TECHNISCHE UNIVERSITÄT DARMSTADT

AIML Lab

Winter Semester 2025/26 Lecture

Causality for AI & ML

"Neuro-Causal Models"

Prof. Dr. Kristian Kersting

Moritz Willig

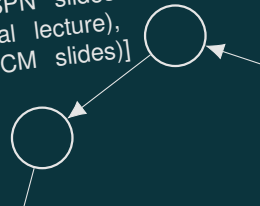
Tim Woydt

Florian Busch

Matej Zečević

Today's speaker

[Rerecording. Thanks to Florian Busch (SPN slides+ giving the original lecture), Matej Zečević (NCM slides)]



Why Neuro-Causal

Pros of classical ML:

1. Applicable to a wide range of problems.
2. Straightforward training process.
3. Easy large-scale adaption.

Why Neuro-Causal

Pros of classical ML:

1. Applicable to a wide range of problems.
2. Straightforward training process.
3. Easy large-scale adaption.

Cons of classical ML:

1. Does not generalize well out-of-distribution.
2. Models often stay on the correlational level.
3. Might include confounding factors in decisions.

Why Neuro-Causal

Pros of classical ML:

1. Applicable to a wide range of problems.
2. Straightforward training process.
3. Easy large-scale adaption.

Cons of classical ML:

1. Does not generalize well out-of-distribution.
2. Models often stay on the correlational level.
3. Might include confounding factors in decisions.

$\rightarrow$ **Can we counter the cons of ML by leveraging causal methods & models?**

Why Neuro-Causal

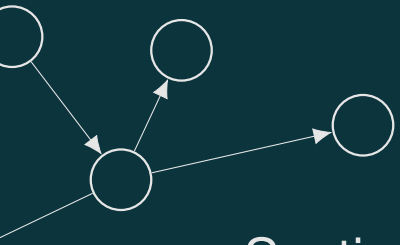
Pros of classical ML:

1. Applicable to a wide range of problems.
2. Straightforward training process.
3. Easy large-scale adaption.

Cons of classical ML:

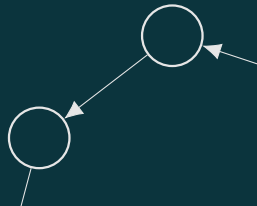
1. Does not generalize well out-of-distribution.
2. Models often stay on the correlational level.
3. Might include confounding factors in decisions.

$\rightarrow$ **Can we counter the cons of ML by leveraging causal methods & models?**
$\leftarrow$ **Can we scale causal methods and models beyond small-scale examples using ML methods?**

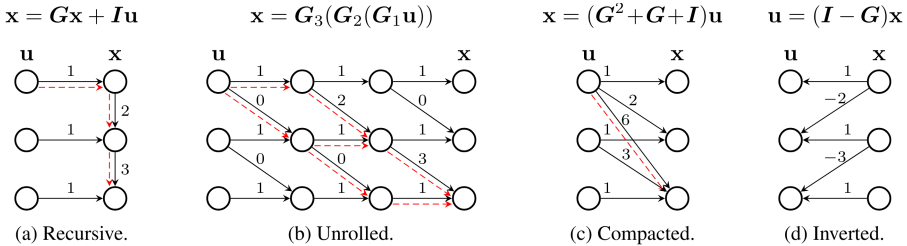


Section 1

Causal Normalizing Flows



SCM Representations



SCM evaluation can be represented in various ways.

Assumption for CNFs: (1) Structural eqs. are diffeomorphisms (‘bijective and smooth’), (2) the model is acyclic and (3) the data is causally sufficient.

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Normalizing Flows

Normalizing Flows (NFs) are *density estimators* that transform distributions into each other, via a series of bijective transforms, $t_\theta^1 \circ \cdots \circ t_\theta^k = T_\theta$.

Every transform t_i needs to

- preserve the probability mass at every step $\forall l \in [1..K]. \int t_\theta^1 \circ \cdots \circ t_\theta^l(P) = 1.0$.
- keep values above (or equal to) 0.

$\rightarrow$ Every step, again, produces a valid density.

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Normalizing Flows

Normalizing Flows (NFs) are *density estimators* that transform distributions into each other, via a series of bijective transforms, $t_\theta^1 \circ \cdots \circ t_\theta^k = T_\theta$.

Every transform t_i needs to

- preserve the probability mass at every step $\forall l \in [1..K]. \int t_\theta^1 \circ \cdots \circ t_\theta^l(P) = 1.0$.
- keep values above (or equal to) 0.

$\rightarrow$ Every step, again, produces a valid density.

Commonly, distributions are transformed into a distribution of independent Gaussians: $T_\theta(\mathbf{x}) =: \mathbf{u} \sim P_\mathbf{u}$ where, e.g., $P_\mathbf{u} = \bigtimes_i \mathbb{N}_i(0, 1)$

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Normalizing Flows

Normalizing Flows (NFs) are *density estimators* that transform distributions into each other, via a series of bijective transforms, $t_\theta^1 \circ \cdots \circ t_\theta^k = T_\theta$.

Every transform t_i needs to

- preserve the probability mass at every step $\forall l \in [1..K]. \int t_\theta^1 \circ \cdots \circ t_\theta^l(P) = 1.0$.
- keep values above (or equal to) 0.

$\rightarrow$ Every step, again, produces a valid density.

Commonly, distributions are transformed into a distribution of independent Gaussians: $T_\theta(\mathbf{x}) =: \mathbf{u} \sim P_\mathbf{u}$ where, e.g., $P_\mathbf{u} = \bigtimes_i \mathbb{N}_i(0,1)$

More generally: $\log p(\mathbf{x}) = \log p(T_\theta(\mathbf{x})) + \log|det(\nabla_\mathbf{x} T_\theta(\mathbf{x}))|$
with parameters θ learned via MLE.

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Auto Regressive NFs

We are interested in Auto Regressive NFs (ANFs) that reconstruct the data.
We not only have a single NF process, but one per variable $i \in [1..N]$.

$$z_i^l := \tau_i^l(z_i^{l-1}; \mathbf{h}_i^l) \text{ where } \mathbf{h}_i^l := c_i^l(\mathbf{z}_{1:i-1}^{l-1}) \tag{1}$$

- The 'transformer' (τ) is strictly monotonic.
- The 'conditioner' ($\mathbf{h}$) only takes previous $\mathbf{z}_{1:i-1}^{l-1}$ values.

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Auto Regressive NFs

We are interested in Auto Regressive NFs (ANFs) that reconstruct the data.
We not only have a single NF process, but one per variable $i \in [1..N]$.

$$z_i^l := \tau_i^l(z_i^{l-1}; \mathbf{h}_i^l) \text{ where } \mathbf{h}_i^l := c_i^l(\mathbf{z}_{1:i-1}^{l-1}) \tag{1}$$

- The 'transformer' (τ) is strictly monotonic.
- The 'conditioner' ($\mathbf{h}$) only takes previous $\mathbf{z}_{1:i-1}^{l-1}$ values.

Every process only depends on it own previous-layer value, z_i^{l-1}, and that of the processes computed 'before' it, $\mathbf{z}_{1:i-1}^{l-1}$.

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Auto Regressive NFs

We are interested in Auto Regressive NFs (ANFs) that reconstruct the data.
We not only have a single NF process, but one per variable $i \in [1..N]$.

$$z_i^l := \tau_i^l(z_i^{l-1}; \mathbf{h}_i^l) \text{ where } \mathbf{h}_i^l := c_i^l(\mathbf{z}_{1:i-1}^{l-1}) \tag{1}$$

- The 'transformer' (τ) is strictly monotonic.
- The 'conditioner' ($\mathbf{h}$) only takes previous $\mathbf{z}_{1:i-1}^{l-1}$ values.

Every process only depends on it own previous-layer value, z_i^{l-1}, and that of the processes computed 'before' it, $\mathbf{z}_{1:i-1}^{l-1}$.

Insight: This looks just like an unrolled/(compacted) SCM.
The 'only' requirement is a valid causal order!

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Causal NFs

Desired properties of a Causal Normalizing Flow (CNF):

1. Capture observational distribution.
2. Isolate true exogenous variables.
3. Causally consistent with the true SCM.

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Causal NFs

Desired properties of a Causal Normalizing Flow (CNF):

1. Capture observational distribution.
2. Isolate true exogenous variables.
3. Causally consistent with the true SCM.

Quick aside: Triangular monotonic increasing maps (TMI)

$$f(x) = \begin{bmatrix} f_1(x_1) \\ \vdots \\ f_N(x_1, \ldots, x_k) \end{bmatrix}, \qquad \text{and} \qquad \delta_{x_i} f_i(x_1, x_2, \ldots, x_i) \geq 0$$

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Causal NFs

Desired properties of a Causal Normalizing Flow (CNF):

1. Capture observational distribution.
2. Isolate true exogenous variables.
3. Causally consistent with the true SCM.

Quick aside: Triangular monotonic increasing maps (TMI)

$$f(x) = \begin{bmatrix} f_1(x_1) \\ \vdots \\ f_N(x_1, \ldots, x_k) \end{bmatrix}, \qquad \text{and} \qquad \delta_{x_i} f_i(x_1, x_2, \ldots, x_i) \geq 0$$

- By construction ANFs are TMIs.
- It is always possible to "monotonize" the SCM
  (using the Knöthe-Rosenblatt (KR) transport. Details in the paper.)

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Causal NFs - Identifiability

Desired properties of a Causal Normalizing Flow (CNF):

1. Capture observational distribution.
2. Isolate true exogenous variables.
3. Causally consistent with the true SCM.

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Causal NFs - Identifiability

Desired properties of a Causal Normalizing Flow (CNF):

1. Capture observational distribution.
2. Isolate true exogenous variables.
3. Causally consistent with the true SCM.

Identifiability: Authors prove that **u** can be identified up to a component-wise transformation. [Thm. 1 in the paper]

Consistency: If **u** can be identified CNFs are consistent with the SCM [Corollary 2 in the paper]

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Causal NFs - Consistency

Desired properties of a Causal Normalizing Flow (CNF):

1. Capture observational distribution.
2. Isolate true exogenous variables.
3. Causally consistent with the true SCM.

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Causal NFs - Consistency

Desired properties of a Causal Normalizing Flow (CNF):

1. Capture observational distribution.
2. Isolate true exogenous variables.
3. Causally consistent with the true SCM.
   - Meaning truthful to interventions and CFs.

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

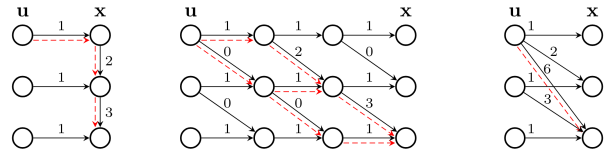
Causal NFs - Consistency

Desired properties of a Causal Normalizing Flow (CNF):

1. Capture observational distribution.
2. Isolate true exogenous variables.
3. Causally consistent with the true SCM.
   - Meaning truthful to interventions and CFs.

Interventions require changing structural equations. However, we learned to estimate an unrolled/compacted graph. $\rightarrow$ How do we apply interventions?

$\mathbf{x} = \boldsymbol{G}\mathbf{x} + \boldsymbol{I}\mathbf{u}$ $\quad$ $\mathbf{x} = \boldsymbol{G}_3(\boldsymbol{G}_2(\boldsymbol{G}_1\mathbf{u}))$ $\quad$ $\mathbf{x} = (\boldsymbol{G}^2 + \boldsymbol{G} + \boldsymbol{I})\mathbf{u}$



Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Causal NFs - Interventions Idea

NFs, T_θ, consist of bijective transforms and are therefore invertible, T_θ^{-1}.

Approach:

1. (Sample form **u** to obtain an observational $\mathbf{x} \leftarrow T_\theta^{-1}(\mathbf{u})$).
2. Intervene on the particular variable $X_i := c$.
3. Map back *only the i-th value*, $u_i \leftarrow T_\theta(\mathbf{x})_i$.
4. Generate **x** from the 'intervened' **u** distribution.

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

Causal NFs - Interventions

Algorithm 1 Algorithm to sample from the interventional distribution, $P(\mathbf{x} \mid do(\mathrm{x}_i = \alpha))$.

function SAMPLEINTERVENEDDIST(i, α)
  $\mathbf{u} \sim P_{\mathbf{u}}$
  $\mathbf{x} \leftarrow T_{\boldsymbol{\theta}}^{-1}(\mathbf{u})$ ▷ Sample a value from the observational distribution.
  $\mathrm{x}_i \leftarrow \alpha$ ▷ Set x_i to the intervened value α.
  $\mathrm{u}_i \leftarrow T_{\boldsymbol{\theta}}(\mathbf{x})_i$ ▷ Change the i-th value of $\mathbf{u}$.
  $\mathbf{x} \leftarrow T_{\boldsymbol{\theta}}^{-1}(\mathbf{u})$
  return $\mathbf{x}$ ▷ Return the intervened sample.
end function

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.

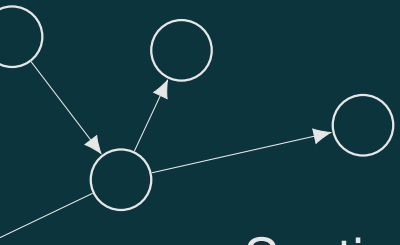
Causal NFs - Counterfactuals

Having a model that allows us to infer **u** from some observation and intervene on the system directly allows us to compute counterfactuals:

Algorithm 2 Algorithm to sample from the counterfactual distribution, $P(\mathbf{x}^{\mathrm{cf}} \mid do(\mathrm{x}_i = \alpha), \mathbf{x}^{\mathrm{f}})$.

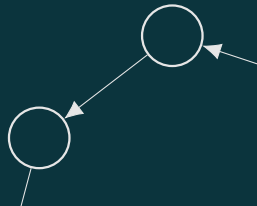
function GETCOUNTERFACTUAL($\mathbf{x}^{\mathrm{f}}$, i, α)
  $\mathbf{u} \leftarrow T_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{f}})$ ▷ Get $\mathbf{u}$ from the factual sample.
  $\mathrm{x}_i^{\mathrm{f}} \leftarrow \alpha$ ▷ Set x_i to the intervened value α.
  $\mathrm{u}_i \leftarrow T_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{f}})_i$ ▷ Change the i-th value of $\mathbf{u}$.
  $\mathbf{x}^{\mathrm{cf}} \leftarrow T_{\boldsymbol{\theta}}^{-1}(\mathbf{u})$
  return $\mathbf{x}^{\mathrm{cf}}$ ▷ Return the counterfactual value.
end function

Javaloy, Adrián, Pablo Sánchez-Martín, and Isabel Valera. "Causal normalizing flows: from theory to practice." Advances in Neural Information Processing Systems 36 (2023): 58833-58864.



Section 2

Causal VAE



CausalVAE

Observation: VAEs are good at capturing image contents in low-dim. latent spaces.

Idea: Constraint the latent space $\mathbf{Z} = [Z_1, \ldots, Z_n]$ to some causal model.

Expectation: Latent factors might capture high-level concepts.

CausalVAE

Observation: VAEs are good at capturing image contents in low-dim. latent spaces.

Idea: Constraint the latent space $\mathbf{Z} = [Z_1, \ldots, Z_n]$ to some causal model.

Expectation: Latent factors might capture high-level concepts.

Consideration: Latent factors in a VAE are commonly trained to have independent Gaussian distributions.

$\rightarrow$ This is usually not the case for variables in an SCM.
(remember: $P(X_i \mid pa(X_i))$).

CausalVAE

Observation: VAEs are good at capturing image contents in low-dim. latent spaces.

Idea: Constraint the latent space $\mathbf{Z} = [Z_1, \ldots, Z_n]$ to some causal model.

Expectation: Latent factors might capture high-level concepts.

Consideration: Latent factors in a VAE are commonly trained to have independent Gaussian distributions.

$\rightarrow$ This is usually not the case for variables in an SCM.
(remember: $P(X_i \mid pa(X_i))$).

$\rightarrow$ Interventions, e.g. setting some latent value to $z_i := c$, don't propagate their effects to their causal children.

CausalVAE

Observation: VAEs are good at capturing image contents in low-dim. latent spaces.

Idea: Constraint the latent space $\mathbf{Z} = [Z_1, \ldots, Z_n]$ to some causal model.

Expectation: Latent factors might capture high-level concepts.

Consideration: Latent factors in a VAE are commonly trained to have independent Gaussian distributions.

→ This is usually not the case for variables in an SCM.
(remember: $P(X_i \mid pa(X_i))$).

→ Interventions, e.g. setting some latent value to $z_i := c$, don't propagate their effects to their causal children.

This is a similar setting as for CNFs!

→ **Solution:** Learn to predict the exogenous terms **u** and apply equations afterwards.

CausalVAE

Process:

- Image $x \rightarrow$ Encoder $\rightarrow$ Exogenous Noise (ϵ)
- Exogenous Noise (ϵ) $\rightarrow$ Causal Layer $\rightarrow$ Causal Variables ($\mathbf{z}$)
- Causal Variables ($\mathbf{z}$) $\rightarrow$ Decoder $\rightarrow$ Reconstruction $\hat{x}$.

Yang, Mengyue, et al. "Causalvae: Disentangled representation learning via neural structural causal models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

CausalVAE

Assume the process to be a linear Structural Equation Model (SEM):

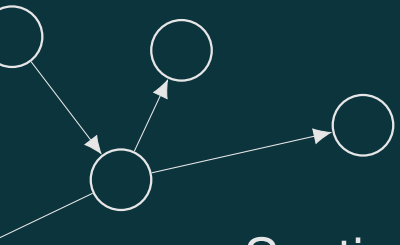
$$\mathbf{z} = A^T \mathbf{z} + \epsilon$$

A is a learnable weighted adjacency matrix, representing the graph. A mask is applied (e.g. multiply A by the binary adj matrix) to enforce the correct causal graph.

Solving for $\mathbf{z}$ gives:

$$\mathbf{z} = (I - A^T)^{-1} \epsilon$$

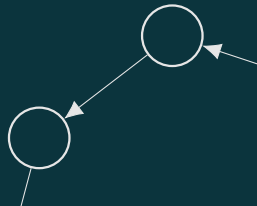
Given the reconstructed $\mathbf{z}$, we can simply decode the image again: $\hat{\mathbf{x}} := d(\mathbf{z})$.



Section

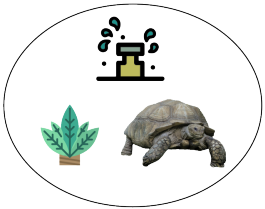
3

Causal SPNs



Motivation for SPNs

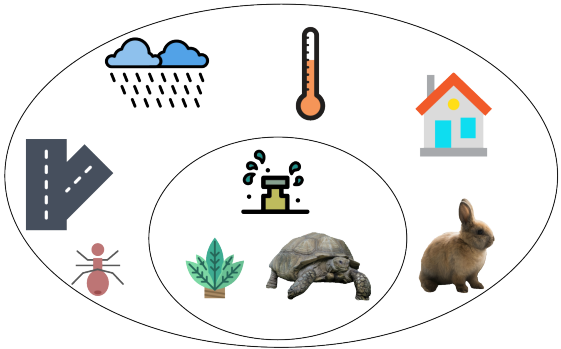
Let's say we want to model a small part of an ecosystem, and we want to find out how we can help a population of, e.g., turtles.



A small problem can be modeled without much trouble.

Motivation for SPNs

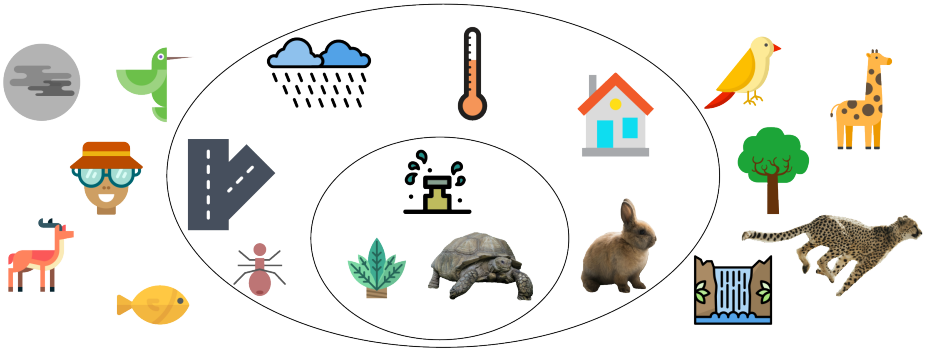
Let's say we want to model a small part of an ecosystem, and we want to find out how we can help a population of, e.g., turtles.



If more variables are included, run time starts being important.

Motivation for SPNs

Let's say we want to model a small part of an ecosystem, and we want to find out how we can help a population of, e.g., turtles.



At a certain problem size, some models become intractable.

Tractable Inference

Full evidence query, e.g.:

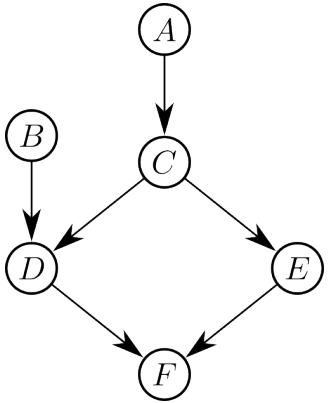
$$P(A = a, B = b, C = c, D = d, E = e, F = f)$$

Marginal probability query, e.g.:

$$P(D = d) = \sum_{a,b,c,e,f} P(D = d|a, b, c, e, f)$$

Conditional probability query, e.g.:

$$P(F = f|D = d) = \frac{P(F = f, D = d)}{P(D = d)}$$



Tractable Inference

Full evidence query, e.g.:

$$P(A = a, B = b, C = c, D = d, E = e, F = f)$$

easy and fast

Marginal probability query, e.g.:

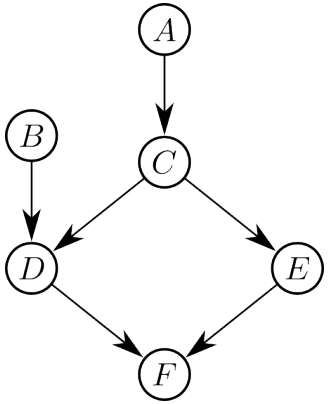
$$P(D = d) = \sum_{a,b,c,e,f} P(D = d | a, b, c, e, f)$$

very costly

Conditional probability query, e.g.:

$$P(F = f | D = d) = \frac{P(F = f, D = d)}{P(D = d)}$$

same as for marginals



Sum-Product Networks (SPNs)

Graphical models.

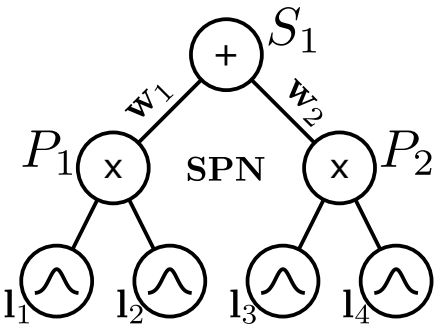
Calculate joint probability distribution.

Tractable marginal inference.

Leaf node: Base distributions.

Product node: Independences.

Sum node: Mixture distributions.

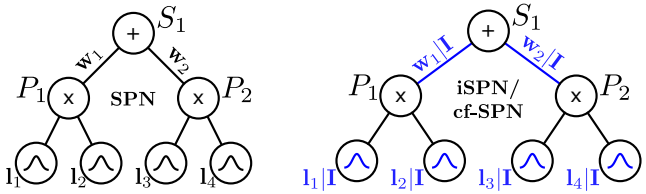


Poon, Hoifung, and Pedro Domingos. "Sum-product networks: a new deep architecture." Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence. 2011.

Interventional Sum-Product Network (iSPN)

SPNs can not compute causal queries without modifications.

iSPN strategy: parametrize SPN parameter such that they can adapt to the interventional setting **I**.

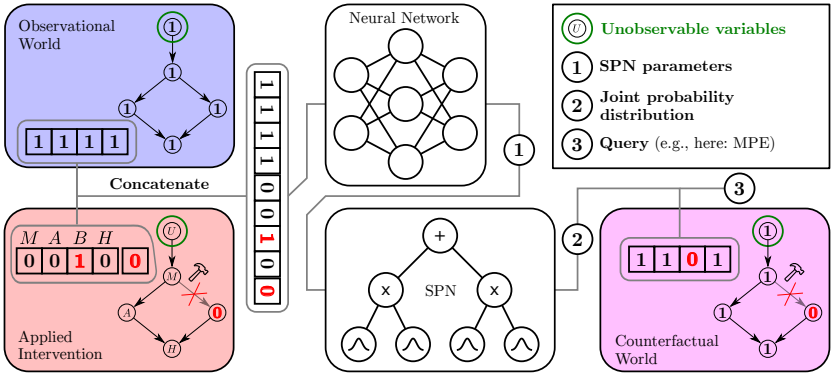


A neural network is used to learn the mapping between **I** and the SPN parameters.

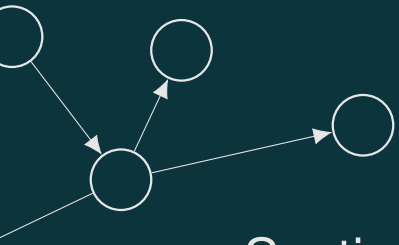
Zečević, Matej, et al. "Interventional sum-product networks: Causal inference with tractable probabilistic models." Advances in neural information processing systems 34 (2021): 15019-15031.

Counterfactual Sum-Product Network (cf-SPN)

For counterfactuals, the input must specify the full counterfactual setting (i.e., the original world / evidence and the counterfactual intervention).



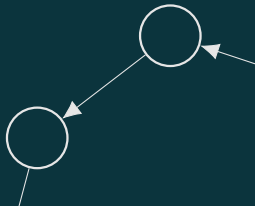
Busch, Florian Peter, et al. "Structural Causal Circuits: Probabilistic Circuits Climbing All Rungs of Pearl's Ladder of Causation." Transactions on Machine Learning Research, 2025, https://openreview.net/forum?id=25XyUTICdZ.



Section

4

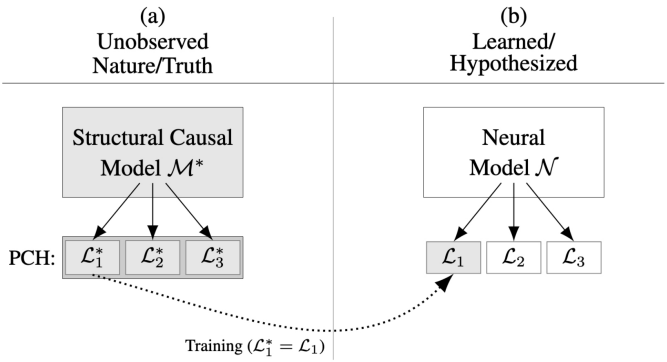
Neural Causal Models



NCM

First proper formalization by “The Neural-Causal Connection” paper.

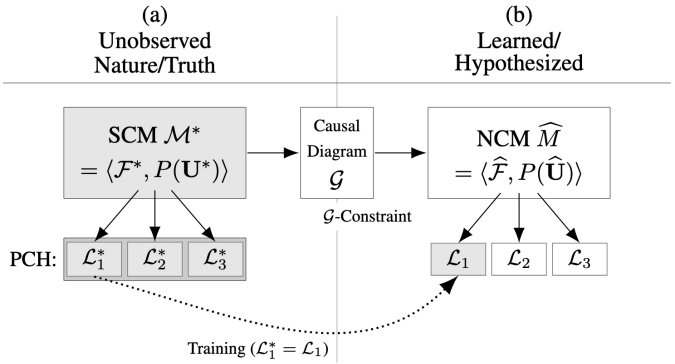
Motivation: neural models seem unable to learn causality.



K. Xia et al. “The Causal-Neural Connection: Expressiveness, Learnability, and Inference.” NeurIPS 2021.

Solution is to use structure as an inductive bias

Use the graph to build an overarching model consisting of neural net modules



K. Xia et al. "The Causal-Neural Connection: Expressiveness, Learnability, and Inference." NeurIPS 2021.

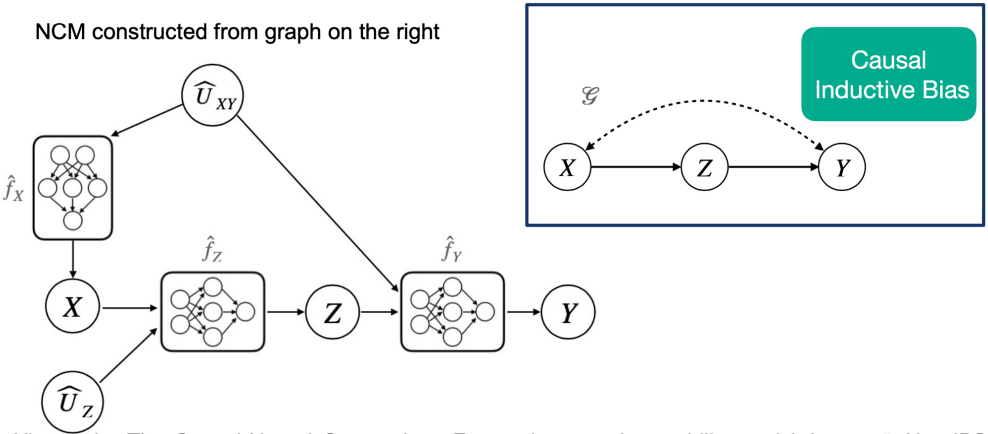
How to build an NCM, a definition:

We define a Neural Causal Model (for short, NCM) $\widehat{M}(\theta)$ over variables V with parameters $\theta = \{\theta_{V_i} : V_i \in V\}$ as an SCM $\langle \widehat{U}, V, \widehat{\mathcal{F}}, \widehat{P}(\widehat{U}) \rangle$ such that

- $\widehat{U} \subseteq \{\widehat{U}_C : C \subseteq V\}$, where each $\widehat{U}$ is associated with some subset of variables $C \subseteq V$, and $\mathcal{D}_{\widehat{U}} = [0, 1]$ for all $\widehat{U} \in \widehat{U}$. (Unobserved confounding is present whenever $|C| > 1$.)
- $\widehat{\mathcal{F}} = \{\widehat{f}_{V_i} : V_i \in V\}$, where each $\widehat{f}_{V_i}$ is a feedforward neural network parameterized by $\theta_{V_i} \in \theta$ mapping values of $U_{V_i} \cup \mathrm{Pa}_{V_i}$ to values of V_i for some $\mathrm{Pa}_{V_i} \subseteq V$ and $U_{V_i} = \{\widehat{U}_C : \widehat{U}_C \in \widehat{U},\ V_i \in C\}$;
- $\widehat{P}(\widehat{U})$ is defined s.t. $\widehat{U} \sim \mathrm{Unif}(0, 1)$ for each $\widehat{U} \in \widehat{U}$.

K. Xia et al. "The Causal-Neural Connection: Expressiveness, Learnability, and Inference." NeurIPS 2021.

Example with 3 exogenous vars. and 1 known hidden confounder



K. Xia et al. "The Causal-Neural Connection: Expressiveness, Learnability, and Inference." NeurIPS 2021.

How to learn an NCM, an algorithm:

Input: causal query $Q = P(y \mid do(x))$, L_1 data $P(v)$, and causal diagram $\mathcal{G}$
Output: $P^{\mathcal{M}^*}(y \mid do(x))$ if identifiable, FAIL otherwise

1. $\widehat{M} \leftarrow \mathrm{NCM}(V, \mathcal{G})$ // from Def. on previous slide
2. $\theta^*_{\min} \leftarrow \arg\min_\theta P^{\widehat{M}(\theta)}(y \mid do(x))$ s.t. $L_1(\widehat{M}(\theta)) = P(v)$
3. $\theta^*_{\max} \leftarrow \arg\max_\theta P^{\widehat{M}(\theta)}(y \mid do(x))$ s.t. $L_1(\widehat{M}(\theta)) = P(v)$
4. **if** $P^{\widehat{M}(\theta^*_{\min})}(y \mid do(x)) \neq P^{\widehat{M}(\theta^*_{\max})}(y \mid do(x))$ **then**
5. **return** FAIL
6. **else**
7. **return** $P^{\widehat{M}(\theta^*_{\min})}(y \mid do(x))$ // choose min or max arbitrarily

K. Xia et al. "The Causal-Neural Connection: Expressiveness, Learnability, and Inference." NeurIPS 2021.

Approximating the optimization problem (binary variables)

Input: Data $\{v_k\}_{k=1}^n$, variables V, $X \subseteq V$, $x \in \mathcal{D}_X$, $Y \subseteq V$, $y \in \mathcal{D}_Y$, causal diagram $\mathcal{G}$, number of Monte Carlo samples m, regularization constant λ, learning rate η

1. Initialize $\widehat{M} \leftarrow \mathrm{NCM}(V, \mathcal{G})$ (NCM parameters $\theta_{\max}$)
2. **for** $k \leftarrow 1$ **to** n **do** // $\tilde{\sigma}_{v_i} = f$ if $v_i = 1$, else $1 - f$ where $f := \sigma(\phi_{\theta_{max},i}(pa_{V_i}, u^c_{V_i}))$

   $\widehat{q}_{\max} \leftarrow 0, \quad \widehat{p}_{\max} \leftarrow \frac{1}{m} \sum_{j=1}^{m} \prod_{V_i \in V} \tilde{\sigma}_{v_i}(\theta_{\max})$
3. **for** $v \in \mathcal{D}_V$ **do**

   if v and y match on corresp. vars. **then**

   $\widehat{q}_{\max} \leftarrow \widehat{q}_{\max} + \frac{1}{m} \sum_{j=1}^{m} \prod_{V_i \in V \setminus X} \tilde{\sigma}_{v_i}(\theta_{\max})$
4. $\mathcal{L}_{\max} \leftarrow -\log \widehat{p}_{\max} \; - \; \lambda \log \widehat{q}_{\max}$
5. $\theta_{\max} \leftarrow \theta_{\max} + \eta \nabla \mathcal{L}_{\max}$

Above details line 3 from the previous algorithm, correspondingly for $\theta_{\min}$

K. Xia et al. "The Causal-Neural Connection: Expressiveness, Learnability, and Inference." NeurIPS 2021.

Takeaways

Since using neural nets as your modelling class for SCM structural equations does not break the SCM's definition, we have that NCMs are SCMs. Mathematically, $\exists\, \widehat{M}(\theta)$ s.t. $\mathcal{L}(\widehat{M}(\theta)) = \mathcal{L}(M^*)$ where $\mathcal{L}(\cdot)$ returns the PCH, that is, all quantities derivable from any level $\mathcal{L}_i$ with $i \in \{1, 2, 3\}$.

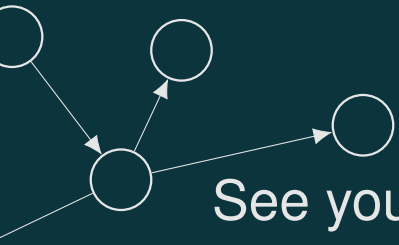
Their equivalence then again assures us that if a causal effect can be estimated with just $\mathcal{L}_1$ data and the graph structure $\mathcal{G}$, then we can in principle do it practically with NCMs.

Optimizing for θ_{min} and θ_{max} when sufficiently satisfying the $(\mathcal{L}_1, \mathcal{G})$-constraints then guarantees that we can in principle estimate the desired true causal effect stemming from M^* (**warning:** this does not imply that the NCM then *equals* M^*, only on a quantity of $\mathcal{L}_2$ and not necessarily all of $\mathcal{L}_2$ or $\mathcal{L}_3$).

NCMs are an instance of SCMs & there are many more

For a discussion of ideas around different modelling schemes when it comes to SCMs themselves and their neural instance classes such as NCM, iSPN etc. you might want to consider:

Blom et al. “Beyond SCMs: Causal Constraints Models.” UAI 2019.
Zečević et al. “Not All Causal Inference is the Same.” TMLR 2023.



See you next week!