Causal Inference and Machine Learning
Session 8: Causal discovery from observational and experimental data using GIES

Alessandro Mascaro

February 25th, 2025

Barcelona School of Economics,
Master's degree in Data Science Methodology

Greedy Equivalence Search from observational data

Causal discovery in one slide

- Suppose we observe a (n, p) data matrix of observations $\boldsymbol{X}$, containing i.i.d. samples of a random vector $X := (X_1, \ldots, X_p)$;
- If X is generated from a Markovian Structural Causal Model with causal structure represented by $\mathcal{D}$, then a set of DAG-specific conditional independencies $\mathcal{I}(\mathcal{D})$ holds in the distribution of X, so that its pdf factorises as

$$p(x_1, \ldots, x_p) = \prod_{j=1}^{p} p(x_j \mid x_{\mathrm{pa}_{\mathcal{D}}(j)});$$

- If **faithfulness** is assumed, then one can try to learn the DAG from data by "testing" for these conditional independencies;
- Two main categories of causal discovery methods: constraint-based (PC algorithm) and score-based (Hill climbing);

Score-based methods: Introduction

- Score-based methods assign a score to each DAG and then select the DAG that maximises that score;
- Assuming a parametric distribution for X with parameter $\Theta_{\mathcal{D}}$, a popular choice for the score is the BIC:

$$\text{BIC}(\mathcal{D}, \boldsymbol{X}) = \log p(\boldsymbol{X}; \hat{\Theta}_{\mathcal{D}}) - \frac{\dim(\Theta_{\mathcal{D}})}{2} \log(n),$$

where

- $\hat{\Theta}_{\mathcal{D}}$ is the maximum likelihood estimate of $\Theta_{\mathcal{D}}$;
- $\log p(\boldsymbol{X}; \hat{\Theta}_{\mathcal{D}})$ is the likelihood of the data $\boldsymbol{X}$;
- $\dim(\Theta_{\mathcal{D}})$ is the number of free parameters in the distribution induced by the DAG $\mathcal{D}$;

Score-based methods: BIC

- Different scores may lead to different results!
- However, the BIC enjoys many useful properties:
  - It is **decomposable**: it can be written as

$$\text{BIC}(\mathcal{D}, \boldsymbol{X}) = \sum_{j=1}^{p} \log p(\boldsymbol{X}_{\cdot j} \mid \boldsymbol{X}_{\cdot \text{pa}_{\mathcal{D}}(j)}; \hat{\Theta}_{\mathcal{D}}^{(j)}) - \frac{d_j}{2} \log(n)$$

    where $\Theta_{\mathcal{D}}^{(j)}$ are the parameters associated with the j-th conditional density of the factorization and $d_j = \dim(\Theta_{\mathcal{D}}^{(j)})$
  - It is **consistent**: as $n \to \infty$, it is highest for the true DAG;
  - It guarantees **score equivalence**: two unidentifiable (Markov equivalent) DAGs $\mathcal{D}_1, \mathcal{D}_2$ are assigned the same score;

Score-based methods: fundamental problem

- We are interested in

$$\hat{\mathcal{D}} = \text{argmax}_{\mathcal{D} \in \mathcal{S}_p} \text{BIC}(\mathcal{D}, \boldsymbol{X});$$

- You can guess what's the problem...

Score-based methods: fundamental problem

- We are interested in

$$\hat{\mathcal{D}} = \text{argmax}_{\mathcal{D} \in \mathcal{S}_p} \text{BIC}(\mathcal{D}, \boldsymbol{X});$$

- You can guess what's the problem...

p	$\lvert\mathcal{S}_p\rvert$ (No. of DAGs)
1	1
3	3
5	543
7	3781503
9	783702329343
11	4175098976430598143
13	521939651343829405020504063
15	143942814104439833494179071983953510 3

- We can't enumerate all the DAGs, we need **greedy algorithms**!

Score-based methods: Hill Climbing

- The easiest example of **greedy algorithm** for causal discovery is the **hill climbing algorithm**;
- It starts from an initial DAG $\mathcal{D}^{(0)}$ (often the empty DAG) and, at each iteration $s > 1$:
  - It constructs the set neigh_s of neighboring DAGs that can be obtained by **adding**, **deleting**, or **reversing** a single edge;
  - For each $\tilde{\mathcal{D}} \in \text{neigh}_s$, computes $\text{score}(\tilde{\mathcal{D}}, \boldsymbol{X})$;
  - Sets $\mathcal{D}^{(s)} = \text{argmax}_{\tilde{\mathcal{D}} \in \text{neigh}_s} \text{score}(\mathcal{D}, \boldsymbol{X})$;

  these steps are repeated until the score can't be improved

Score-based methods: Comments Hill Climbing

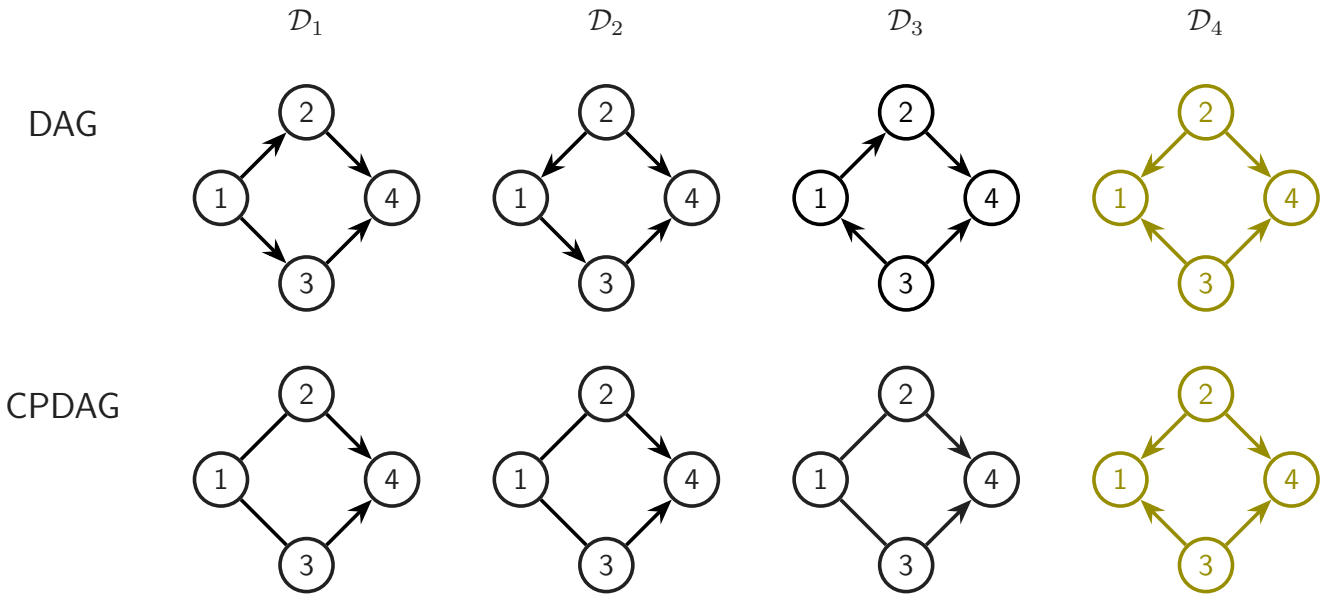
- The combination of local moves considered by the algorithm and decomposable scores leads to significant computational savings;
- However, even if the score is consistent, the algorithm is not: it may remain stuck in local modes, without reaching the local optimum $\hat{\mathcal{D}}$;
- Morover, if the score satisfies score equivalence, there is no unique global optimum, but as many as the DAGs in a Markov Equivalence class;
- **Solution:** Greedy algorithm defined on the space of Markov equivalence classes: **Greedy Equivalence Search** (GES);

Markov Equivalence classes and CPDAGs

- A DAG $\mathcal{D}_1$ implies a set of conditional independencies $\mathcal{I}(\mathcal{D}_1)$ that can be read-off from the graph using d-separation and is reflected in the factorization of the pdf;
- It may happen that, for two DAGs $\mathcal{D}_1$ and $\mathcal{D}_2$, $\mathcal{I}(\mathcal{D}_1) = \mathcal{I}(D_2)$. When this happens, we say that $\mathcal{D}_1$ and $\mathcal{D}_2$ are **Markov Equivalent**;
- Two Markov equivalent DAGs share the same **skeleton** and the same set of **v-structures**: they differ only for the reversal of **some** edges;
- One can use a graph with undirected and directed edges to represent a Markov Equivalence class of DAGs. This is called **Essential graph** or **Completed Partially Directed Acyclic Graph** (CPDAG);

CPDAGs: Example

- For instance, here are four DAGs and their CPDAGs:



- $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ are Markov equivalent and share the same CPDAG!

GES: Introduction

- Greedy Equivalence Search is a greedy causal discovery algorithm that works directly on the space of CPDAGs $\mathcal{G}$;
- As such, it requires (i) a **score** associated with each CPDAG and (ii) a set of **local moves** to explore the space of possible CPDAGs;
- Moreover, GES is composed of two distinct phases.
  - **Forward phase**: starting from a CPDAG $\mathcal{G}^{(0)}$ with no edges, undirected and directed edges are greedily inserted until the score can't be improved. It returns a CPDAG $\mathcal{G}^{(fp)}$;
  - **Backward phase**: starting from $\mathcal{G}^{(fp)}$, edges are greedily removed. It returns the candidate CPDAG $\hat{\mathcal{G}}$;

GES: Score

- Different possibilities for the score. We will focus again on the BIC:

$$\mathrm{BIC}(\mathcal{G}, \boldsymbol{X}) = \log p(\boldsymbol{X}; \hat{\Theta}_{\mathcal{G}}) - \frac{\dim(\Theta_{\mathcal{G}})}{2} \log(n),$$

- For any two DAGs $\mathcal{D}_1$ and $\mathcal{D}_2$ belonging to the same Markov equivalence class represented by $\mathcal{G}$, we have that

$$\mathrm{BIC}(\mathcal{D}_1, \boldsymbol{X}) = \mathrm{BIC}(\mathcal{D}_2, \boldsymbol{X}) = \mathrm{BIC}(\mathcal{G}, \boldsymbol{X}).$$

  Because of score equivalence, the score of the CPDAG is the same of any of the DAGs it represents

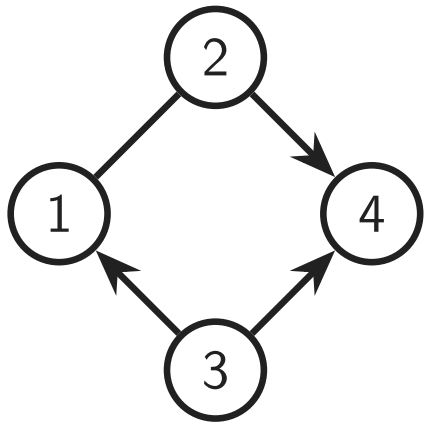
- The BIC inherits all the nice properties it has in the DAG setting (it is *consistent* and *decomposable*)

GES: Local moves

- For two nodes i and j in $\mathcal{G}$, we say that i, j are *adjacent* if they are connected by a directed or undirected edge in $\mathcal{G}$ and that i, j are *neighbors* if they are connected by an undirected edge;
- Chickering (2002) proposed the following two operations, each corresponding to one of the two phases:
  - **Insert**$(i, j, \boldsymbol{T})$: For non-adjacent nodes i and j in $\mathcal{G}$, and for any subset $\boldsymbol{T}$ of the neighbors of j that are not adjacent to i, insert $i \to j$ and, for each $k \in \boldsymbol{T}$, direct $k - j$ as $k \to j$;
  - **Delete**$(i, j, \boldsymbol{H})$: For adjacent nodes i and j in $\mathcal{G}$, and for any subset $\boldsymbol{H}$ of the neighbors of j that are adjacent to i, delete the edge between i and j, and for each $k \in \boldsymbol{H}$, direct the $j - k$ as $j \to k$ and direct any previously undirected edge between i and k as $i \to k$.

GES: Invalid local moves

- The local moves by Chickering can still lead to **invalid** CPDAGs, i.e. graphs with undirected and directed edges that do not represent any Markov equivalence class;
- Consider for instance the following graph:



  It is not a CPDAG, as directing in any way the undirected edge $1 - 2$ leads to DAGs which are in two different Markov Equivalence classes!
- The validity of CPDAG must be manually checked. Chickering (2002) also provides a relatively quick way to do that;

GES: Summary

- In the **forward** (**backward**) phase, GES starts from the empty CPDAG $\mathcal{G}^{(0)}$ (the CPDAG $\mathcal{G}^{(fs)}$) and, at each iteration $s > 1$:
  - It constructs the set neigh_s of neighboring CPDAGs that can be obtained by one **Insert** (**Delete**) operation;
  - For each $\tilde{\mathcal{G}} \in \text{neigh}_s$, computes $\text{score}(\tilde{\mathcal{G}}, \boldsymbol{X})$;
  - Sets $\mathcal{G}^{(s)} = \text{argmax}_{\tilde{\mathcal{G}} \in \text{neigh}_s} \text{score}(\mathcal{G}, \boldsymbol{X})$;

  these steps are repeated until the score can't be improved. Then a CPDAG $\mathcal{G}^{(\text{fs})}$ ($\hat{\mathcal{G}}$) is returned;

GES: Final comments

- If used with a consistent score, **GES is consistent**: It identifies the CPDAG corresponding to the true DAG as $n \to \infty$;
- Working directly on the space of Markov equivalence classes, it also provides better results in finite samples;
- However, the local moves are more difficult to define and their validity more difficult to check than for Hill-Climbing. It may be more computational demanding;
- It is still much more efficient than constraint-based methods, and recent developments are making it even faster using heuristics (Ramsey et al., 2017)

GES: Software

- An implementation of GES is available via the constructor `new()` and the function `ges()` of the `R` package `pcalg`
- Inputs of `new()` are
  - `Score` : which score must be used;
  - `data`: the data matrix $\boldsymbol{X}$
- Input of `ges()` is only the score produced by `new()`
- Let's see how it works in `R`!

Beyond Markov equivalence: Making use of experimental data

Experimental data: definition

- When only **observational data** are available, even assuming that X is generated by a Markovian SCM only leads us as far as the Markov equivalence class of the true causal structure;
- We define **experimental data** as any data observed after an external intervention to the system. In particular, we consider only **hard interventions**, fixing the value of a **target** variable to a constant;
- Intuitively, using experimental data should help up uncovering more of the causal structure, but how exactly?

Experimental data: how

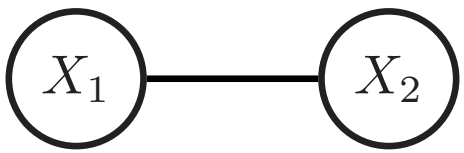
- Key assumptions of SCMs: each mechanism is stable and autonomous

$\Longrightarrow$ an external intervention *only* affects the mechanisms of the target variables, destroying the dependence with their parents (i.e. its causes) in the DAG!

$\Longrightarrow$ If we are undecided on the direction of an edge $i - j$, we can intervene on one of the two nodes and see whether the dependence represented by $i - j$ is destroyed.
If **yes**, the intervened node is the effect.
If **no**, the intervened node is the cause!

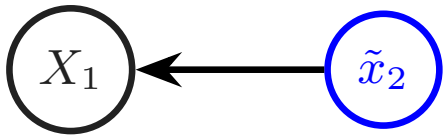
Using experimental data: example - 1

- Let's consider a very simple case in which we have a $(n_1, 2)$ data matrix $\boldsymbol{X}^{(1)}$ with i.i.d. measurements of (X_1, X_2);
- Suppose there are no condtional independencies in the data, so that any causal discovery algorithm returns the following CPDAG:



Using experimental data: example - 2

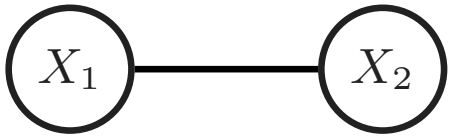
- Suppose now we have access to measurements of $(X_1, X_2) \,|\, \text{do}(X_2)$, stored in the $(n_2, 2)$ matrix $\boldsymbol{X}^{(2)}$.
- The intervention on X_2 only modifies its mechanism, meaning that
  - if $X_2 \to X_1$ the post-intervention DAG will be



  - If $X_1 \to X_2$, the post-intervention DAG will be

Using experimental data: example - 3

- Suppose we now use a causal discovery algorithm on $\boldsymbol{X}^{(2)}$;
  - If its output is



    then $X_2 \to X_1$ is the true causal structure!
  - Otherwise, if its output is

    then $X_1 \to X_2$ is the true causal structure!
- By using experimental data, we are going beyond Markov equivalence!

Using experimental data: comments

- As the **invariances**/**differences** between observational and experimental data are DAG-specific, we can use them to learn the causal structure!
- But **how far** can these invariances bring us?

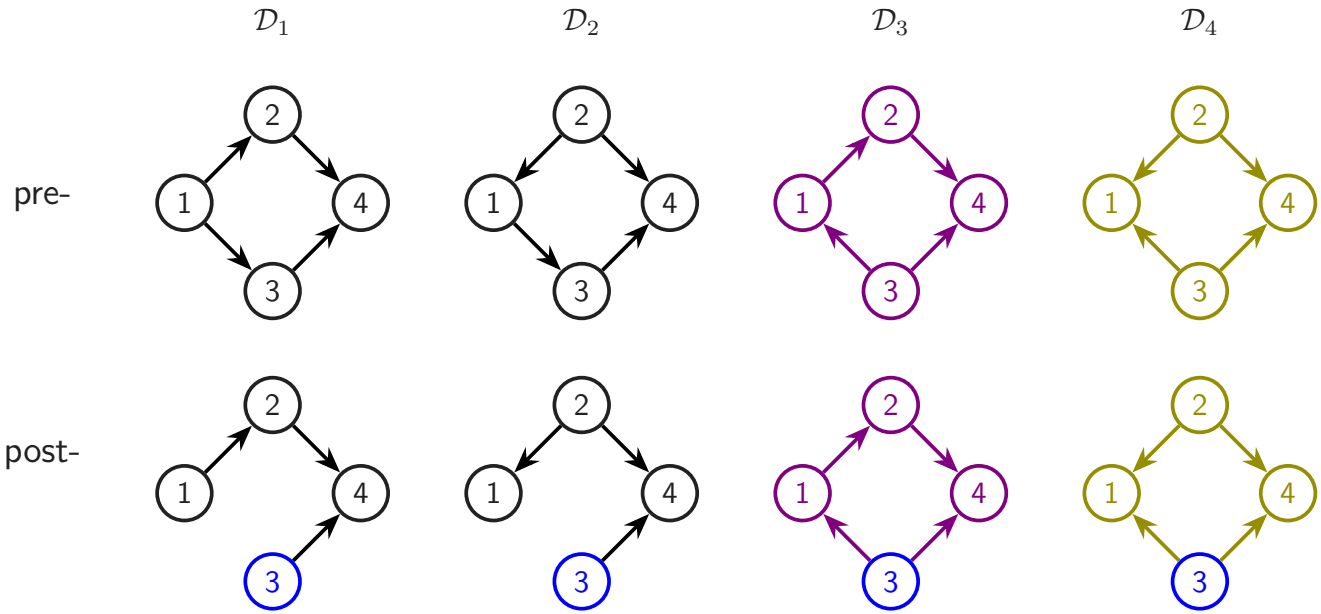
Using experimental data: setting

- **Setting:** we observe a set of K data matrices $\{\boldsymbol{X}^{(k)}\}_{k=1}^{K}$, where
  - $\boldsymbol{X}^{(1)}$ contains i.i.d. samples from the observational distribution of X;
  - $\boldsymbol{X}^{(2)}, \dots, \boldsymbol{X}^{(K)}$ contain samples from the post-intervention distributions of X given a hard intervention on the target nodes $\boldsymbol{T}^{(k)}$.

  We also denote with $\mathcal{T} = \{\boldsymbol{T}^{(k)}\}_{k=1}^{K}$ the multi-set of intervention targets and $\{p_k(x)\}_{k=1}^{K}$ the corresponding post-intervention distributions;
- **Goal:** Design a causal discovery algorithm that takes as input $\{\boldsymbol{X}^{(k)}\}_{k=1}^{K}$ and $\mathcal{T} = \{\boldsymbol{T}^{(k)}\}_{k=1}^{K}$ and returns an estimate of the causal structure $\mathcal{D}$
- **Preliminary question**: Can we actually identify the DAG?

I-Markov equivalence

- Consider the following four DAGs and the corresponding post-intervention DAGs produced by an intervention on X_3:



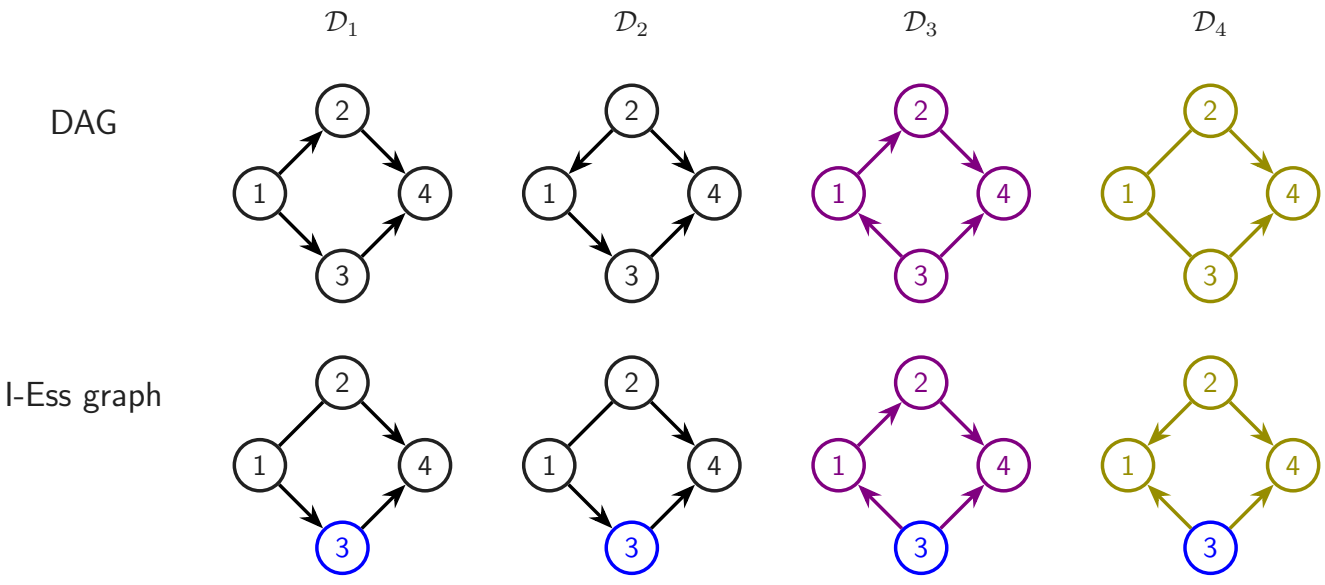
- $\mathcal{D}_1, \mathcal{D}_2$ imply the same set of conditional independencies and invariances: they're **I-Markov equivalent** given an intervention on X_3;

I-Markov equivalence and I-Essential graphs

- I-Markov equivalence classes are the identifiability limit when also experimental data are available;
- Hauser & Bühlmann (2012) provide two **graphical characterizations** of I-Markov equivalence:
  - All I-Markov equivalent DAGs have the same *skeleton* and the same set of *v-structures* in all pre- and post-intervention DAGs;
  - Two DAGs are I-Markov equivalent if they have the same **I-Essential graph**;
- As for the observational case, the **I-Essential Graph** is a graph with undirected edges in correspondence of the edges that can be directed in both ways without going out of the I-Markov equivalence class;

I-Essential graph: Example

- Let's consider the same four DAGs as before, with $\mathcal{T} = \{\emptyset, \{3\}\}$:



GIES:

- We want to design algorithms that use **both** the *conditional independencies* within each dataset and the *invariances* across datasets to learn the causal structure
- In what follows, we will consider an extension of GES called Greedy Interventional Equivalence Search (Hauser & Buhlmann, 2012);

GIES: Introduction

- Greedy Interventional Equivalence Search is a **greedy**, **score-based** causal discovery algorithm that works directly on the space of I-Essential graphs $\mathcal{E}$;
- As usual, it requires (i) a **score** associated with each I-Essential graph and (ii) a set of **local moves** to explore the space of possible I-Essential graphs;
- GIES is composed of three distinct phases.
  - **Forward phase**: starting from a CPDAG $\mathcal{E}^{(0)}$ with no edges, directed and undirected edges are greedily inserted until the score can't be improved. It returns an I-Essential graph $\mathcal{E}^{(\text{fp})}$;
  - **Backward phase**: starting from $\mathcal{E}^{(\text{fp})}$, edges are greedily removed. It returns an I-Essential graph $\mathcal{E}^{(\text{bp})}$;
  - **Turning phase**: starting from $\mathcal{E}^{(\text{bp})}$, edges are greedily turned. It returns the candidate I-Essential graph $\hat{\mathcal{E}}$

GIES: Score

- As for any score-based method, there are different possibilities for the score. Usually, consistent and score-equivalent ones are preferred
- We will use again the BIC that, in this setting becomes:

$$\text{BIC}(\mathcal{E}, \{\boldsymbol{X}\}_{k=1}^{K}) = \sum_{k=1}^{K} \log p(\boldsymbol{X}^{(k)}; \hat{\Theta}_{\mathcal{E}}) - \frac{\dim(\Theta_{\mathcal{E}})}{2} \log(n),$$

- BIC is still decomposable, as it can be written as

$$\text{BIC}(\mathcal{E}, \{\boldsymbol{X}\}_{k=1}^{K}) = \sum_{j=1}^{p} \log p(\boldsymbol{X}_{.j}^{A(j)} \,|\, p(\boldsymbol{X}_{.\text{pa}_{\tilde{\mathcal{D}}}(j)}^{A(j)}; \hat{\Theta}_{\mathcal{E}}^{(j)}) - \frac{\dim(\Theta_{\mathcal{E}}^{(j)})}{2} \log(n)$$

  where $A(j) := \{k \in [K] : j \notin \boldsymbol{T}^{(k)}\}$ and $\tilde{\mathcal{D}}$ a DAG represented by $\mathcal{E}$
- The BIC is **consistent** in the Gaussian setting (Hauser & Buhlmann, 2015);

GIES: Local moves

- We will not go into the details of the local moves used by GIES in each phase, as they require some preliminary work to be understood and can be quite complicated;

Input : $G = ([p], E)$: $\mathcal{I}$-essential graph; $(\mathcal{T}, \mathbf{X})$: interventional data for $\mathcal{I}$
Output: $G' \in \mathcal{E}^+_{\mathcal{I}}(G)$, or G
$\Delta S_{\max} \leftarrow 0$;
foreach $v \in [p]$ **do**
  foreach $u \in [p] \setminus \mathrm{ad}_G(v)$ **do**
    $N \leftarrow \mathrm{ne}_G(v) \cap \mathrm{ad}_G(u)$;
    foreach *clique* $C \subset \mathrm{ne}_G(v)$ *with* $N \subset C$ **do** `// Proposition 25(i) and (ii)`
      if $\nexists$ *path from* v *to* u *in* $G[[p] \setminus C]$ **then** `// Proposition 25(iii)`
        $\Delta S \leftarrow s(v, \mathrm{pa}_G(v) \cup C \cup \{u\}; \mathcal{T}, \mathbf{X}) - s(v, \mathrm{pa}_G(v) \cup C; \mathcal{T}, \mathbf{X})$;
        if $\Delta S > \Delta S_{\max}$ **then**
          $\Delta S_{\max} \leftarrow \Delta S$;
          $(u_{\max}, v_{\max}, C_{\max}) \leftarrow (u, v, C)$;
if $\Delta S_{\max} > 0$ **then**
  $\sigma \leftarrow \textsc{LexBFS}((C_{\max}, v_{\max}, \ldots), E[T_G(v_{\max})])$;
  Orient edges of $G[T_G(v_{\max})]$ according to σ;
  Insert edge $(u_{\max}, v_{\max})$ into G;
  return $\textsc{ReplaceUnprotected}(\mathcal{I}, G)$; `// See Algorithm 1`
else return G;

Algorithm 3: $\textsc{ForwardStep}(G; \mathcal{T}, \mathbf{X})$. One step of the forward phase of GIES.

GIES: Final comments

- GIES uses both observational and experimental data to learn the causal structure generating the data;
- Its output is an I-Essential graph representing an I-Markov equivalence class of indistinguishable DAGs;
- I-Markov equivalence classes contain strictly fewer DAGs than Markov equivalence classes $\implies$ we are learning more of the causal structure!
- **However**, unline GES, **GIES** is **not consistent**, even when a consistent score such as the BIC is used;
- We will see in the following lectures other methods that overcome this problem;

GIES: Software

- An implementation of GIES is available via the constructor `new()` and the function `gies()` of the `R` package `pcalg`
- Inputs of `new()` are:
  - `Score` : which score must be used;
  - `data` : data matrix $\boldsymbol{X}$ obtained by stacking $\{\boldsymbol{X}^{(k)}\}_{k=1}^{K}$ vertically;
  - `targets` : list of targets $\mathcal{T}$;
  - `target.index` : vector of length n indicating which observation belongs to which experimental setting;
- Inputs of `gies()` is only the score produced by `new()`:
- Let's see how it works in `R`!

Thank you!

References:

- Chickering, D. M. (2002). *Optimal structure identification with greedy search*. Journal of machine learning research, 3(11), 507-554.
- Hauser, A., & Bühlmann, P. (2012). *Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs*. The Journal of Machine Learning Research, 13(1), 2409-2464.
- Hauser, A., & Bühlmann, P. (2015). *Jointly interventional and observational data: estimation of interventional Markov equivalence classes of directed acyclic graphs*. Journal of the Royal Statistical Society Series B: Statistical Methodology, 77(1), 291-318.
- Ramsey, J., Glymour, M., Sanchez-Romero, R., & Glymour, C. (2017). *A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images*. International journal of data science and analytics, 3, 121-129.