Score matching enables causal discovery of nonlinear additive noise models

P. Rolland, V. Cevher, M. Kleindessner, C. Russel, B. Schölkopf, D. Janzing, F. Locatello

ICML 2022

Introduction

- Classical ML task : "If I observe $X = x$, what do I expect the distribution of Y to be ?"

$\rightarrow$ requires approximating the distribution of $Y|X = x$.

Introduction

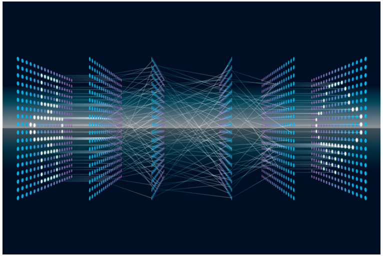
• Classical ML task : "If I observe $X = x$, what do I expect the distribution of Y to be ?"
$\rightarrow$ requires approximating the distribution of $Y|X = x$.

• Causal task : "If I observe $(X, Y) = (x, y)$ and modify the value of X to x', what do I expect the new distribution of Y to be ?"
$\rightarrow$ requires to know the process underlying the generation of the observed data.

Introduction

• Classical ML task : "If I observe $X = x$, what do I expect the distribution of Y to be ?"
$\rightarrow$ requires approximating the distribution of $Y|X = x$.

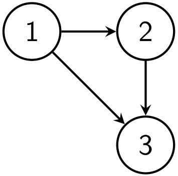
• Causal task : "If I observe $(X, Y) = (x, y)$ and modify the value of X to x', what do I expect the new distribution of Y to be ?"
$\rightarrow$ requires to know the process underlying the generation of the observed data.



Structural Causal Model

We want to know the details of the generative model underlying the observed data : The **Structural Causal Model**.

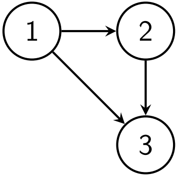
$$
\begin{aligned}
X_1 &\leftarrow \xi_1 \\
X_2 &\leftarrow f_2(X_1, \xi_2) \\
X_3 &\leftarrow f_3(X_1, X_2, \xi_3) \\
&\dots
\end{aligned}
$$



Structural Causal Model

We want to know the details of the generative model underlying the observed data : The **Structural Causal Model**.

$$
\begin{aligned}
X_1 &\leftarrow \xi_1 \\
X_2 &\leftarrow f_2(X_1, \xi_2) \\
X_3 &\leftarrow f_3(X_1, X_2, \xi_3) \\
&\dots
\end{aligned}
$$



In general, various SCM's can lead to the same joint p.d.f. !

We need either

- Interventional data (hard to obtain !)
- Assumptions on the generative model

Non-linear additive Gaussian noise model

Consider the following non-linear additive Gaussian noise SCM :

$$X_i \leftarrow f_i(\mathrm{pa}_i(X)) + \xi_i, \tag{1}$$

where $\xi_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_i^2)$, $\mathrm{pa}_i(X)$ selects the coordinates of X which are parents of node i in some DAG and the functions f_i's are **non-linear**.

Non-linear additive Gaussian noise model

Consider the following non-linear additive Gaussian noise SCM :

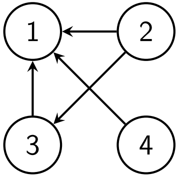
$$X_i \leftarrow f_i(\mathrm{pa}_i(X)) + \xi_i, \tag{1}$$

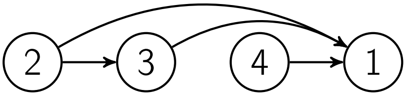
where $\xi_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_i^2)$, $\mathrm{pa}_i(X)$ selects the coordinates of X which are parents of node i in some DAG and the functions f_i's are **non-linear**.

Goal : Using observational data, recover the DAG associated with (1).

The search for the causal DAG can be broken into 2 parts :

1. Find a *topological order* (**our focus**).
2. Estimate the functions f_i's in the SCM, in particular their dependence on *previous* variables.

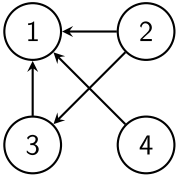


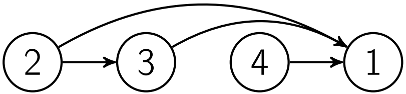


Ordering-based causal discovery

The search for the causal DAG can be broken into 2 parts :

1. Find a *topological order* (**our focus**).
2. Estimate the functions f_i's in the SCM, in particular their dependence on *previous* variables.





Topological order search : We need a way to sequentially identify a leaf in the graph.

Data distribution under AGN model

Additive Gaussian noise model :

$$X_i = f_i(\mathrm{pa}_i(X)) + \xi_i, \quad \xi_i \overset{indep.}{\sim} \mathcal{N}(0, \sigma_i^2)$$

We can write the data probability density function p as

$$p(x) = \prod_{i=1}^{d} p(x_i | \mathrm{pa}_i(x))$$

$$\log p(x) = -\frac{1}{2} \sum_{i=1}^{d} \left(\frac{x_i - f_i(\mathrm{pa}_i(x))}{\sigma_i} \right)^2 - \frac{1}{2} \sum_{i=1}^{d} \log(2\pi\sigma_i^2).$$

Data distribution under AGN model

Additive Gaussian noise model :

$$X_i = f_i(\mathrm{pa}_i(X)) + \xi_i, \quad \xi_i \overset{indep.}{\sim} \mathcal{N}(0, \sigma_i^2)$$

We can write the data probability density function p as

$$p(x) = \prod_{i=1}^{d} p(x_i|\mathrm{pa}_i(x))$$

$$\log p(x) = -\frac{1}{2}\sum_{i=1}^{d}\left(\frac{x_i - f_i(\mathrm{pa}_i(x))}{\sigma_i}\right)^2 - \frac{1}{2}\sum_{i=1}^{d}\log(2\pi\sigma_i^2).$$

The score function of a distribution with density p is defined as $s(x) \equiv \nabla \log p(x)$. For the AGN model, we have

$$s_j(x) \equiv \frac{\partial \log p}{\partial x_j}(x) = -\frac{x_j - f_j(\mathrm{pa}_j(x))}{\sigma_j^2} + \sum_{i \in \mathrm{children}(j)} \frac{\partial f_i}{\partial x_j}(\mathrm{pa}_i(x))\frac{x_i - f_i(\mathrm{pa}_i(x))}{\sigma_i^2}$$

Score function under AGN model

$$s_j(x) = -\frac{x_j - f_j(\mathrm{pa}_j(x))}{\sigma_j^2} + \sum_{i \in \mathrm{children}(j)} \frac{\partial f_i}{\partial x_j}(\mathrm{pa}_i(x)) \frac{x_i - f_i(\mathrm{pa}_i(x))}{\sigma_i^2}$$

Score function under AGN model

$$s_j(x) = -\frac{x_j - f_j(\mathrm{pa}_j(x))}{\sigma_j^2} + \sum_{i \in \mathrm{children}(j)} \frac{\partial f_i}{\partial x_j}(\mathrm{pa}_i(x)) \frac{x_i - f_i(\mathrm{pa}_i(x))}{\sigma_i^2}$$

Observations :

- j is a leaf $\Rightarrow \frac{\partial s_j}{\partial x_j}(x) = -\frac{1}{\sigma_j^2}$ independent of x, i,e., $\mathrm{Var}_X \left[\frac{\partial s_j(X)}{\partial x_j}\right] = 0$.

Score function under AGN model

$$s_j(x) = -\frac{x_j - f_j(\mathrm{pa}_j(x))}{\sigma_j^2} + \sum_{i \in \mathrm{children}(j)} \frac{\partial f_i}{\partial x_j}(\mathrm{pa}_i(x)) \frac{x_i - f_i(\mathrm{pa}_i(x))}{\sigma_i^2}$$

Observations :

- j is a leaf $\Rightarrow \frac{\partial s_j}{\partial x_j}(x) = -\frac{1}{\sigma_j^2}$ independent of x, i,e., $\mathrm{Var}_X\left[\frac{\partial s_j(X)}{\partial x_j}\right] = 0$.
- If j is a leaf, i is a parent of $j \Rightarrow s_j(x)$ depends on x_i, i.e., $\mathrm{Var}_X\left[\frac{\partial s_j(x)}{\partial x_i}\right] \neq 0$

Score function under AGN model

$$s_j(x) = -\frac{x_j - f_j(\mathrm{pa}_j(x))}{\sigma_j^2} + \sum_{i \in \mathrm{children}(j)} \frac{\partial f_i}{\partial x_j}(\mathrm{pa}_i(x)) \frac{x_i - f_i(\mathrm{pa}_i(x))}{\sigma_i^2}$$

Observations :

- j is a leaf $\Leftrightarrow \frac{\partial s_j}{\partial x_j}(x) = -\frac{1}{\sigma_j^2}$ independent of x, i,e., $\mathrm{Var}_X \left[\frac{\partial s_j(X)}{\partial x_j}\right] = 0$.
- If j is a leaf, i is a parent of $j \Leftrightarrow s_j(x)$ depends on x_i, i.e., $\mathrm{Var}_X \left[\frac{\partial s_j(x)}{\partial x_i}\right] \neq 0$

Hence, all the information about the leaves is hidden in the Jacobian of s.

SCORE

Input : Data matrix $X \in \mathbb{R}^{n \times d}$.
Initialize $\pi = []$, nodes $= \{1, \ldots, d\}$
for $k = 1, \ldots, d$ **do**
    Estimate the score function $s_{nodes} = \nabla \log p_{nodes}$
    Estimate $V_j = \mathrm{Var}_{X_{nodes}} \left[\frac{\partial s_j(X)}{\partial x_j} \right]$
    $l \leftarrow \text{nodes}[\arg\min_j V_j]$
    $\pi \leftarrow [l, \pi]$
    nodes $\leftarrow$ nodes $- \{l\}$
    Remove l-th column of X
Get the final DAG by pruning the full DAG associated with the topological order π

Learning the score function

We need to estimate the Jacobian of the score function $\nabla s(x) = \nabla_{xx} \log p(x)$, at least at the score at sample points $\{\nabla s(x_i)\}_{i=1,\dots,n}$.

Learning the score function

We need to estimate the Jacobian of the score function $\nabla s(x) = \nabla_{xx} \log p(x)$, at least at the score at sample points $\{\nabla s(x_i)\}_{i=1,\ldots,n}$.

- Method 1 : Score matching [4]

$$\begin{aligned} \min_{\theta} \frac{1}{2}\mathbb{E}_{x\sim p}[\|\nabla_x \log p(x) - s_\theta(x)\|^2] \quad &\text{(Fisher divergence)} \\ = \mathbb{E}_{x\sim p}\left[\frac{1}{2}\|s_\theta(x)\|^2 + Tr(\nabla_x s_\theta(x))\right] & \\ \simeq \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\|s_\theta(x_i)\|^2 + Tr(\nabla_x s_\theta(x_i))\right) & \end{aligned}$$

Requires retraining a neural network after each node removal...

We need to estimate the Jacobian of the score function $\nabla s(x) = \nabla_{xx} \log p(x)$, at least at the score at sample points $\{\nabla s(x_i)\}_{i=1,\dots,n}$.

- Method 1 : Score matching [4]

$$
\begin{aligned}
&\min_{\theta} \frac{1}{2}\mathbb{E}_{x\sim p}[\|\nabla_x \log p(x) - s_\theta(x)\|^2] \quad \text{(Fisher divergence)}\\
&= \mathbb{E}_{x\sim p}\left[\frac{1}{2}\|s_\theta(x)\|^2 + Tr(\nabla_x s_\theta(x))\right]\\
&\simeq \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\|s_\theta(x_i)\|^2 + Tr(\nabla_x s_\theta(x_i))\right)
\end{aligned}
$$

Requires retraining a neural network after each node removal...

- Method 2 : Stein estimators

Stein Gradient Estimator [6]

If $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^{d'}$ is such that $\lim_{\mathbf{x} \to \infty} \mathbf{h}(\mathbf{x}) p(\mathbf{x}) = 0$, then

$$\mathbb{E}_p[\mathbf{h}(\mathbf{x}) \nabla \log p(\mathbf{x})^T + \nabla \mathbf{h}(\mathbf{x})] = 0 \quad \text{(Stein identity)}$$

Stein Gradient Estimator [6]

If $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^{d'}$ is such that $\lim_{\mathbf{x}\to\infty} \mathbf{h}(\mathbf{x})p(\mathbf{x}) = 0$, then

$$\mathbb{E}_p[\mathbf{h}(\mathbf{x})\nabla \log p(\mathbf{x})^T + \nabla\mathbf{h}(\mathbf{x})] = 0 \quad \text{(Stein identity)}$$

$$-\frac{1}{n}\sum_{k=1}^{n}\mathbf{h}(\mathbf{x}^k)\nabla \log p(\mathbf{x}^k)^T + \text{err} = \frac{1}{n}\sum_{k=1}^{n}\nabla\mathbf{h}(\mathbf{x}^k)$$

$$-\frac{1}{n}\mathbf{H}\mathbf{G} + \text{err} = \overline{\nabla\mathbf{h}}$$

where $\text{err} \xrightarrow{n\to\infty} 0$. Let $\mathbf{H} = (\mathbf{h}(\mathbf{x}^1), \ldots, \mathbf{h}(\mathbf{x}^n)) \in \mathbb{R}^{d'\times n}$, $\overline{\nabla\mathbf{h}} = \frac{1}{n}\sum_{k=1}^{n}\nabla\mathbf{h}(\mathbf{x}^k)$ and $\mathbf{G} = (\nabla \log p(\mathbf{x}^1), \ldots, \nabla \log p(\mathbf{x}^n))$.

Stein Gradient Estimator [6]

If $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^{d'}$ is such that $\lim_{\mathbf{x}\to\infty} \mathbf{h}(\mathbf{x})p(\mathbf{x}) = 0$, then

$$\mathbb{E}_p[\mathbf{h}(\mathbf{x})\nabla \log p(\mathbf{x})^T + \nabla \mathbf{h}(\mathbf{x})] = 0 \quad \text{(Stein identity)}$$

$$-\frac{1}{n}\sum_{k=1}^{n} \mathbf{h}(\mathbf{x}^k)\nabla \log p(\mathbf{x}^k)^T + \text{err} = \frac{1}{n}\sum_{k=1}^{n} \nabla \mathbf{h}(\mathbf{x}^k)$$

$$-\frac{1}{n}\mathbf{H}\mathbf{G} + \text{err} = \overline{\nabla \mathbf{h}}$$

where err $\xrightarrow{n\to\infty}$ 0. Let $\mathbf{H} = (\mathbf{h}(\mathbf{x}^1), \dots, \mathbf{h}(\mathbf{x}^n)) \in \mathbb{R}^{d' \times n}$, $\overline{\nabla \mathbf{h}} = \frac{1}{n}\sum_{k=1}^{n} \nabla \mathbf{h}(\mathbf{x}^k)$ and $\mathbf{G} = (\nabla \log p(\mathbf{x}^1), \dots, \nabla \log p(\mathbf{x}^n))$.

$$\hat{\mathbf{G}}^{\text{Stein}} \equiv \arg\min_{\hat{\mathbf{G}}} \|\overline{\nabla \mathbf{h}} + \frac{1}{n}\mathbf{H}\hat{\mathbf{G}}\|_F^2 + \frac{\eta}{n^2}\|\hat{\mathbf{G}}\|_F^2$$

If $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^{d'}$ is such that $\lim_{\mathbf{x}\to\infty} \mathbf{h}(\mathbf{x})p(\mathbf{x}) = 0$, then

$$\mathbb{E}_p[\mathbf{h}(\mathbf{x})\nabla \log p(\mathbf{x})^T + \nabla\mathbf{h}(\mathbf{x})] = 0 \quad \text{(Stein identity)}$$

$$-\frac{1}{n}\sum_{k=1}^{n} \mathbf{h}(\mathbf{x}^k)\nabla \log p(\mathbf{x}^k)^T + \text{err} = \frac{1}{n}\sum_{k=1}^{n} \nabla\mathbf{h}(\mathbf{x}^k)$$

$$-\frac{1}{n}\mathbf{H}\mathbf{G} + \text{err} = \overline{\nabla\mathbf{h}}$$

where err $\stackrel{n\to\infty}{\longrightarrow}$ 0. Let $\mathbf{H} = (\mathbf{h}(\mathbf{x}^1), \ldots, \mathbf{h}(\mathbf{x}^n)) \in \mathbb{R}^{d'\times n}$, $\overline{\nabla\mathbf{h}} = \frac{1}{n}\sum_{k=1}^{n} \nabla\mathbf{h}(\mathbf{x}^k)$ and $\mathbf{G} = (\nabla \log p(\mathbf{x}^1), \ldots, \nabla \log p(\mathbf{x}^n))$.

$$\begin{aligned}\hat{\mathbf{G}}^{\text{Stein}} &\equiv \arg\min_{\hat{\mathbf{G}}} \|\overline{\nabla\mathbf{h}} + \frac{1}{n}\mathbf{H}\hat{\mathbf{G}}\|_F^2 + \frac{\eta}{n^2}\|\hat{\mathbf{G}}\|_F^2 \\ &= -(\mathbf{K} + \eta\mathbf{I})^{-1}\langle\nabla, \mathbf{K}\rangle,\end{aligned}$$

$\mathbf{K}_{ij} = \kappa(\mathbf{x}^i, \mathbf{x}^j) \equiv \mathbf{h}(x^i)^T\mathbf{h}(\mathbf{x}^j)$, $\langle\nabla, \mathbf{K}\rangle_{ij} = \sum_{k=1}^{n} \nabla_{x_j^k}\kappa(\mathbf{x}^i, \mathbf{x}^k)$.

Stein Hessian Estimator

If $q : \mathbb{R}^d \to \mathbb{R}$ is such that $\lim_{\mathbf{x}\to\infty} q(\mathbf{x})p(\mathbf{x}) = 0$, then

$$\mathbb{E}[q(\mathbf{x})p(\mathbf{x})^{-1}\nabla^2 p(\mathbf{x})] = \mathbb{E}[\nabla^2 q(\mathbf{x})] \quad \text{(SSI : Second-order Stein identity)}$$

Stein Hessian Estimator

If $q : \mathbb{R}^d \to \mathbb{R}$ is such that $\lim_{\mathbf{x}\to\infty} q(\mathbf{x})p(\mathbf{x}) = 0$, then

$$\mathbb{E}[q(\mathbf{x})p(\mathbf{x})^{-1}\nabla^2 p(\mathbf{x})] = \mathbb{E}[\nabla^2 q(\mathbf{x})] \quad \text{(SSI : Second-order Stein identity)}$$
$$\mathbb{E}[q(\mathbf{x})\nabla^2 \log p(\mathbf{x})] = \mathbb{E}[\nabla^2 q(\mathbf{x}) - q(\mathbf{x})\nabla \log p(\mathbf{x})\nabla \log p(\mathbf{x})^T].$$

Stein Hessian Estimator

If $q : \mathbb{R}^d \to \mathbb{R}$ is such that $\lim_{\mathbf{x}\to\infty} q(\mathbf{x})p(\mathbf{x}) = 0$, then

$$\mathbb{E}[q(\mathbf{x})p(\mathbf{x})^{-1}\nabla^2 p(\mathbf{x})] = \mathbb{E}[\nabla^2 q(\mathbf{x})] \quad (\text{SSI : Second-order Stein identity})$$
$$\mathbb{E}[q(\mathbf{x})\nabla^2 \log p(\mathbf{x})] = \mathbb{E}[\nabla^2 q(\mathbf{x}) - q(\mathbf{x})\nabla \log p(\mathbf{x})\nabla \log p(\mathbf{x})^T].$$

We only need to approximate $\{\mathrm{diag}(\nabla^2 \log p(\mathbf{x}^i))\}_{i=1,\dots,n}$. Hence, applying the diagonal part of (SSI) for various test functions gathered in $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^{d'}$, we have

$$\mathbb{E}[\mathbf{h}(\mathbf{x})\mathrm{diag}(\nabla^2 \log p(\mathbf{x}))^T] = \mathbb{E}[\nabla^2_{\mathrm{diag}}\mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{x})\mathrm{diag}(\nabla \log p(\mathbf{x})\nabla \log p(\mathbf{x})^T)]$$

where $(\nabla^2_{\mathrm{diag}}\mathbf{h}(\mathbf{x}))_{ij} = \frac{\partial^2 h_i(\mathbf{x})}{\partial x_j^2}$.

Stein Hessian Estimator

Let $\mathbf{J} \equiv (\mathrm{diag}(\nabla^2 \log p(\mathbf{x}^1)), \ldots, \mathrm{diag}(\nabla^2 \log p(\mathbf{x}^n)))^T \in \mathbb{R}^{n \times d}$. Approximating the expectations by an empirical average, we obtain

$$\frac{1}{n}\mathbf{H}\mathbf{J} + \mathrm{err} = \overline{\nabla^2_{\mathrm{diag}}\mathbf{h}} - \mathbf{H}\mathrm{diag}(\mathbf{G}\mathbf{G}^T). \tag{2}$$

where $\overline{\nabla^2_{\mathrm{diag}}\mathbf{h}} = \frac{1}{n}\sum_{k=1}^{n} \nabla^2_{\mathrm{diag}}\mathbf{h}(\mathbf{x}^k)$.

Stein Hessian Estimator

Let $\mathbf{J} \equiv (\mathrm{diag}(\nabla^2 \log p(\mathbf{x}^1)), \ldots, \mathrm{diag}(\nabla^2 \log p(\mathbf{x}^n)))^T \in \mathbb{R}^{n \times d}$. Approximating the expectations by an empirical average, we obtain

$$\frac{1}{n}\mathbf{H}\mathbf{J} + \mathrm{err} = \overline{\nabla^2_{\mathrm{diag}}\mathbf{h}} - \mathbf{H}\mathrm{diag}(\mathbf{G}\mathbf{G}^T). \tag{2}$$

where $\overline{\nabla^2_{\mathrm{diag}}\mathbf{h}} = \frac{1}{n}\sum_{k=1}^{n} \nabla^2_{\mathrm{diag}}\mathbf{h}(\mathbf{x}^k)$. We thus approximate $\mathbf{J}$ by

$$\hat{\mathbf{J}}^{\mathrm{Stein}} \equiv \arg\min_{\hat{\mathbf{J}}} \left\| \frac{1}{n}\mathbf{H}\hat{\mathbf{J}} + \frac{1}{n}\mathbf{H}\mathrm{diag}\left(\hat{\mathbf{G}}^{\mathrm{Stein}}\left(\hat{\mathbf{G}}^{\mathrm{Stein}}\right)^T\right) - \overline{\nabla^2_{\mathrm{diag}}\mathbf{h}} \right\|_F^2 + \frac{\eta}{n^2}\|\hat{\mathbf{J}}\|_F^2$$

Stein Hessian Estimator

Let $\mathbf{J} \equiv (\mathrm{diag}(\nabla^2 \log p(\mathbf{x}^1)), \ldots, \mathrm{diag}(\nabla^2 \log p(\mathbf{x}^n)))^T \in \mathbb{R}^{n \times d}$. Approximating the expectations by an empirical average, we obtain

$$\frac{1}{n}\mathbf{H}\mathbf{J} + \mathrm{err} = \overline{\nabla^2_{\mathrm{diag}}\mathbf{h}} - \mathbf{H}\mathrm{diag}(\mathbf{G}\mathbf{G}^T). \tag{2}$$

where $\overline{\nabla^2_{\mathrm{diag}}\mathbf{h}} = \frac{1}{n}\sum_{k=1}^{n} \nabla^2_{\mathrm{diag}}\mathbf{h}(\mathbf{x}^k)$. We thus approximate $\mathbf{J}$ by

$$\begin{aligned}\hat{\mathbf{J}}^{\mathrm{Stein}} &\equiv \arg\min_{\hat{\mathbf{J}}} \left\| \frac{1}{n}\mathbf{H}\hat{\mathbf{J}} + \frac{1}{n}\mathbf{H}\mathrm{diag}\left(\hat{\mathbf{G}}^{\mathrm{Stein}}\left(\hat{\mathbf{G}}^{\mathrm{Stein}}\right)^T\right) - \overline{\nabla^2_{\mathrm{diag}}\mathbf{h}} \right\|_F^2 + \frac{\eta}{n^2}\|\hat{\mathbf{J}}\|_F^2 \\ &= -\mathrm{diag}\left(\hat{\mathbf{G}}^{\mathrm{Stein}}\left(\hat{\mathbf{G}}^{\mathrm{Stein}}\right)^T\right) + (\mathbf{K} + \eta \mathbf{I})^{-1}\langle \nabla^2_{\mathrm{diag}}, \mathbf{K}\rangle\end{aligned}$$

where $\langle \nabla^2_{\mathrm{diag}}, \mathbf{K}\rangle = n\mathbf{H}^T\overline{\nabla^2_{\mathrm{diag}}\mathbf{h}}$.

Numerical experiments : Synthetic data

We generate synthetic data from AGN model as follows :

1. Sample a DAG in dimension $d = 10, 20, 50$ whose skeleton is an Erdös-Renyi graph with d (ER1) or $4d$ (ER4) edges.
2. The variances σ_i^2 are sampled i.i.d uniformly in $[0.4, 0.8]$.
3. The link functions are sampled from Gaussian processes with lengthscale 1.

We run SCORE using Stein estimator with RBF kernel, and using CAM for pruning the final DAG.

We compute various metrics on the resulting graphs, i.e., SHD, SID and D_{top} defined as

$$D_{top}(\pi, A) = \sum_{i=1}^{d} \sum_{j:\pi_i > \pi_j} A_{ij}.$$

Synthetic data in $d = 10$

ER1	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{1.1 \pm 0.9}$	$\mathbf{4.5 \pm 5.3}$	$\mathbf{0.4 \pm 0.6}$
CAM [1]	1.7 ± 1.0	6.4 ± 4.2	$\mathbf{0.4 \pm 0.5}$
GraN-DAG [5]	1.5 ± 1.4	6.5 ± 7.2	–
SELF [2]	8.4 ± 1.6	32.5 ± 7.6	–
GES [3]	7.8 ± 2.7	32.5 ± 13.6	–
VarSort	–	–	1.9 ± 1.1

ER4	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{19.5 \pm 2.9}$	$\mathbf{35.0 \pm 9.1}$	$\mathbf{0.3 \pm 0.3}$
CAM	24.4 ± 3.1	45.2 ± 10.2	4.4 ± 3.2
GraN-DAG	22.2 ± 2.6	42.0 ± 6.2	–
SELF	37.2 ± 2.1	83.0 ± 5.2	–
GES	34.3 ± 3.0	78.9 ± 6.0	–
VarSort	–	–	9.7 ± 3.1

Synthetic data in $d = 20$

ER1	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{2.6 \pm 1.9}$	$\mathbf{9.9 \pm 8.5}$	1.2 ± 1.7
CAM	3.5 ± 1.6	14.3 ± 9.8	$\mathbf{0.8 \pm 1.0}$
GraN-DAG	7.6 ± 4.2	31.6 ± 22.7	–
SELF	16.6 ± 2.1	89.9 ± 31.2	–
GES	17.7 ± 3.8	77.3 ± 30.5	–
VarSort	–	–	3.7 ± 1.6

ER4	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{47.5 \pm 4.5}$	$\mathbf{177.5 \pm 11.6}$	$\mathbf{3.1 \pm 1.5}$
CAM	54.2 ± 5.4	201.9 ± 29.0	13.6 ± 6.9
GraN-DAG	49.3 ± 4.5	211.4 ± 36.6	–
SELF	75.5 ± 1.6	336.8 ± 31.2	–
GES	67.4 ± 6.1	322.9 ± 21.7	–
VarSort	–	–	18.3 ± 6.7

Synthetic data in $d = 50$

ER1	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	10.4 ± 3.9	$\mathbf{50.9 \pm 32.9}$	3.9 ± 2.4
CAM	$\mathbf{8.3 \pm 2.9}$	53.7 ± 31.9	–
GraN-DAG	20.2 ± 6.1	135.3 ± 45.9	–
SELF	45.4 ± 3.5	326.6 ± 74.3	–
GES	50.5 ± 4.2	233.5 ± 60.8	–
VarSort	–	–	8.8 ± 3.0

ER4	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{131.5 \pm 7.5}$	$\mathbf{1262 \pm 110}$	16.3 ± 6.1
CAM	140.8 ± 5.5	1337 ± 94	–
GraN-DAG	140.8 ± 9.5	1432 ± 110	–
SELF	192.7 ± 3.2	2097 ± 103	–
GES	182.9 ± 7.3	2003 ± 105	–
VarSort	–	–	43.3 ± 9.7

Synthetic data in $d = 20$ with **Laplace noise**

ER1	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{1.6 \pm 1.2}$	$\mathbf{6.8 \pm 11.4}$	0.5 ± 0.9
CAM	2.3 ± 1.4	10.0 ± 7.0	$\mathbf{0.3 \pm 0.5}$
GraN-DAG	4.9 ± 2.1	27.5 ± 13.2	–
SELF	16.4 ± 3.6	87.5 ± 32.3	–
GES	17.7 ± 6.8	72.6 ± 25.5	–
VarSort	–	–	3.4 ± 2.0

ER4	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{48.0 \pm 4.0}$	$\mathbf{199.8 \pm 21.4}$	$\mathbf{4.9 \pm 1.8}$
CAM	52.4 ± 3.9	208.7 ± 17.5	11.6 ± 7.9
GraN-DAG	48.2 ± 3.8	198.3 ± 42.8	–
SELF	77.4 ± 2.2	349.5 ± 19.0	–
GES	69.7 ± 7.1	325.5 ± 28.3	–
VarSort	–	–	20.8 ± 4.5

Computation time comparison

	$d = 10$	$d = 20$	$d = 50$
SCORE order	3.3 ± 0.1	8.5 ± 0.8	31 ± 2.9
SCORE	6.3 ± 0.2	32.7 ± 6.7	257 ± 17
CAM	30.1 ± 3.7	313 ± 80	$1143 \pm 79^{(*)}$
GraN-DAG	185 ± 26	357 ± 47	1410 ± 73

Table – Run time in seconds on ER1. $(*)$ For scalability, we restricted the maximum number of neighbours to 20

Numerical experiments : Real data

- Sachs [7] : Real dataset 11 nodes, 17 edges, 853 observations

- SynTRen [8] : Pseudo-real dataset generating simulated gene expression data. We generated 10 datasets containing 500 samples coming from a 20 nodes graph.

	Sachs		SynTReN	
	SHD	SID	SHD	SID
SCORE	**12**	**45**	36.2 ± 4.7	193.4 ± 60.2
CAM	**12**	55	40.5 ± 6.8	$\mathbf{152.3 \pm 48.0}$
GraN-DAG	13	47	$\mathbf{34.0 \pm 8.5}$	161.7 ± 53.4

Conclusion

- **Learning the data score function allows to recover the causal graph in additive Gaussian noise model.**

Conclusion

- **Learning the data score function allows to recover the causal graph in additive Gaussian noise model.**
- Questions for future work :
  - How to extend this approach to other identifiable models ?

Conclusion

- **Learning the data score function allows to recover the causal graph in additive Gaussian noise model.**
- Questions for future work :
  - How to extend this approach to other identifiable models ?
  - Can we implement score matching efficiently for this setup ?

References I

Peter Bühlmann, Jonas Peters, and Jan Ernest.
Cam : Causal additive models, high-dimensional order search and penalized regression.
The Annals of Statistics, 42(6) :2526–2556, 2014.

Ruichu Cai, Jie Qiao, Zhenjie Zhang, and Zhifeng Hao.
Self : structural equational likelihood framework for causal discovery.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

David Maxwell Chickering.
Optimal structure identification with greedy search.
Journal of machine learning research, 3(Nov) :507–554, 2002.

Aapo Hyvärinen and Peter Dayan.
Estimation of non-normalized statistical models by score matching.
Journal of Machine Learning Research, 6(4), 2005.

References II

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien.
Gradient-based neural dag learning.
arXiv preprint arXiv :1906.02226, 2019.

Yingzhen Li and Richard E Turner.
Gradient estimators for implicit models.
arXiv preprint arXiv :1705.07107, 2017.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan.
Causal protein-signaling networks derived from multiparameter single-cell data.
Science, 308(5721) :523–529, 2005.

Tim Van den Bulcke, Koenraad Van Leemput, Bart Naudts, Piet van Remortel, Hongwu Ma, Alain Verschoren, Bart De Moor, and Kathleen Marchal.
Syntren : a generator of synthetic gene expression data for design and analysis of structure learning algorithms.
BMC bioinformatics, 7(1) :1–12, 2006.