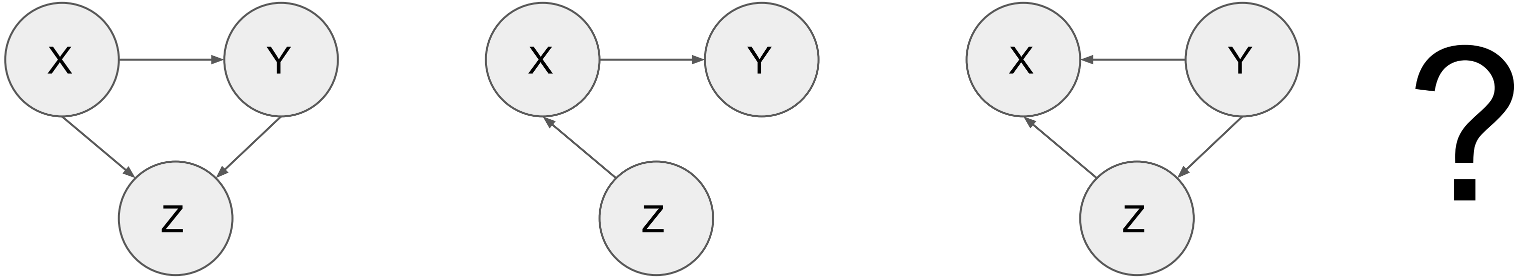
Score matching enables causal discovery of nonlinear additive noise models

Paul Rolland, Volkan Cevher, Matthäus Kleindessner, Chris Russel, Bernhard Schölkopf, Dominik Janzing, Francesco Locatello

Causal discovery

- Finding the Directed Acyclic Graph (DAG) underlying the generative process of the data from observational data;
- Causal discovery from observational data is unidentifiable. Several generative models and causal structures can produce the same observational distribution;
- Additional assumptions are necessary to make the model identifiable.



Different categories of causal discovery methods

- Traditional constraint-based and score-based models:
  - Use d-separation and conditional independence tests to infer causal structure up to MEC;
  - Pros: Guaranteed convergence*; Generality;
  - Cons: Requires ability to perform independence tests; strong assumptions; does not work for 2 vars;
  - Examples: PC algorithm; FCI; GES.

- Non-Gaussian or non-linear methods based on SCMs.
  - Use statistical independence between vars and noise to determine causal direction;
  - Noise will be independent from cause but not from effect;
  - Pros: Works for two vars; non-linear models; includes non-Gaussian linear models and non-linear additive noise models;
  - Cons: Computational cost, still several unidentifiable scenarios;

Additive noise models

- Under some additional assumptions on the link functions, additive noise models are identifiable.
- An additive noise model SCM is given by:

$$X_i = f_i(\mathrm{pa}_i(X)) + \epsilon_i,$$

  where a random variable is a (non-linear) function of its parents plus an additive noise term.

- The model is identifiable from observational data. It is possible to recover the DAG underlying the generative model from the joint probability distribution of *X*.

Order-based methods

Searching over DAGs difficult because:

1. The size of the set of DAGs, which grows super-exponentially with the number of nodes;
2. The acyclicity constraint.

Order-based methods tackle the problem in two phases.

1. Find a topological ordering of the nodes, such that a node in the ordering can be a parent only of the nodes appearing after it in the same ordering.
2. The graph is constructed respecting the topological ordering and pruning spurious edges.

Proposed method

The proposed method is order-based;

The topological order is estimated using an approximation of the distribution's score function (gradient of the log-likelihood);

For a non-linear additive Gaussian noise model, it is possible to identify leaves of the graph using the observational score;

By sequentially identifying the leaves of the graph, and removing the identified leafs, we can obtain the topological order with a time complexity linear in the number of nodes;

Classical pruning techniques can then be used in order to obtain the final graph.

Score matching (1)

The score function is the gradient of the log-likelihood.

$$s(x) \equiv \nabla \log p(x)$$

The zero of the score function is the maximum of the log-likelihood.

The goal of score matching is to learn the score function of a distribution with density p(x) given an i.i.d. samples $\{x^k\}_{k=1,\dots,n}$

The goal is to approximate the score function at the sample points:

$$\mathbf{G} \equiv (\nabla \log p(x^1), \dots \nabla \log p(x^n))^T \in \mathbb{R}^{n \times d}$$

Score matching (2)

Using the Stein gradient estimator, proposed in “Gradient estimators for implicit models” Yingzhen Li & Richard E. Turner, we can directly estimate the score function of the implicitly defined distribution:

The math is too involved to present here, but check the reference if interested.

It is basically a type of kernel density estimation method but for the score function instead of the density function.

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

Jacobian Approximation

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

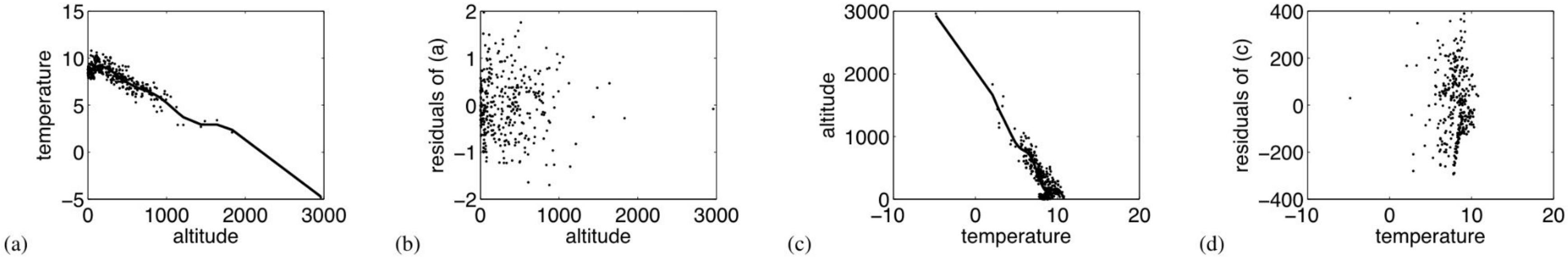
$$= -\text{diag}\left(\hat{\mathbf{G}}^{\text{Stein}}\left(\hat{\mathbf{G}}^{\text{Stein}}\right)^T\right) + (\mathbf{K} + \eta\mathbf{I})^{-1}\langle\nabla^2_{\text{diag}}, \mathbf{K}\rangle,$$

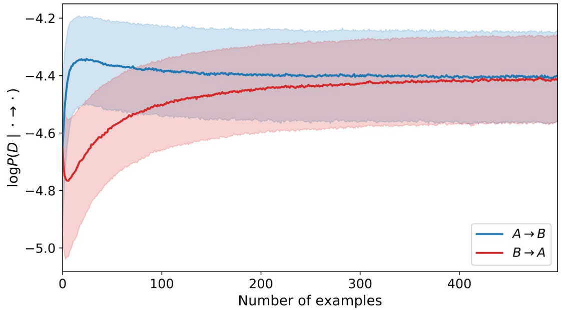
- Based on the kernel choice, the jacobian and gradient computation is simplified
- Using RBF kernel:

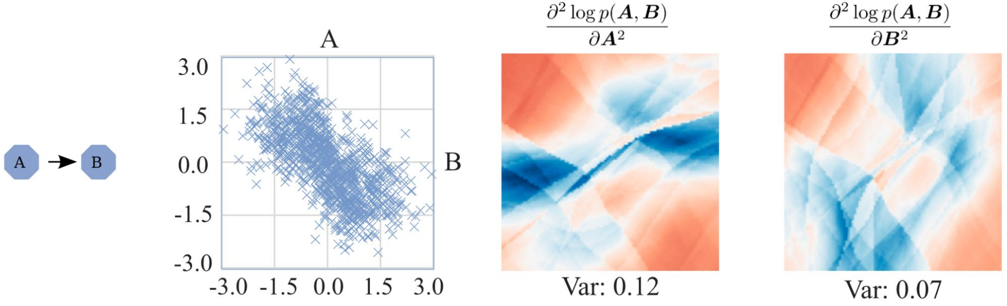
Input: Data matrix $X \in \mathbb{R}^{n\times d}$, regularisation parameter $\eta > 0$.
$s \leftarrow \text{median}(\{\|x_i - x_j\|_2 : i, j = 1, \ldots, n, x_k = X[k, :]\})$.
Compute $\hat{\mathbf{J}}^{\text{Stein}}$ using RBF kernel κ_s, regularisation parameter η and data matrix X based on (12).

Additive noise models conti…

- Implicit asymmetry:
  - If X = f(Y) +eps1, why can't Y be modelled as: Y = g(X) +eps2







Distribution with ANM assumption (1)

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

$$\log p(x) = \sum_{i=1}^{d} \log p(x_i|\text{pa}_i(x)) \quad = \quad \sum_{i=1}^{d} \log p^{\epsilon}\left(\mathrm{x}_i - f_i\right) \qquad \triangleright \text{ Using } \epsilon_i = \mathrm{x}_i - f_i$$

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

score function $s(\mathbf{x}) \equiv \nabla \log p(\mathbf{x})$ reads

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

Properties of leaf node

Lemma 1. *Let p be the probability density function of a random variable X defined via a non-linear additive Gaussian noise model (1), and let $s(x) = \nabla \log p(x)$ be the associated score function. Then, $\forall j \in \{1, \dots, d\}$, we have:*

(i) j is a leaf $\Leftrightarrow \forall x, \frac{\partial s_j(x)}{\partial x_j} = c$, with $c \in \mathbb{R}$ independent of x, i.e., $Var_X \left[\frac{\partial s_j(X)}{\partial x_j} \right] = 0$.

(ii) If j is a leaf, i is a parent of $j \Leftrightarrow s_j(x)$ depends on x_i, i.e., $Var_X \left[\frac{\partial s_j(X)}{\partial x_i} \right] \neq 0$.

- For leaf nodes, gradient of score function wrt itself is constant
- i is an ancestor of j, if var is not zero (may not be parent)

If i is not a parent of j, then $\frac{\partial s_j}{\partial x_i} \equiv 0$, and hence we have $\mathrm{Var}_X \left[\frac{\partial s_j(x)}{\partial x_i} \right] = 0$. On the other hand, if i is a parent of j, then we have $\frac{\partial s_j}{\partial x_i}(x) = \frac{1}{\sigma_j^2} \frac{\partial f_j}{\partial x_i}(\mathrm{pa}_j(x))$. Moreover, since f_j cannot be linear in x_i, $\frac{\partial f_j}{\partial x_i}(\mathrm{pa}_j(x))$ cannot be a constant, and hence $\mathrm{Var}_X \left[\frac{\partial s_j(X)}{\partial x_i} \right] \neq 0$.

Non Gaussian Extension

Lemma 2. *Suppose that the random variable X is generated from (1) where the noise variables ϵ_i are i.i.d. with smooth probability distribution function p^{ϵ}. Then, the score function of X can be written as follows:*

$$s_j(\boldsymbol{x}) = \frac{d \log p^{\epsilon}}{dx}(x_j - f_j(pa_j(\boldsymbol{x}))) - \sum_{i \in children(j)} \frac{\partial f_i}{\partial x_j}(pa_i(\boldsymbol{x})) \frac{d \log p^{\epsilon}}{dx}(x_i - f_i(pa_i(\boldsymbol{x}))). \quad (13)$$

- All the properties of leaf node holds, iff log-distribution is at-max twice differentiable

Algorithm

Algorithm 1 SCORE-matching causal order search

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 $s_{nodes} = \nabla \log p_{nodes}$ (for example using Algorithm 1).
  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
end for
Get the final DAG by pruning the full DAG associated with the topological order π.

Experiments

Metrics:

- **SHD**. Structural Hamming distance between the output and the true causal graph, which counts the number of missing, falsely detected, or reversed edges.
- **SID**. Structural Intervention Distance is based on a graphical criterion only and quantifies the closeness between two DAGs in terms of their corresponding causal inference statements.
- **Order Divergence** measures how well the topological order is estimated. For an ordering π, and a target adjacency matrix A.

Results

Table 6: Synthetic experiment for $d = 50$ with Laplace noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	11.0 ± 4.5	71.8 ± 50.2	4.0 ± 2.5	$\mathbf{128.1 \pm 7.9}$	1384 ± 131	19.8 ± 3.5
CAM	$\mathbf{10.1 \pm 3.4}$	$\mathbf{66.1 \pm 47.9}$	–	134.6 ± 7.2	$\mathbf{1361 \pm 136}$	–
GraN-DAG	21.9 ± 3.9	165.7 ± 46.2	–	138.3 ± 8.8	1603 ± 166	–
VarSort	–	–	8.1 ± 4.2	–	–	47.3 ± 8.7

Table 3: Synthetic experiment for $d = 50$ with Gaussian noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	10.4 ± 3.9	$\mathbf{50.9 \pm 32.9}$	3.9 ± 2.4	$\mathbf{131.5 \pm 7.5}$	$\mathbf{1262 \pm 110}$	16.3 ± 6.1
CAM	$\mathbf{8.3 \pm 2.9}$	53.7 ± 31.9	–	140.8 ± 5.5	1337 ± 94	–
GraN-DAG	20.2 ± 6.1	135.3 ± 45.9	–	140.8 ± 9.5	1432 ± 110	–
VarSort	–	–	8.8 ± 3.0	–	–	43.3 ± 9.7