (4.3)	122 (76)	0.51 (0.11)	15.6 (5.7)	175 (80)
CGNN ($\widehat{\text{MMD}}_k^m$)	0.77 (0.09)	7.1 (2.7)	141 (59)	0.54 (0.08)	20 (10)	179 (102)
CGNN ($\widehat{\text{MMD}}_k$)	0.89* (0.09)	2.5* (2.0)	50.45* (45)	0.62 (0.12)	16.9 (4.5)	134.0* (55)

CGNN pairwise edge orientation more robust to incorrect skeleton

Takeaways and Discussion Points



Key Takeaways

- Pros:
  - CGNN can learn the structure of causal relationships between observed variables
  - Robust performance on real data or given a noisy skeleton of dependencies between variables
  - Provides a generative model to simulate interventions on one or more variables in a system and evaluate their impact
- Cons:
  - Models highly sensitive to n_h , the # neurons in each hidden layer in the causal mechanisms f_i
  - Graph searching algorithm is time expensive and does not parallelize

Discussion Points

- Is it possible to have a better causal graph searching algorithm in score-based methods, e.g., gradient free optimisation algorithms (simulated annealing and genetic algorithm) or Bayesian optimisation for better sampling?
- Are there any well-known graph structures that allow you to parallelize the searching algorithm by searching over subgraphs and optimizing local MMD_k score?

Thank you.

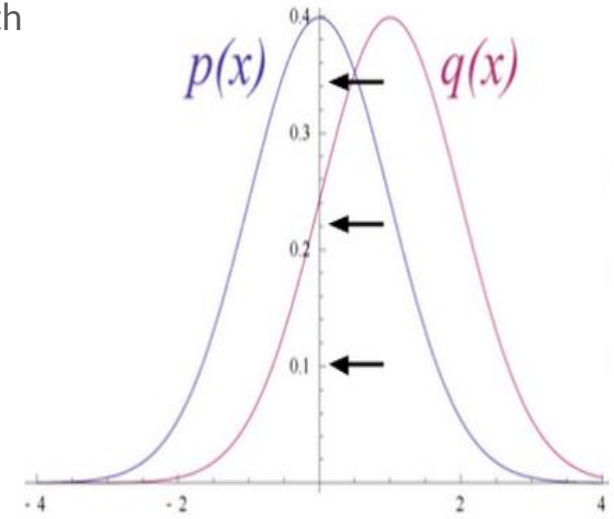
Acknowledgements:

Helen Ngo
Jack Gao
Andee Liao
Medha Patki

Appendix (extra slides in case)

CGNN: Loss Function

Maximum Mean Discrepancy (MMD) measures the distance between th means of two probability distributions p(x) and q(x) in some kernel embedding space



$$\text{Loss}\left(C_{\hat{G},\hat{f}}, D\right) = \frac{1}{n^2}\sum_{i,j}^{n} k(x_i, x_j) + \frac{1}{n^2}\sum_{i,j}^{n} k(\hat{x}_i, \hat{x}_j) - \frac{2}{n^2}\sum_{i,j}^{n} k(x_i, \hat{x}_j) + \lambda\,|\hat{G}|$$

where:

$|\hat{G}|$: # edges in $\hat{G}$

x_i: sample observational data

$\dot{x}_i$ sampled from generative model,

kernel k is Gaussian kernel: $k(x, x') = \exp(-\gamma||x - x'||_2^2)$

Causal Generative Neural Network: Definition

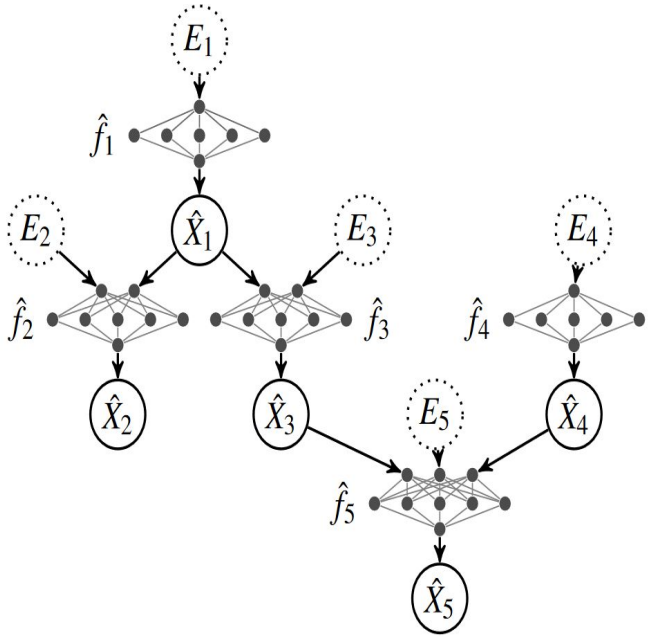
A CGNN over $[\dot{X}_1, \ldots \dot{X}_d]$ is a triplet $C_{\hat{G},\dot{f}} = (\hat{G}, \dot{f}, \varepsilon)$ where:

causal mechanism $\dot{f}_i$ are 1-hidden layer regression neural networks with n_h hidden neurons:

$$\hat{X}_i = \hat{f}_i(\hat{X}_{Pa(i;\hat{\mathcal{G}})}, E_i) = \sum_{k=1}^{n_h} \bar{w}_k^i \sigma \left(\sum_{j \in Pa(i;\hat{\mathcal{G}})} \hat{w}_{jk}^i \hat{X}_j + w_k^i E_i + b_k^i \right) + \bar{b}^i$$

with $n_h \in \mathbb{N}*$ the number of hidden units, $\bar{w}_k^i, \hat{w}_{jk}^i, w_k^i, b_k^i, \bar{b}^i \in \mathbb{R}$ the parameters of the neural network, and σ a continuous activation function .

Each E_i is independent of X_i. Further, all noise variables are i.i.d ~ ε



Causal Inference: Motivation