Causal Discovery for Earth System Sciences

Section 4 – Learning the DAG

Emiliano Díaz Salas Porras

Image and Signal Processing @ Universitat de València

Learning the DAG from Data

Goal: Given observational (or interventional) data, learn the structure of the underlying causal DAG.

Why is this hard?

- The space of DAGs is super-exponential in the number of nodes
- Many DAGs encode the same conditional independencies (Markov Equivalence)
- Need to balance faithfulness, causal sufficiency, and statistical power

Two Main Classes of Methods:

1. **Constraint-based methods** (e.g. PC, FCI): Use conditional independence tests to infer the graph.
2. **Score-based methods** (e.g. GES, NOTEARS): Search over DAGs using a score function (likelihood + complexity penalty).

Other categories:

- Functional causal model–based methods (e.g. LiNGAM, ANM)
- Time series–specific extensions (e.g. PCMCI, tsFCI)

Score-Based Structure Learning: GES

GES = Greedy Equivalence Search

- A two-phase greedy algorithm:
    1. **Forward phase:** Start with an empty graph, greedily add edges to improve the score
    2. **Backward phase:** Remove edges to simplify model without hurting the score too much
- Operates over **equivalence classes** of DAGs (CPDAGs)

Score Function:

$$\text{Score}(\mathcal{G}, D) = \log P(D \mid \mathcal{G}) - \lambda \cdot f(\#\text{edges})$$

Common choices: BIC, AIC, or marginal likelihoods with priors

Assumptions:

- i.i.d. data
- Faithfulness
- Sufficient sample size

python notebook GES

Summary: Score-Based Learning with GES

GES (Greedy Equivalence Search):

- Operates over **CPDAGs** (equivalence classes of DAGs)
- Two phases: **forward** (edge addition), then **backward** (edge deletion)
- Evaluates **directed edge moves** based on score improvement (e.g., BIC)

Key Properties:

- Consistent under faithfulness, correct data model and BIC
- Exploits decomposability of score (local updates)
- Greedy but efficient; avoids full DAG enumeration

When to use GES:

- When a reliable score model is available (e.g. Gaussian, discrete)
- For datasets with a moderate number of variables (10–100)
- Less sensitive to independence test errors than constraint-based methods

Causal Discovery for Time Series Data

Challenges unique to time series:

- Variables evolve over time — temporal index is crucial
- Autocorrelation within variables → violates i.i.d. assumptions
- Cross-lagged dependencies: X_t may cause Y_{t+1}
- Feedback loops, seasonality, and stationarity issues

Goal:

- Recover the time-lagged causal structure
- Understand temporal order and delayed effects

Strategies:

- Use **rolled-out graphs** (lags as explicit nodes, only contemporaneous loops possible! more identifiability)
- Separate **instantaneous** and **lagged** effects
- Use independence tests that account for autocorrelation

The MCI Test: Controlling False Positives in Time Series

Problem: Autocorrelation causes spurious associations

- In time series, many variables are correlated with their own past (e.g., $X_t \sim X_{t-1}$)
- This can leak into apparent dependencies:

$$X_{t-1} \to Y_t \quad \text{may look significant just because } Y_t \sim Y_{t-1}$$

- Standard conditional independence tests may wrongly find such links unless autocorrelation is blocked

MCI = Momentary Conditional Independence test

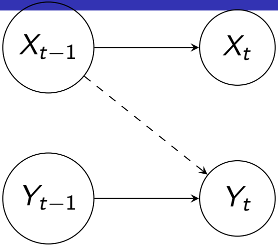
- Tests: Is $X_{t-\tau}$ conditionally independent of Y_t given:

$$\mathrm{Pa}(Y_t) \setminus \{X_{t-\tau}\} \cup \mathrm{Pa}(X_{t-\tau})$$

Why it works:

- Filters edges that only appear significant due to time dependence
- Proven to control false positive rate under autocorrelated noise

DAG Example: Autocorrelation Induces False Positives



Explanation:

- Both X and Y are autocorrelated
- There is no real causal link from $X_{t-1} \to Y_t$
- But marginally, X_{t-1} and Y_t may appear dependent due to:

$$X_{t-1} \to X_t \qquad Y_{t-1} \to Y_t$$

- Standard CI tests may incorrectly detect $X_{t-1} \to Y_t$

MCI Test Fix:

- Conditions on:

$$\mathrm{Pa}(Y_t) = \{Y_{t-1}\}, \quad \mathrm{Pa}(X_{t-1}) = \{X_{t-2}\}$$

Why MCI Conditions on Parents of Both Variables

Consider the FCM:

$$X_t = f_X(X_{t-1}, e_t^X)$$
$$Y_t = f_Y(Y_{t-1}, e_t^Y)$$

No true causal link: $X_{t-1} \not\to Y_t$

But suppose:

- $e_t^X \not\perp\!\!\!\perp e_t^Y$ (correlated noise)
- $X_t \sim X_{t-1}$ and $Y_t \sim Y_{t-1}$ (autocorrelation)

Then: $X_{t-1} \sim Y_t$ — even though there's no causation

Spurious path:

$$X_{t-1} \to X_t \sim Y_t \leftarrow Y_{t-1}$$

MCI test blocks this:

- Conditions on:

$$\mathrm{Pa}(Y_t) \setminus \{X_{t-\tau}\} \cup \mathrm{Pa}(X_{t-\tau})$$

PCMCI: Causal Discovery in High-Dimensional Time Series

PCMCI = PC algorithm + M(omentary) CI test

Main idea:

- First use a PC-style selection phase to find candidate parents
- Then apply a second filter (MCI test) to prune false positives

Step-by-step:

1. Choose a **maximum time lag** $\tau_{\max}$
2. For each variable X_t, construct a set of candidate parents $\mathrm{Pa}(X_t)$ from past values (PC-1 phase)
3. Apply the **MCI test** for each edge $Y_{t-\tau} \to X_t$ by conditioning on remaining parents

Advantages:

- Controls false positives under autocorrelation
- Scalable to high-dimensional time series
- Separates lagged and instantaneous effects

What if We Don't Observe All Variables?

Causal Sufficiency: Assumes all common causes (confounders) are observed

But in practice:

- Earth system datasets may miss important environmental or anthropogenic factors
- Hidden variables can induce spurious associations
- Standard PC or GES may infer incorrect edges or orientations

Enter: **FCI** and time-series variants (tsFCI, LPCMCI)

Goal:

- Learn causal structure while allowing for unobserved variables
- Return a Partial Ancestral Graph (PAG) — a generalization of CPDAG

How FCI Works: CD with Latent Confounders

FCI = Fast Causal Inference

Main Ideas:

1. **Skeleton discovery:** Like PC — use CI tests to remove edges
2. **Sepset recording:** Keep track of conditioning sets for each removed edge
3. **Orientation rules:**
   - Infer v-structures
   - Apply more general orientation rules to form a PAG (Partial Ancestral Graph)
4. **Edge marks encode uncertainty:**

$$A \circ\!\!\rightarrow B, \quad A \leftrightarrow B, \quad A \circ\!\!-\!\!\circ B$$

Output: A **PAG** — encodes the equivalence class of graphs under hidden confounding

Why use FCI?

- Robust to latent variables and selection bias
- More conservative but safer graph

How to Read a PAG (Partial Ancestral Graph)

PAG = output of FCI (or tsFCI, LPCMCI)

Why not just arrows?

- With hidden variables, we can't always determine causal direction
- Need to represent ambiguity about causality and confounding

PAG edge marks:

- **A** $\rightarrow$ **B** : A is a (possibly indirect) cause of B
- **A** $\leftrightarrow$ **B** : A and B share an unobserved common cause
- **A** $\circ\!\!\rightarrow$ **B** : A may or may not be a cause of B
- **A** $\circ\!\!-\!\!\circ$ **B** : Ambiguous direction and possible confounding

Examples:

$$A \rightarrow B \quad \text{(confident: A is ancestral to B)}$$

$$A \leftrightarrow B \quad \text{(confounded: unobserved common cause)}$$

$$A \circ\!\!\rightarrow B \quad \text{(possibly A causes B, but unsure)}$$

$$A \circ\!\!-\!\!\circ B \quad \text{(no definite claim about direction or confounding)}$$

Key Idea:

- Unroll the time series into a **lagged graph**:

$$\text{Nodes: } X_t, X_{t-1}, X_{t-2}, \dots$$

- Apply standard FCI over this expanded graph

Adaptations:

- Use CI tests adapted for autocorrelated data (e.g., ParCorr, CMI)
- Edge orientation rules follow standard FCI logic
- Output: **Time-lagged PAG** — includes $\rightarrow$, $\leftrightarrow$, $\circ\!\!\rightarrow$, etc.

Pros:

- Sound under hidden confounders
- Explicit representation of lags and confounding

Cons:

- Graph size scales with $N \times \tau_{\text{max}}$
- Slower and harder to interpret in high dimensions

LPCMCI: Scalable Discovery with Confounding Awareness

How it works:

1. **Lag selection (like PCMCI):** Candidate parents from past lags
2. **MCI Test:** Filters out false positives using autocorrelation-aware conditioning
3. **Orientation phase:** Applies FCI-style logic to construct a **summary PAG**

Output:

- A PAG over **original variables**, with lags attached to edges
- Conservatively oriented edges to reflect uncertainty and possible confounding

Why it's useful:

- More scalable than tsFCI
- Handles hidden variables and autocorrelation
- Useful for exploratory Earth system science

Algorithmic Independence: Identifiablity within MECs

Key Principle:

- The **cause** and the **mechanism** that maps cause to effect should be algorithmically independent

Implication:

- If $Y = f(X) + N$, then P_X and f are independent
- But if you model X as a function of Y, you may get dependencies between P_Y and the inverse function

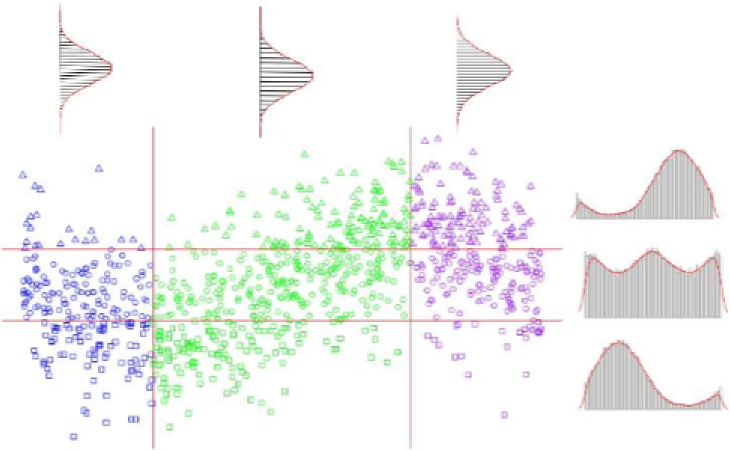
What this gives us:

- An **asymmetry** between $X \to Y$ and $Y \to X$
- No conditional independencies needed!

Used in:

- Additive Noise Models (ANM)
- Information-Geometric Causal Inference (IGCI)
- RESIT, HSIC-based methods

Identifiability beyond Markov equivalence class



- modularity/autonomy assumption
- $p(x)$ algorithmically independent from $p(y|x)$)
- ie no info about $p(y|x)$ in $p(x)$

Granger Causality: Causality as Predictive Improvement

Key Idea: Variable X **Granger-causes** Y if past values of X help predict Y, over and above past values of Y alone.

$$X \text{ Granger-causes } Y \iff E(Y_t \mid Y_{t-1}^{(p)}, X_{t-1}^{(p)}) \neq E(Y_t \mid Y_{t-1}^{(p)})$$

Assumptions:

- Only first order dependence
- Don't care about the lag of relationships
- No instantaneous causality (no $X_t \to Y_t$)
- Causal sufficiency (no hidden confounders)

Strengths:

- Simple, fast, widely available
- Works well when assumptions are met

Limitations:

- Sensitive to sampling rate and lag order
- Cannot handle instantaneous effects or hidden variables

Invariant Causal Prediction (ICP)

Goal: Find the set of causal parents of a target variable that are invariant across multiple environments (or interventions)

Core Assumption:

- The **conditional distribution of the target given its true causes remains invariant** under different environments

$$P(Y \mid \mathrm{Pa}(Y)) \text{ is the same across environments}$$

Setup:

- Multiple datasets or environments $e \in \mathcal{E}$
- Assume that interventions may affect covariates, but not directly the target
- Search for subset $S \subseteq$ Predictors such that:

$$P(Y^e \mid S^e) \text{ is invariant across all } e \in \mathcal{E}$$

What ICP gives you:

- A set of variables that are **causal parents** of Y
- No need to know what was intervened on!

Example of Invariant Causal Prediction

Scenario: You're studying what causes vegetation growth Y in different regions or seasons (environments e).

Available variables:

- X_1: Soil moisture
- X_2: Solar radiation
- X_3: Precipitation
- X_4: Human intervention

Observation:

- Across different environments (e.g., wet vs dry seasons):

$$P(Y \mid X_1, X_2) \text{ remains stable}$$

- But:

$$P(Y \mid X_3) \text{ or } P(Y \mid X_4) \text{ varies across environments}$$

Example of Invariant Causal Prediction

Conclusion from ICP:

- X_1 and X_2 are likely **direct causes** of Y
- X_3 and X_4 are likely **associated but non-causal** (affected by the environment)

ICP's strength: Causal inference without knowing what was intervened on — just by looking for stability!

Learning DAGs with Neural Networks: NOTEARS

Challenge: Learning a DAG is a **combinatorial problem**

Key Insight (Zheng et al., 2018): Relax the DAG constraint into a continuous space using a differentiable constraint:

$$h(A) = \mathrm{tr}(e^{A \circ A}) - d = 0$$

where:

- A is the weighted adjacency matrix
- $h(A) = 0$ iff A encodes a DAG

Objective:

$$\min_{A} \ \mathcal{L}(\hat{X}, X) + \lambda \cdot h(A)$$

Train a linear (or nonlinear) model and penalize deviations from acyclicity.

Learning DAGs with Neural Networks: NOTEARS

Advantages:

- Allows use of gradient descent and auto-diff
- Scales to larger graphs
- Easy to integrate with deep learning pipelines

Limitations:

- Requires continuous data
- Still assumes causal sufficiency and faithfulness
- sparsity is hard to obtain with optimizers not designed for looking in restriction boundaries.

Limitations of Continuous DAG Learning Approaches

1. **DAG Constraint Is Soft in Many Models**
   - NOTEARS enforces $h(A) = 0$ as a hard constraint
   - But most neural models (e.g., GraN-DAG, DAG-GNN) only penalize:

$$\min_{A} \mathcal{L} + \lambda_1 \|A\|_1 + \lambda_2 \cdot h(A)$$

   - In practice, $h(A) \approx \epsilon$, not exactly zero
2. **Sparsity Is Not Guaranteed**
   - Graphs may be dense unless explicitly regularized
   - Sparsity requires:
     - ℓ_1-penalty on adjacency matrix
     - Special optimizers (e.g., proximal methods, interior-point)
3. **Requires Thresholding + Postprocessing**
   - Final A must be thresholded to obtain discrete structure
   - Must check for cycles after thresholding!

Pathwise Multiplication: Extracting Adjacency from Deep Networks

Problem: In multi-layer neural nets, the causal structure is **implicit**. You can't read adjacency A_{ij} directly from input weights.

Solution: Pathwise Multiplication

- Estimate the influence of $X_i \to X_j$ by multiplying weights along all paths
- Each path: sequence of weights from input X_i to output node of X_j's network
- Total influence:

$$A_{ij} = \sum_{\text{paths } X_i \to X_j} \prod_{\text{edges in path}} |w|$$

Pathwise Multiplication: Extracting Adjacency from Deep Networks

Example (3-layer net):

$$X_i \to h_1 \to h_2 \to X_j \Rightarrow \text{influence} = |w^{(1)}| \cdot |w^{(2)}| \cdot |w^{(3)}|$$

Why it's useful:

- Allows extraction of soft adjacency from black-box neural nets
- No need to constrain the network structure directly

Limitations:

- May overestimate inactive paths (nonlinearities, ReLU)
- Doesn't guarantee acyclicity or causality
- Still requires thresholding or pruning

Pathwise Influence via Weight and Mask Multiplication

Goal: Estimate how much input X_i influences output X_j $\rightarrow$ without tracing every path explicitly

Assume:

- Fully connected feedforward network with layers $W^{(1)}, \ldots, W^{(L)}$
- ReLU activations (or similar)
- Optional: input gates $G \in \mathbb{R}^{d \times d}$

Step-by-step:

1. Perform a forward pass to identify **active neurons**
2. For each layer l, create a diagonal mask matrix:

$$D_{ii}^{(l)} = \begin{cases} 1 & \text{if neuron } i \text{ is active} \\ 0 & \text{otherwise} \end{cases}$$

3. Total influence approximation:

$$A = W^{(L)} D^{(L-1)} W^{(L-1)} \cdots D^{(1)} W^{(1)} G$$

Pathwise Influence via Weight and Mask Multiplication

What this gives you:

- A_{ij}: Approximate influence of $X_i \to X_j$
- Can be thresholded to extract adjacency structure

Advantages:

- No need to enumerate paths
- Fully differentiable, GPU-friendly
- Captures both structure and nonlinear activations

Causal Representation Learning (CRL)

In many real-world systems, the true causal variables are **not directly observed**

Examples:

- Satellite pixels $\neq$ true physical processes
- fMRI voxels $\neq$ cognitive mechanisms
- Raw sensors $\neq$ high-level climate drivers

Challenge:

- Observed variables may be:
  - Entangled mixtures of latent causes
  - High-dimensional with noisy redundancies
  - Measured at the wrong scale
- Standard DAG methods break down under these conditions

Goal of CRL:

- Learn a representation $Z = \phi(X)$ that:
  - Recovers a set of latent variables with causal meaning (sparse DAG between them)
  - Makes causal discovery or prediction more reliable

Why Is Causal Representation Learning Hard?

Key problem:

- There are infinitely many ways to represent high-dimensional data X
- Most latent representations will not correspond to causal variables

Main difficulties:

- **Identifiability:** Without extra assumptions, it's impossible to tell which latent structure is causal
- **Entanglement:** Observed variables may be linear or nonlinear mixtures of underlying causes
- **Scale mismatch:** Mechanisms act at spatial/temporal scales different from measurement
- **Noisy observations:** Noise and redundancies distort the true graph structure

Why Is Causal Representation Learning Hard?

What can help?

- **Interventions or environments:** Causal mechanisms remain invariant across changes
- **Temporal structure:** Time helps disentangle dependencies and directionality
- **Inductive biases:** Assume sparsity, modularity, or disentanglement in representation
- **Learning constraints:** Use priors or structural assumptions to guide neural encoders

CRL is an open research frontier — but crucial for causal discovery in complex systems!