Learning Causal Structures via Gradient-Based Optimization

Sébastien Lachapelle

Mila, Université de Montréal

March 4th, 2020

Overview

- Causality Framework
  - Causal Graphical Models
  - Motivating example
  - Markov Equivalence and Structure Identifiability
- Causal Structure Learning
  - Problem formulation
  - Discrete Search Algorithms
  - Gradient-Based Algorithms
- GraN-DAG & extensions
  - The algorithm
  - With interventional data
  - Neural Autoregressive Flows

Causal graphical models (CGM)

- Random vector $X \in \mathbb{R}^d$ (d variables)
- Let $\mathcal{G}$ be a **directed acyclic graph** (DAG)
- Assume $p(x) = \prod_{i=1}^{d} p(x_i | x_{\pi_i^{\mathcal{G}}})$
  $\pi_i^{\mathcal{G}} =$ parents of i in $\mathcal{G}$
- Encodes **(conditional) independence statements** (via *d-separation*, see [Koller & Friedman, 2009])
- Almost identical to Bayesian Networks but allows for **interventional distributions:** $p(x|do(z))$

Simple example

$\mathcal{G} = (V, E)$

$$p(x, y, z) = p(x)p(z \mid x)p(y \mid z)$$

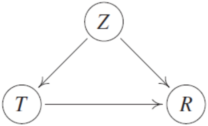
$$\implies p(x, y|z) = p(x|z)p(y|z)$$

i.e. $X \perp\!\!\!\perp Y \mid Z$

The **do operator** will be explained in the following example...

Why should you care: Kidney Stone Treatment

T = Treatment $\in \{A, B\}$
Z = Stone size $\in \{\text{small}, \text{large}\}$
R = Patient recovered $\in \{0, 1\}$



	Overall	Patients with small stones	Patients with large stones
Treatment *a*: Open surgery	78% (273/350)	**93%** (81/87)	**73%** (192/263)
Treatment *b*: Percutaneous nephrolithotomy	**83%** (289/350)	87% (234/270)	69% (55/80)

(Example taken from *Element of Causal Inference* by Peters et al. p111)

Why should you care: Kidney Stone Treatment

Pay attention to these two questions...
Assuming the size of your stone is unknown...

Why should you care: Kidney Stone Treatment

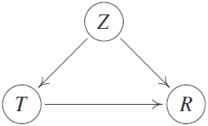
Pay attention to these two questions...
Assuming the size of your stone is unknown...

What is your chance of recovery knowing that the doctor gave you treatment A?

What is your chance of recovery if you decide to take treatment A?

Why should you care: Kidney Stone Treatment

T = Treatment $\in \{A, B\}$
Z = Stone size $\in$ {small, large}
R = Patient recovered $\in \{0, 1\}$



What is your chance of recovery knowing that the doctor gave you treatment A?

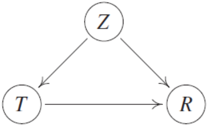
- Knowing that your doctor gave you treatment A tells you that you probably have a large kidney stone ... $P(Z = large | T = A) = 0.75$

- ... which reduces your chance of recovery $P(R = 1 | T = A, Z = large) = 0.73 < 0.93 = P(R = 1 | T = A, Z = small)$

Mila Université de Montréal

Why should you care: Kidney Stone Treatment

T = Treatment $\in \{A, B\}$
Z = Stone size $\in \{\text{small}, \text{large}\}$
R = Patient recovered $\in \{0, 1\}$



What is your chance of recovery knowing that the doctor gave you treatment A?

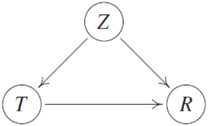
- Knowing that your doctor gave you treatment A tells you that you probably have a large kidney stone ... $P(Z = large | T = A) = 0.75$
- ... which reduces your chance of recovery $P(R = 1 | T = A, Z = large) = 0.73 < 0.93 = P(R = 1 | T = A, Z = small)$

What is your chance of recovery if you decide to take treatment A?

- Your really don't know anything about your kidney stone
- **You taking treatment A is not a function of any variable**

Why should you care: Kidney Stone Treatment

T = Treatment $\in \{A, B\}$
Z = Stone size $\in \{\text{small}, \text{large}\}$
R = Patient recovered $\in \{0, 1\}$



What is your chance of recovery knowing that the doctor gave you treatment A?

$P(R = 1|T = A) = 0,78$ $P(R = 1|T = B) = \mathbf{0,83}$

Why should you care: Kidney Stone Treatment

T = Treatment $\in \{A, B\}$
Z = Stone size $\in$ {small, large}
R = Patient recovered $\in \{0, 1\}$

$$
\begin{tikzcd}
 & Z \arrow[dl] \arrow[dr] & \\
T \arrow[rr] & & R
\end{tikzcd}
$$

What is your chance of recovery knowing that the doctor gave you treatment A?

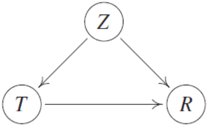
$P(R = 1|T = A) = 0,78$ $\quad$ $P(R = 1|T = B) = \mathbf{0{,}83}$

What is your chance of recovery if you decide to take treatment A?

$P(R = 1|do(T = A)) = \mathbf{0{,}832}$ $\quad$ $P(R = 1|do(T = B)) = 0,782$

Why should you care: Kidney Stone Treatment

T = Treatment $\in \{A, B\}$
Z = Stone size $\in \{\text{small}, \text{large}\}$
R = Patient recovered $\in \{0, 1\}$



What is your chance of recovery knowing that the doctor gave you treatment A?

$P(R = 1 | T = A) = 0,78$ $\quad$ $P(R = 1 | T = B) = \mathbf{0{,}83}$

What is your chance of recovery if you decide to take treatment A?

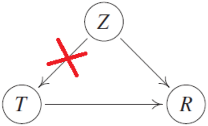
$P(R = 1 | do(T = A)) = \mathbf{0{,}832}$ $\quad$ $P(R = 1 | do(T = B)) = 0,782$

But how do we compute these **interventional distributions**?!

Mila Université de Montréal

Why should you care: Kidney Stone Treatment

T = Treatment $\in \{A, B\}$
Z = Stone size $\in \{\text{small}, \text{large}\}$
R = Patient recovered $\in \{0, 1\}$



$$P(R, Z|do(T = A)) = P(R|Z, T = A)\underbrace{\cancel{P(T = A|Z)}}_{\text{The decision of taking treatment } A \text{ does not depend on } Z \text{ anymore}} P(Z)$$

Then simply marginalize as usual:

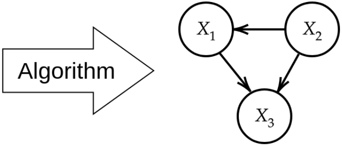
$$P(R = 1|do(T = A)) = \sum_Z P(R = 1, Z|do(T = A))$$
$$= \sum_Z P(R = 1|Z, T = A)P(Z) = 0,832$$

Structure Learning

- In the kidney stone example, **the causal graph was known**
- What if we don't have it? **Learn it!**

Purely observational data

	X_1	X_2	X_3
sample 1	1.76	10.46	0.002
sample2	3.42	78.6	0.011
...		...	
sample n	4.56	9.35	1.96

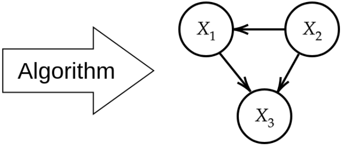


Structure Learning

- In the kidney stone example, **the causal graph was known**
- What if we don't have it? **Learn it!**

Purely observational data

	X_1	X_2	X_3
sample 1	1.76	10.46	0.002
sample2	3.42	78.6	0.011
...		...	
sample n	4.56	9.35	1.96



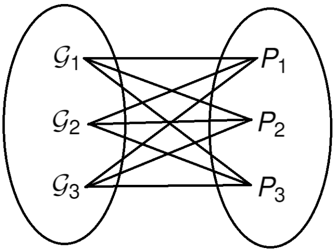
Is it even possible?

Identifiability

- In general, this is impossible without interventional data...

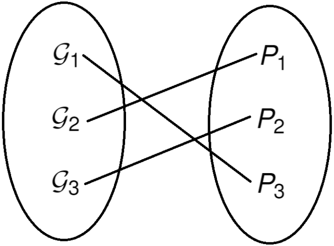
Identifiability

- In general, this is impossible without interventional data...
- Multiple DAGs can express the same distribution...



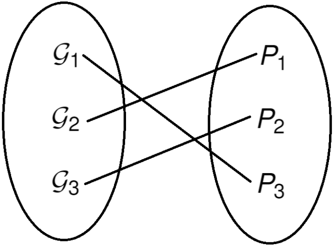
Identifiability

- If we assume causal mechanisms are "simple", then $\mathcal{G}$ can be identified...



Identifiability

- If we assume causal mechanisms are "simple", then $\mathcal{G}$ can be identified...



An example (useful later!)

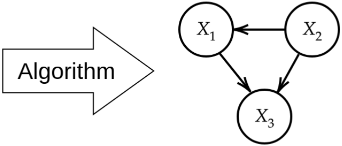
If data follows this model...

$$X_i | X_{\pi_i^{\mathcal{G}}} \sim \mathcal{N}(f_i(X_{\pi_i^{\mathcal{G}}}), \sigma_i^2)$$

...then **correct causal DAG $\mathcal{G}$ can be identified from purely observational data** (see [Peters et al., 2014] for proof and regularity conditions)

Structure Learning

	X_1	X_2	X_3
sample 1	1.76	10.46	0.002
sample2	3.42	78.6	0.011
...		...	
sample n	4.56	9.35	1.96



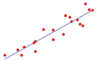
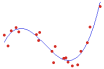
Score-based algorithms

$$\hat{\mathcal{G}} = \arg\max_{\mathcal{G} \in \text{DAG}} \text{Score}(\mathcal{G})$$

Often, $\text{Score}(\mathcal{G}) =$ regularized maximum likelihood under $\mathcal{G}$

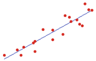
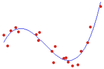
Structure Learning

Taxonomy of score-based algorithms (non-exhaustive)

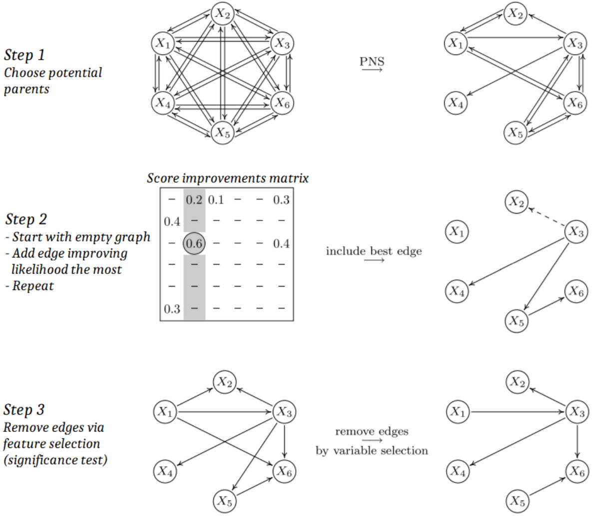
		Discrete optim.	Continuous optim.
	Linear	GES [Chickering, 2003]	NOTEARS [Zheng et al., 2018]
	Nonlinear	CAM [Bühlmann et al., 2014]	GraN-DAG [Lachapelle et al., 2020]

Structure Learning

Taxonomy of score-based algorithms (non-exhaustive)

		Discrete optim.	Continuous optim.
	Linear	GES [Chickering, 2003]	NOTEARS [Zheng et al., 2018]
	Nonlinear	**CAM** **[Bühlmann et al., 2014]**	GraN-DAG [Lachapelle et al., 2020]

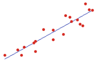
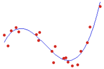
A greedy algorithm - CAM [Bühlmann et al., 2014]



Figures from [Bühlmann et al., 2014]

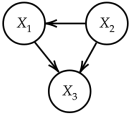
Structure Learning

Taxonomy of score-based algorithms (non-exhaustive)

		Discrete optim.	Continuous optim.
	Linear	GES [Chickering, 2003]	**NOTEARS [Zheng et al., 2018]**
	Nonlinear	CAM [Bühlmann et al., 2014]	GraN-DAG [Lachapelle et al., 2020]

NOTEARS: Continuous optimization for structure learning

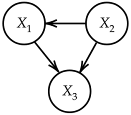
- Encode graph as a **weighted adjacency matrix** $U = [u_1 | \dots | u_d] \in \mathbb{R}^{d \times d}$



$$\underbrace{A = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{Adjacency matrix}} \quad \underbrace{U = \begin{pmatrix} 0 & 0 & 4.8 \\ 0.2 & 0 & -1.7 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{Weighted adjacency matrix}}$$

NOTEARS: Continuous optimization for structure learning

- Encode graph as a **weighted adjacency matrix** $U = [u_1 | \dots | u_d] \in \mathbb{R}^{d \times d}$



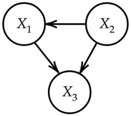
$$\underbrace{A = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{Adjacency matrix}} \qquad \underbrace{U = \begin{pmatrix} 0 & 0 & 4.8 \\ 0.2 & 0 & -1.7 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{Weighted adjacency matrix}}$$

- Represents coefficients in a **linear model**:

$$X_i := u_i^\top X + \texttt{noise}_i \ \ \forall i$$

NOTEARS: Continuous optimization for structure learning

- Encode graph as a **weighted adjacency matrix** $U = [u_1 | \dots | u_d] \in \mathbb{R}^{d \times d}$



$$\underbrace{A = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{Adjacency matrix}} \qquad \underbrace{U = \begin{pmatrix} 0 & 0 & 4.8 \\ 0.2 & 0 & -1.7 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{Weighted adjacency matrix}}$$

- Represents coefficients in a **linear model**:

$$X_i := u_i^\top X + \texttt{noise}_i \ \ \forall i$$

- For an arbitrary U, associated graph might be cyclic

Acyclicity constraint

NOTEARS [Zheng et al., 2018] uses this **differentiable acyclicity constraint**:

$$\operatorname{Tr} e^{U \odot U} - d = 0 \qquad \left(e^M \triangleq \sum_{k=0}^{\infty} \frac{M^k}{k!} \right)$$

Mila Université de Montréal

NOTEARS: Continuous optimization for structure learning

- NOTEARS [Zheng et al., 2018]:
  Solve this **continuous constrained optimization problem**:

$$\max_{U} \underbrace{-\|\mathbf{X} - \mathbf{X}U\|_F^2 - \lambda\|U\|_1}_{\text{Score}} \quad \text{s.t.} \quad \operatorname{Tr} e^{U \odot U} - d = 0$$

- where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the design matrix containing all n samples

NOTEARS: Continuous optimization for structure learning

- NOTEARS [Zheng et al., 2018]:
  Solve this **continuous constrained optimization problem**:

$$\max_{U} \underbrace{-\|\mathbf{X} - \mathbf{X}U\|_F^2 - \lambda\|U\|_1}_{\text{Score}} \quad \text{s.t.} \quad \operatorname{Tr} e^{U \odot U} - d = 0$$

- where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the design matrix containing all n samples
- Solve approximately using an **Augmented Lagrangian method**
- Amounts to maximizing (with gradient ascent)

$$-\|\mathbf{X} - \mathbf{X}U\|_F^2 - \lambda\|U\|_1 - \alpha_t(\operatorname{Tr} e^{U \odot U} - d) - \frac{\mu_t}{2}(\operatorname{Tr} e^{U \odot U} - d)^2$$

- while gradually increasing α_t and μ_t

NOTEARS: The acyclicity constraint

$$\operatorname{Tr} e^{U \odot U} - d = 0 \qquad \left(e^M \triangleq \sum_{k=0}^{\infty} \frac{M^k}{k!} \right)$$

Suppose $A \in \{0,1\}^{d \times d}$ is an adjacency matrix for a certain directed graph

NOTEARS: The acyclicity constraint

$$\mathrm{Tr}\, e^{U \odot U} - d = 0 \qquad \left(e^M \triangleq \sum_{k=0}^{\infty} \frac{M^k}{k!} \right)$$

Suppose $A \in \{0, 1\}^{d \times d}$ is an adjacency matrix for a certain directed graph

$(A^k)_{ii}$ = number of **cycles** of length k passing through i

NOTEARS: The acyclicity constraint

$$\operatorname{Tr} e^{U \odot U} - d = 0 \qquad \left(e^{M} \triangleq \sum_{k=0}^{\infty} \frac{M^k}{k!} \right)$$

Suppose $A \in \{0,1\}^{d \times d}$ is an adjacency matrix for a certain directed graph

$(A^k)_{ii} =$ number of **cycles** of length k passing through i

Graph acyclic $\iff$ $(A^k)_{ii} = 0$ for all i and all k

NOTEARS: The acyclicity constraint

$$\operatorname{Tr} e^{U \odot U} - d = 0 \qquad \left(e^M \triangleq \sum_{k=0}^{\infty} \frac{M^k}{k!} \right)$$

Suppose $A \in \{0, 1\}^{d \times d}$ is an adjacency matrix for a certain directed graph

$(A^k)_{ii} =$ number of **cycles** of length k passing through i

Graph acyclic $\iff (A^k)_{ii} = 0$ for all i and all k

$$\iff \operatorname{Tr}\left[\sum_{k=1}^{\infty} \frac{A^k}{k!}\right] = 0$$

NOTEARS: The acyclicity constraint

$$\text{Tr}\, e^{U \odot U} - d = 0 \qquad \left(e^M \triangleq \sum_{k=0}^{\infty} \frac{M^k}{k!} \right)$$

Suppose $A \in \{0,1\}^{d \times d}$ is an adjacency matrix for a certain directed graph

$(A^k)_{ii}$ = number of **cycles** of length k passing through i

Graph acyclic $\iff (A^k)_{ii} = 0$ for all i and all k

$$\iff \text{Tr}\left[\sum_{k=1}^{\infty} \frac{A^k}{k!}\right] = 0$$

$$\iff \text{Tr}\left[\sum_{k=0}^{\infty} \frac{A^k}{k!} - A^0\right] = 0$$

NOTEARS: The acyclicity constraint

$$\mathrm{Tr}\, e^{U \odot U} - d = 0 \qquad \left(e^M \triangleq \sum_{k=0}^{\infty} \frac{M^k}{k!} \right)$$

Suppose $A \in \{0,1\}^{d \times d}$ is an adjacency matrix for a certain directed graph

$(A^k)_{ii}$ = number of **cycles** of length k passing through i

$$\begin{aligned}
\text{Graph acyclic} &\iff (A^k)_{ii} = 0 \text{ for all } i \text{ and all } k \\
&\iff \mathrm{Tr}\left[\textstyle\sum_{k=1}^{\infty} \frac{A^k}{k!}\right] = 0 \\
&\iff \mathrm{Tr}\left[\textstyle\sum_{k=0}^{\infty} \frac{A^k}{k!} - A^0\right] = 0 \\
&\iff \mathrm{Tr}\, e^A - d = 0
\end{aligned}$$

NOTEARS: The acyclicity constraint

$$\mathrm{Tr}\, e^{U \odot U} - d = 0 \qquad \left(e^M \triangleq \sum_{k=0}^{\infty} \frac{M^k}{k!} \right)$$

Suppose $A \in \{0, 1\}^{d \times d}$ is an adjacency matrix for a certain directed graph

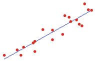
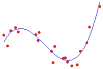
$(A^k)_{ii}$ = number of **cycles** of length k passing through i

$$\begin{aligned} \text{Graph acyclic} \iff & (A^k)_{ii} = 0 \text{ for all } i \text{ and all } k \\ \iff & \mathrm{Tr}\left[\textstyle\sum_{k=1}^{\infty} \frac{A^k}{k!}\right] = 0 \\ \iff & \mathrm{Tr}\left[\textstyle\sum_{k=0}^{\infty} \frac{A^k}{k!} - A^0\right] = 0 \\ \iff & \mathrm{Tr}\, e^A - d = 0 \end{aligned}$$

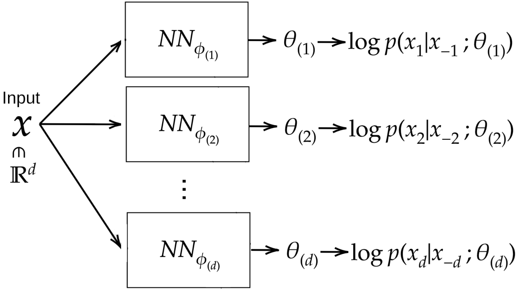
The argument is almost identical when using weighted adjacency U instead of A...

Structure Learning

Taxonomy of score-based algorithms (non-exhaustive)

		Discrete optim.	Continuous optim.
	Linear	GES [Chickering, 2003]	NOTEARS [Zheng et al., 2018]
	Nonlinear	CAM [Bühlmann et al., 2014]	**GraN-DAG** **[Lachapelle et al., 2020]**

Gradient-Based Neural DAG Learning

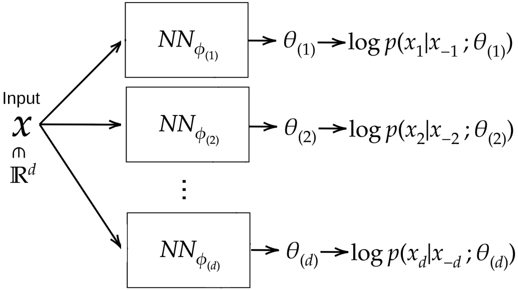


$\phi_{(i)} \triangleq \{W_{(i)}^{(1)}, \dots, W_{(i)}^{(L+1)}\}$

$W_{(i)}^{(\ell)} = \ell$th weight matrix of $NN_{\phi_{(i)}}$

$\phi \triangleq \{\phi_{(i)}\}_{i=1}^{d}$

Gradient-Based Neural DAG Learning



$\phi_{(i)} \triangleq \{W^{(1)}_{(i)}, \dots, W^{(L+1)}_{(i)}\}$

$W^{(\ell)}_{(i)} =$ ℓth weight matrix of $NN_{\phi_{(i)}}$

$\phi \triangleq \{\phi_{(i)}\}_{i=1}^d$

$\prod_{i=1}^d p(x_i|x_{-i};\theta_{(i)})$ does not decompose according to a DAG!

We need to constrain the networks to be acyclic! How?

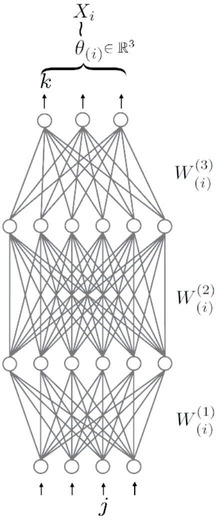
Gradient-Based Neural DAG Learning

Key idea:
Construct a **weighted adjacency matrix** A_ϕ (analogous to U from the linear case) which could be used in the acyclicity constraint

Then maximize likelihood under acyclicity constraint via **augmented Lagrangian**

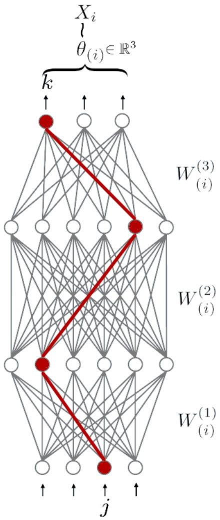
$$\max_{\phi} \underbrace{\mathbb{E}_{X\sim P_X} \sum_{i=0}^{d} \log p_\phi(X_i|X_{-i}) - \alpha_t(\mathrm{Tr}\, e^{A_\phi} - d) - \frac{\mu_t}{2}(\mathrm{Tr}\, e^{A_\phi} - d)^2}_{\text{Augmented Lagrangian}}$$

Constructing weighted adjacency matrix A_ϕ



Let's measure the "strength" of edge $X_j \rightarrow X_i$

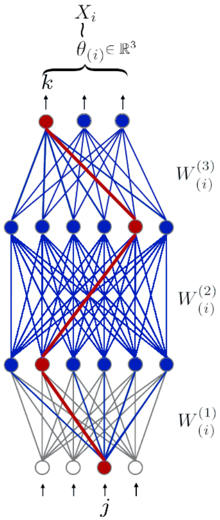
Constructing weighted adjacency matrix A_ϕ



Let's measure the "strength" of edge $X_j \to X_i$

- **Path product:**
  $|W^{(1)}_{h_1 j}||W^{(2)}_{h_2 h_1}||W^{(3)}_{k h_2}| \geq 0$

Constructing weighted adjacency matrix A_ϕ

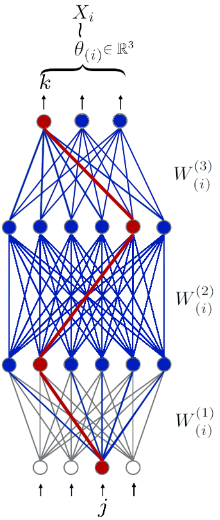


Let's measure the "strength" of edge $X_j \rightarrow X_i$

- **Path product:**
  $|W^{(1)}_{h_1 j}||W^{(2)}_{h_2 h_1}||W^{(3)}_{k h_2}| \geq 0$

- $C \triangleq |W^{(3)}||W^{(2)}||W^{(1)}|$
  "Connection strength" from X_j to $\theta_{(i)}$:
  $\sum_{k=1}^{m} C_{kj} \geq 0$

Constructing weighted adjacency matrix A_ϕ

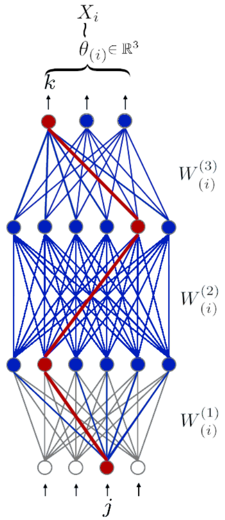


Let's measure the "strength" of edge $X_j \to X_i$

- **Path product:** $|W^{(1)}_{h_1 j}||W^{(2)}_{h_2 h_1}||W^{(3)}_{k h_2}| \geq 0$
- $C \triangleq |W^{(3)}||W^{(2)}||W^{(1)}|$
  "Connection strength" from X_j to $\theta_{(i)}$:
  $\sum_{k=1}^{m} C_{kj} \geq 0$
- $\sum_{k=1}^{m} C_{kj} = 0 \Rightarrow$ All paths from X_j to X_i are **inactive!**

Mila Université de Montréal

Constructing weighted adjacency matrix A_ϕ



Let's measure the "strength" of edge $X_j \to X_i$

- **Path product:** $|W^{(1)}_{h_1 j}||W^{(2)}_{h_2 h_1}||W^{(3)}_{k h_2}| \geq 0$
- $C \triangleq |W^{(3)}||W^{(2)}||W^{(1)}|$
  "Connection strength" from X_j to $\theta_{(i)}$: $\sum_{k=1}^m C_{kj} \geq 0$
- $\sum_{k=1}^m C_{kj} = 0 \Rightarrow$ All paths from X_j to X_i are **inactive!**

$$(A_\phi)_{ji} \triangleq \begin{cases} \sum_{k=1}^m \left(C_{(i)}\right)_{kj}, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases}$$

Gradient-Based Neural DAG Learning

The algorithm:

1. Preliminary neighborhood selection (analogous to CAM)
   i.e. for each node, select potential parents via any *variable selection* approach
2. Maximize likelihood under acyclicity constraint via **augmented Lagrangian**

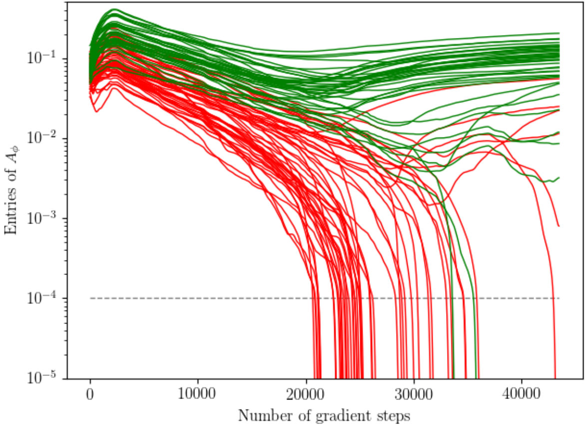
$$\max_{\phi} \underbrace{\mathop{\mathbb{E}}_{X\sim P_X} \sum_{i=0}^{d} \log p_\phi(x_i|x_{-i}) - \alpha_t(\operatorname{Tr} e^{A_\phi} - d) - \frac{\mu_t}{2}(\operatorname{Tr} e^{A_\phi} - d)^2}_{\text{Augmented Lagrangian}}$$

3. DAG Pruning (analogous to CAM)
   i.e. for each node, get rid of some parents via any *variable selection* approach

Step 1 and 3 helps reducing overfitting.
Important since **adding edges cannot reduce maximum likelihood**

Mila Université de Montréal

Gradient-Based Neural DAG Learning



Correct edges
Wrong edges

Experiments

Synthetic data: $X_i | X_{\pi_i^{\mathcal{G}}} \sim \mathcal{N}(f_i(X_{\pi^{\mathcal{G}}}), \sigma_i^2)$ $\quad f_i \sim$ Gaussian Process

Models: GraN-DAG, NOTEARS and CAM makes the Gaussian assumption

Real data: Measurements of expression levels of proteins and phospholipids in human immune system cells [Sachs et al., 2005]

		Synthetic (50 nodes)		Protein data set	
		SHD	SID	SHD	SID
Continuous	**GraN-DAG**	**102.6±21.2**	**1060.1±109.4**	**13**	**47**
	DAG-GNN	191.9±15.2	2146.2±64	16	44
	NOTEARS	202.3±14.3	2149.1±76.3	21	44
Discrete	**CAM**	**98.8±20.7**	**1197.2±125.9**	**12**	**55**
	RANDOM	708.4±234.4	1921.3±203.5	21	60

DAG-GNN [Yu et al., 2019]

Experiments

- In previous setup, synthetic data generation and model matched
- Here: **model misspecification**

Synthetic post nonlinear data sets

		PNL-GP		PNL-MULT	
		SHD	SID	SHD	SID
10 nodes ER1	GraN-DAG	**1.6±3.0**	**3.9±8.0**	**13.1±3.8**	35.7±12.3
	DAG-GNN	11.5±6.8	32.4±19.3	17.900±6.2	40.700±14.743
	NOTEARS	10.7±5.5	34.4±19.1	**14.0±4.0**	38.6±11.9
	CAM	**1.5±2.6**	**6.8±12.1**	**12.0±6.4**	36.3±17.7
	GSF	**6.2±3.3**	[7.7±8.7, 18.9±12.4]	**10.7±3.0**	**[9.8±11.9, 25.3±11.5]**
	RANDOM	23.8±2.9	36.8±19.1	23.7±2.9	37.7±20.7

GSF [Huang et al., 2018a]

Experiments: Effect of sample size

- Previous experiment: relatively small dataset: 1000 examples
- GraN-DAG is more expressive than CAM
- Advantage shows up in large sample size regimes

Effect of sample size - Gauss-ANM 50 nodes ER4 (averaged over 10 datasets)

Sample size	Method	SHD	SID
500	CAM	123.5 ± 13.9	1181.2 ± 160.8
	GraN-DAG	130.2 ± 14.4	1246.4 ± 126.1
1000	CAM	103.7 ± 15.2	1074.7 ± 125.8
	GraN-DAG	104.4 ± 15.3	942.1 ± 69.8
5000	CAM	74.1 ± 13.2	845.0 ± 159.8
	GraN-DAG	71.9 ± 15.9	554.1 ± 117.9
10000	CAM	66.3 ± 16.0	808.1 ± 142.9
	GraN-DAG	65.9 ± 19.8	453.4 ± 171.7

Mila Université de Montréal

GraN-DAG with interventions [Brouillard et al., 2020]

Can we make use of **interventional data**?

GraN-DAG with interventions [Brouillard et al., 2020]

Some **terminology** and **setting**:

- $I \subset \{1, ..., n\}$ is an *interventional target* (set of nodes on which we intervene)

- **Definition** of *stochastic intervention*:

$$p(x_1, ..., x_d | do(X_I)) \triangleq \prod_{j \notin I} p_j(x_j | x_{\pi_j^{\mathcal{G}}}) \prod_{j \in I} \tilde{p}_j(x_j)$$

  where $\tilde{p}_j(x_j)$ is the *new marginal* replacing $p_j(x_j | x_{\pi_j^{\mathcal{G}}})$ (parents are "cut out")

GraN-DAG with interventions [Brouillard et al., 2020]

Some **terminology** and **setting**:

- $I \subset \{1, ..., n\}$ is an *interventional target* (set of nodes on which we intervene)

- **Definition** of *stochastic intervention*:

$$p(x_1, ..., x_d | do(X_I)) \triangleq \prod_{j \notin I} p_j(x_j | x_{\pi_j^{\mathcal{G}}}) \prod_{j \in I} \tilde{p}_j(x_j)$$

  where $\tilde{p}_j(x_j)$ is the *new marginal* replacing $p_j(x_j | x_{\pi_j^{\mathcal{G}}})$ (parents are "cut out")

- **Observed:** $\{(X^{(1)}, I^{(1)}), ..., (X^{(n)}, I^{(n)})\}$ where $I^{(i)}$ is the interventional target associated to observation $X^{(i)}$.

$$I^{(i)} \sim P(I) \text{ i.i.d. } \forall i$$
$$X^{(i)} | I^{(i)} \sim P(X | I = I^{(i)}) \triangleq p(x_1, ..., x_d | do(X_{I^{(i)}})) \; \forall i \quad (1)$$

  where $P(I)$ is a distribution over a collection of interventional targets $\mathcal{I}$

GraN-DAG with interventions [Brouillard et al., 2020]

- Think about a CGM as a family of models of the form

$$\left\{ \prod_{j \notin I} p_j(x_j | x_{\pi_j^{\mathcal{G}}}; \phi_j) \prod_{j \in I} \tilde{p}_j(x_j; \omega_j^I) | I \in \mathcal{I} \right\}$$

  where $\omega^I \triangleq \{\omega_j^I\}_{j \in I}$ for each $I \in \mathcal{I}$ are learnable parameters.

- The natural **optimization problem:**

$$\max_{\phi, \{\omega^I\}_{I \in \mathcal{I}}} \mathbb{E}_{(X,I) \sim P(X,I)} \left[\sum_{j \notin I} \log p_j(X_j | X_{-j}; \phi_j) + \sum_{j \in I} \log p_j(X_j; \omega_j^I) \right] \quad \text{s.t.} \quad \mathrm{Tr}\, e^{A_\phi} = d$$

- But we do not really care about learning the $p_j(X_j; \omega_j^I)$ ...

- ... and problem trivially decomposes as a sum of $\max_\phi$ and $\max_{\{\omega^I\}_{I \in \mathcal{I}}}$ so ...

GraN-DAG with interventions [Brouillard et al., 2020]

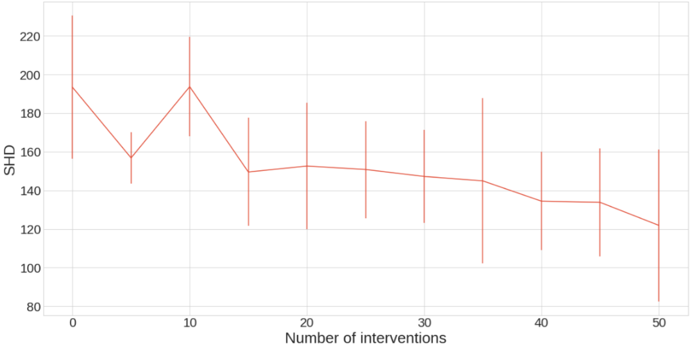
- ... can forget about $p_j(X_j; \omega_j^I)$ altogether and get

The optimization problem:

$$\max_{\phi} \mathbb{E}_{(X,I)\sim P(X,I)} \sum_{j \notin I} \log p(X_j | X_{-j}; \phi_j) \quad \text{s.t.} \quad \mathrm{Tr}\, e^{A_\phi} = d$$

In a nutshell: We throw out the conditionals associated with the intervention variables

GraN-DAG with interventions [Brouillard et al., 2020]



- Linear data (unidentifiable without interventions)
- 50 nodes and $\approx$ 200 edges
- Intervention on one node at a time

GraN-DAG with interventions [Brouillard et al., 2020]

- Nonlinear data

	20 nodes, $e = 1$		20 nodes, $e = 4$		50 nodes, $e = 1$		50 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
GraN-DAG	**1.0** ± 1.2	**1.9** ± 3.7	**33.3** ± 10.0	**138.9** ± 21.0	**85.3** ± 20.2	**885.3** ± 151.4	**3.2** ± 2.8	**12.7** ± 13.6
GraN-DAG no interv	**0.7** ± 0.8	**1.1** ± 2.3	**41.5** ± 8.0	164.2 ± 27.9	109.8 ± 17.6	**1021.7** ± 109.0	**4.1** ± 2.3	**18.6** ± 15.1
GIES	14.6 ± 5.8	34.9 ± 23.1	64.3 ± 6.1	282.7 ± 36.7	170.2 ± 21.1	1820.7 ± 183.0	41.6 ± 9.9	107.9 ± 48.9
CAM*	1.9 ± 2.5	4.8 ± 6.3	48.8 ± 23.5	144.3 ± 38.6	**91.9** ± 11.9	**1024.4** ± 118.0	**4.7** ± 3.8	**24.5** ± 18.1

- Linear data

	20 nodes, $e = 1$		20 nodes, $e = 4$		50 nodes, $e = 1$		50 nodes, $e = 4$	
Method	SHD	SID	SHD	SID	SHD	SID	SHD	SID
GraN-DAG	5.7 ± 5.4	31.3 ± 37.5	**24.5** ± 7.4	**159.6** ± 34.9	12.5 ± 7.2	63.4 ± 36.0	**52.7** ± 16.0	**699.9** ± 166.3
GraN-DAG no interv	17.7 ± 8.2	91.8 ± 71.9	88.3 ± 23.2	275.2 ± 17.0	41.2 ± 7.3	163.8 ± 70.4	193.4 ± 39.3	1667.7 ± 169.2
GIES	**2.4** ± 1.1	**0.0** ± 0.0	**29.7** ± 28.2	**122.3** ± 103.7	13.9 ± 3.8	**0.0** ± 0.0	129.7 ± 75.1	**757.8** ± 477.7
CAM*	6.5 ± 8.7	19.0 ± 35.0	59.2 ± 27.2	173.1 ± 69.2	**1.9** ± 2.4	9.8 ± 16.5	135.4 ± 29.8	1484.1 ± 274.8

- More experiments in workshop paper...

GraN-DAG with Neural Autoregressive flows

- In previous experiments, GraN-DAG models was:

$$X_i = NN_{\phi_i}(X_{\pi_i^{\mathcal{G}}}) + \sigma_i Z \quad \text{with} \quad Z \sim \mathcal{N}(0,1) \ \ \forall i$$

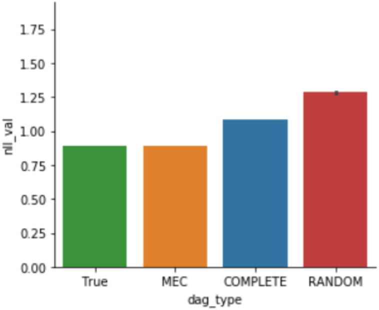
- GraN-DAG's framework allows for usage of "Neural Autoregressive Flows" [Huang et al., 2018b]

$$X_i = NAF(Z; NN_{\phi_i}(X_{\pi_i^{\mathcal{G}}})) \quad \text{with} \quad Z \sim \mathcal{N}(0,1) \ \ \forall i$$

- The function $NAF(\cdot; NN_{\phi_i}(X_{\pi_i^{\mathcal{G}}}))$ is **invertible** and with **tractable Jacobian** so the likelihood of X can be computed exactly and maximized

GraN-DAG with Neural Autoregressive flows

- Without interventions, we run into identifiability problems ...



- **Future work:** make it works with interventional data (since identifiability is less of a problem)

Conclusion and future work

Gradient-based DAG search...

- ... performs similarly to its discrete analogs
- ... scales well with number of samples (since amenable to stochastic optimization)
- ... can be easily adapted to work with interventional data
- ... allows for very expressive density models (Neural Autoregressive flow)

Future work:

- DAGs appear in many places, could we adapt the neural acyclicity constraint to other problems? (Not causality?)
- Drawing links between causality and representation learning

References

Brouillard, P., Drouin, A., Lachapelle, S., Lacoste, A., & Lacoste-Julien, S. (2020).
Gradient-based neural dag learning with interventions.

Bühlmann, P., Peters, J., & Ernest, J. (2014).
CAM: Causal additive models, high-dimensional order search and penalized regression.
Annals of Statistics.

Chickering, D. (2003).
Optimal structure identification with greedy search.
Journal of Machine Learning Research.

Huang, B., Zhang, K., Lin, Y., Schölkopf, B., & Glymour, C. (2018a).
Generalized score functions for causal discovery.
In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*

Huang, C.-W., Krueger, D., Lacoste, A., & Courville, A. (2018b).
Neural autoregressive flows.

Koller, D. & Friedman, N. (2009).
Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning.
MIT Press.

Lachapelle, S., Brouillard, P., Deleu, T., & Lacoste-Julien, S. (2020).
Gradient-based neural dag learning.
In *Proceedings of the Eighth International Conference on Learning Representations (to appear).*

Peters, J., M. Mooij, J., Janzing, D., & Schölkopf, B. (2014).
Causal discovery with continuous additive noise models.
Journal of Machine Learning Research.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D., & Nolan, G. (2005).
Causal protein-signaling networks derived from multiparameter single-cell data.
Science.

Yu, Y., Chen, J., Gao, T., & Yu, M. (2019).
DAG-GNN: DAG structure learning with graph neural networks.
In *Proceedings of the 36th International Conference on Machine Learning.*

Zheng, X., Aragam, B., Ravikumar, P., & Xing, E. (2018).
Dags with no tears: Continuous optimization for structure learning.
In *Advances in Neural Information Processing Systems 31.*


Philippe Brouillard


Alexandre Drouin


Tristan Deleu


Alexandre Lacoste


Simon Lacoste-Julien

Mila Université de Montréal