score based causal structural learning: an tutorial of GIES

Yufang Wang

November 2024

1 Introduction

Empirical questions are always interested in finding and/or quantifying the causal relationship. As one of the main methods to answer such empirical questions, causal inference methods (e.g., outcome regression with standardization, inverse probability weighing) requires a predefined causal structure, namely Directed Acyclic Graph (DAG). However, such predefined DAGs are not always accessible from expert knowledge, especially in the following cases: (1) the relationship among some variables is not studied in literature; (2) the relationship among some variables is not agreed in literature. Thus, finding the causal structure, namely casual structural learning (i.e., learning DAG), becomes prominent.

Learning a DAG in literature can be achieved from both observational and interventional data. Observational data are the type of data that is automatically generated from a given system without setting or forcing any variables. In contrast, interventional data are obtained by setting or forcing one or multiple variables by researchers within the system. The forced variables are called manipulated variables. As a result, the information about the directed cause of the manipulated variables is removed from the interventional data. Although the interventional data is preferred in science, the advantage of interventional data over observational data is not very well-established or documented. In this tutorial, we will employ one of the most commonly used algorithms in causal structural learning (i.e., Greedy Intervention Equivalence Search, GIES) to explain what role interventional data play in causal structural learning. That is, the research question in the current tutorial is what roes interventional data play in causal structural learning.

The tutorial begins with some background knowledge, including key terminologies and concepts related to DAGs. Then we move to the assumptions, the actual algorithm, and the algorithm performance metric. With a clear understanding of the GIES, we employed the GIES algorithm on an stimulated observational dataset together with two stimulated interventional datasets. That is, we applied the GIES algorithm on the observational data alone and the combined observational and interventional data, compare and discuss their corresponding results in detail.

2 Background

2.1 DAGs Terminology

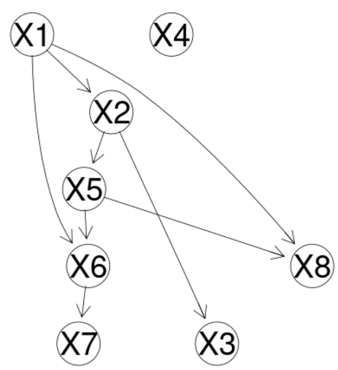
Before moving on to the methodology part, it is worth first mentioning the terminologies and the mathematically meaning of a DAG, which can be seen the following:

An DAG object $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

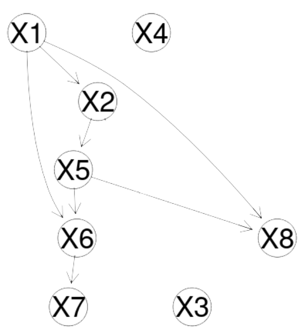
- $\mathcal{V}$ denotes the set of *node* or *vertices*, representing random variables, e.g., X_1, X_2 in Figure 1a.
- $\mathcal{E}$ denotes the set or *arrows* or *directed edges*, representing direct influence, e.g., the direct arrow $\rightarrow$ from X_1 to X_2 in Figure 1a.
- A missing arrow encodes the **absence** of direct influence, e.g., an absent link from X_4 to other nodes, and vice versa in Figure 1a.

- *descendants*: all nodes directly or indirectly caused by focal node, e.g., for the focal node X_5 in Figure 1a, the descendants are X_6, X_7, X_8.
- *child*: all nodes directly caused by the focal node, e.g., for the focal node X_5 in Figure 1a, the descendants are X_6, X_8.
- *ancestors*: all nodes directly or indirectly causing a focal node, e.g., for the focal node X_5 in Figure 1a, the ancestors are X_1, X_2.
- *parents*: all nodes directly causing a focal node, e.g., for the focal node X_5 in Figure 1a, the parents are X_2.
- *confounding variable*: a directed common cause of two variables, e.g., in Figure 1a, X_1 is a confounding variable for nodes X_2 and X_6.

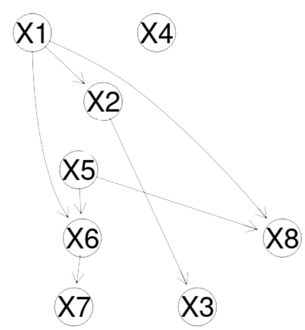
Such an DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can mathematically inform us of the conditionally independency or dependency, which can be used to simplifying/factorizing the joint probability distribution. For instance, the joint probability distribution of all nodes in the DAG given in Figure 1a can be factorized in Equation 1. Such factorization might change in the interventional data. For instance, when manipulating the node $X3$, the factorization of the joint probability distribution becomes Equation 2. The corresponding graphs become those in Figure 1b.



(a) An DAG



(b) The Interventional DAG when X3 is manipulated



(c) The Interventional DAG when X3 is manipulated

Figure 1: A DAG under observational vs. interventional cases

$$\begin{aligned} &P(X_1, X_2, ..., X_7) \\ &= \prod P(X_i|PAR_{G(X_1,X_2,...,X_n)}(X_i) \\ &= P(X_7|X_6)P(X_6|X_5, X_1)P(X_8|X_1, X_2)P(X_3|X_2)P(X_5|X_2)P(X_2|X_1)P(X_1)P(X_4) \end{aligned} \tag{1}$$

where $PAR_{G(X1,X2,...,Xn)}(X_i)$ is the parent(s) of X_i given in G.

$$P(X_1, X_2, ..., X_7)$$
$$= \prod P(X_i | PAR_{G(X_1,X_2,...,X_n)}(X_i)$$
$$= P(X_7|X_6)P(X_6|X_5, X_1)P(X_8|X_1, X_2)P(X_3)P(X_5|X_2)P(X_2|X_1)P(X_1)P(X_4) \quad (2)$$

where $PAR_{G(X1,X2,...,Xn)}(X_i)$ is the parent(s) of X_i given in G.

2.2 Markov Equivalence Class (MEC)

As mentioned before that a DAG mathematically inform us of the way to factorize the joint probability. However, such joint probability factorization and the corresponding DAGs do not always have one-on-one relationship. Sometimes, an identical probability factorization can be informed by a collection of DAGs mathematically, namely MEC (see an example in Figure 2 and details in Chickering, 2002; Pearl et al., 2000; Spirtes et al., 2000). In greater detail, for the given three DAGs in Figure 2, their joint probability can all be factorized identically, i.e., $P(X, Y, Z) = P(Z|Y)P(Y|X)P(X)$, because of the chain rule (i.e., $P(X|Y)P(Y) = P(Y|X)P(X)$). Topologically speaking, MEC is a collection of DAGs which share (1) the same skeleton and (2) the same v-structure. A skeleton is the undirected graph that results from removing all directions (orientations) of the edges in a directed graph. A v-structure (also called "collider") is a specific pattern in a directed graph where two parent nodes point to a common child node, but the parents are not directly connected. Note that DAGs such MECs can only be distinguished via manipulating a certain variable. For instance, in the Figure 2, given that node Y is manipulated but the factorized joint probability distribution keeps the same (i.e, $P(X, Y, Z) = P(Z|Y)P(Y|X)P(X)$), the fork will be the final DAG.

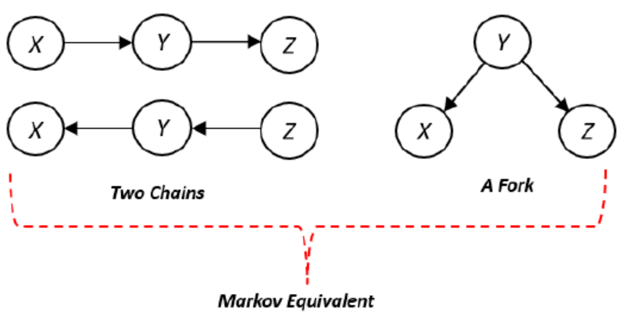


Figure 2: The graphs of Markov Equivalence Class

To serve the purpose of the GIES, we also mimic the procedure of adding edges among nodes and count the corresponding MECs in the following:

1. **Two Nodes:** 1 + 1 MECs
   - **MECs 1:** A and B (no edge, they are independent.
   - **MECs 2:** either $A \rightarrow B$ or $A \leftarrow B$ (There is a directed edge between A and B, which can be represented as an undirected edge $A - B$).
2. **Adding a Third Node to MEC 2 (A and B connected):** 1 + 2 + 2 MEC
   - **MECs 1':** $A - B, C$ (The third node, C, is independent of A and B).
   - **For** $A \rightarrow B$**:**
     - **MECs 2' (No v-structure): Structures:**

$$A \rightarrow B \rightarrow C$$
$$C \rightarrow A \rightarrow B$$
$$C \leftarrow A \rightarrow B$$

– **MECs 3' (One v-structure): Structure:**

$$A \rightarrow B \leftarrow C$$

- **For** $A \leftarrow B$**:**
  – **MECs 2" (No v-structure): Structures:**

$$A \leftarrow B \rightarrow C$$

$$A \leftarrow B \leftarrow C$$

$$C \leftarrow A \leftarrow B$$

  – **MECs 3" (One v-structure): Structure:**

$$C \rightarrow A \leftarrow B$$

Note that when e.g., node B is manipulated, all DAGs having in-bounded edges for node B will be crossed out. That is, the previously mentioned procedure will become:

1. **Two Nodes:** 1 + 1 MECs
   - **MECs 1:** A and B (no edge, they are independent.
   - **MECs 2:** $A \leftarrow B$
2. **Adding a Third Node to MEC 2 (A and B connected):** 1 + 2 MECs
   - **MECs 1':** $A - B, C$ (The third node, C, is independent of A and B).
   - **For** $A \leftarrow B$**:**
     – **MECs 2" (No v-structure): Structures:**

$$A \leftarrow B \rightarrow C$$

$$A \leftarrow B \leftarrow C$$

$$C \leftarrow A \leftarrow B$$

     – **MECs 3" (One v-structure): Structure:**

$$C \rightarrow A \leftarrow B$$

3 Assumptions of the GIES algorithm

With the information of graph terminology, we further move on to the assumptions of GIES as follows:

(1) Causal Faithfulness Assumption: The causal relationships among the nodes are fully reflected in the dataset. This means that the effects of parent nodes on a child node cannot cancel each other out.

(2) Acyclic Assumption: The causal graph is acyclic, meaning there are no cycles. If you follow the directional edges of the graph, you will never return to the starting node.

(3) Causal Markov Assumption: Given its parent nodes, a node is conditionally independent of all other non-descendant nodes in the graph.

(4) Causal Sufficiency Assumption: All relevant variables are included in the dataset, meaning there are no unrecorded confounding variables.

For the real-world data, all of these assumptions are not testable solely from a given dataset. But they are constraints for the DAG that can be obatined from the GIES algorithm are going to obtain from the GIES algorithm. That is, if the underlying causal structure conflicts with these assumptions, the output of GIES is not trustworthy anymore. Regarding the simulated dataset, all the mentioned assumptions can be easily satisfied by generating data accordingly.

4 The GIES algorithm

With all the assumptions made by the GIES algorithm, we now move on to the actual algorithm. Mathematically, causal structural learning aims to identify an underlying DAG that best represents the structures of all variables or nodes in a dataset. This underlying DAG is mathematically represented by the conditional independencies, which are captured through the way of factorizing the joint probability of all variables in a given dataset. As mentioned before, the joint distribution factorization has a one-to-one relationship with an MEC but not necessarily a single DAG. Thus the operation of GIES, especially on the observational data, is also conducted at the level of MEC. In the GIES algorithm, the degree to which a DAG represents these structures is measured by a score function.

4.1 Bayesian Information Criterion (BIC) as score function

In this tutorial, the score function of the GIES was defined by BIC (see in Equation 3 and details in Schwarz, 1978. In the case of multivariate normal distribution, for a given joint probability factorization, the $\hat{L}$ in the BIC can be replaced by Equation 4.

$$\mathrm{BIC} = -2\ln(\hat{L}) + k\ln(n) \tag{3}$$

where:

- $\hat{L}$ is the maximum value of the likelihood function, $L(\theta; \mathbf{X})$, which is defined as $L(\theta; \mathbf{X}) = P(\mathbf{X} \mid \theta)$, where $\mathbf{X}$ represents the observed data, and θ represents the parameters of the model.
- k is the number of free parameters in the model.
- n is the number of observations in the dataset.

$$L(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{X}) = \prod_{i=1}^{n} f(\mathbf{x}_i \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{4}$$

Note that other functions can be used as long as they meet the criteria of consistency, local consistency, and score equivalence (for detailed information, see Meek, 1997).

4.2 The procedure of GIES algorithms

4.2.1 GIES for observational data

With the Bayesian Information Criterion (BIC) as the scoring function to represent the degree of representation for an MEC, the GIES algorithm begins with an empty directed acyclic graph (DAG) and search over MEC spaces in each operation. The operation can be divided into three phases:

1. **Forward Equivalence Search (FES)**: Starting with an empty DAG, the GIES algorithm iteratively adds edges in a greedy fashion by searching among the MEC spaces at each iteration. At each iteration, it selects the edges that locally maximize the BIC score. For instance, for the four nodes in Figure 3, in order to add the first edge, BIC scores over the $\frac{4\times3}{2} + 1 = 7$ MECs at that iteration are calculated and compared. The MECs with highest scores are selected. In a similar fashion, the FES moves to adding the next edge, until the BIC stops increasing.

2. **Backward Equivalence Search (BES)**: Once the forward phase is complete, edges are greedily removed to further optimize the BIC score in a similar way as well. For instance, in Figure 3, with the obtained MEC, to remove an edge, BICs for all possible MECs after removing one edge at that iteration are calculated and compared. The MECs with highest scores are also selected. In a similar fashion, the BES moves to removing the next edge, until the BIC stops increasing.

3. **Turning Phase**: with the MEC obtained from BES, the algorithm attempts to reorient the directions of edges to further optimize the BIC score in a similar way as well. For instance, in Figure 3, to reorient an edge, BICs for all possible MECs after reorientation one edge at that iteration are calculated and compared. The MECs with highest scores are also selected. In a similar fashion, the BES moves to reorienting the next edge, until the BIC stops increasing.

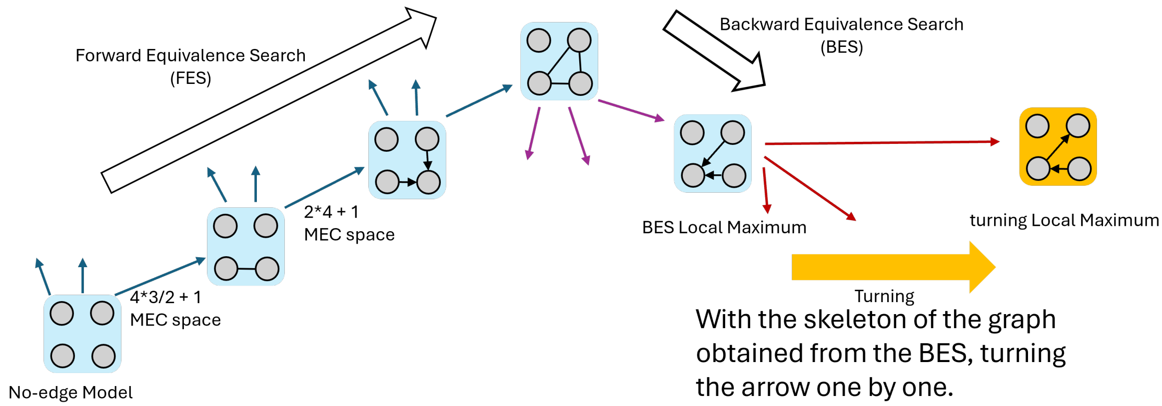


Figure 3: The procedure of the GIES algorithm

Note that since the GIES is operated the MEC spaces, the output of GIES is also an MEC, which might have more than one DAG (Hauser and Bühlmann, 2012).

4.2.2 GIES when aggregating observational data with inteventional data

When aggregating the observational data with inteventional data, the GIES algorithm also follows three phases – FES, BES, and Turning. However, in all three phases—adding, removing, and reorienting edges—decisions must take into account the presence of manipulated variables (Hauser and Bühlmann, 2012).

That is, all edges connected with the manipulated nodes can only be outbound edges. For example, as shown in Figure 4, when adding the first edge, provided the selected MECs is the same as mentioned in the observational data (see in section 4.2.1), it is only possible for the edge to be from the manipulated variables, although the search space of MECs remains the same (i.e., 7). However, when adding the second edge, the search space is reduced to 5, rather than 9, due to the constraints imposed by the manipulated variable. Besides, others are the same with what mentioned in the section of GIES for observational data.

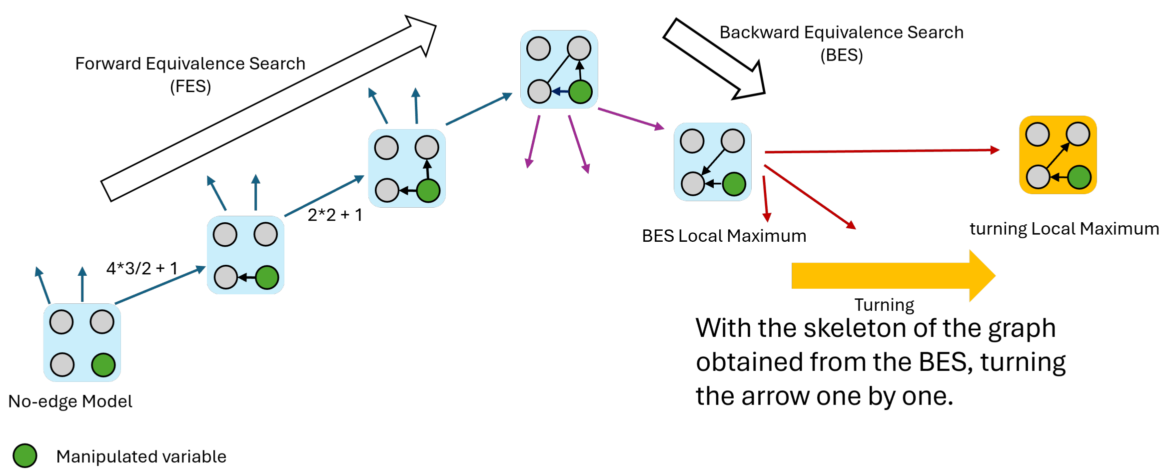


Figure 4: The procedure of the GIES algorithm in the case of interventional data

4.3 SHD as a metric of GIES performance

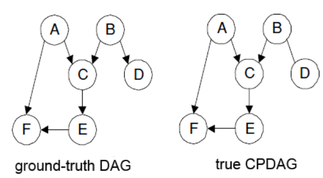
The output of GIES is an MEC when using observational data, or a subset of an MEC when combining interventional and observational data. This output is not necessarily identical to the ground-truth DAG. Therefore, it is important to quantify the similarity between the DAG produced by GIES and the ground-truth DAG. In this tutorial, the similarity is measured using the Structural Hamming Distance (SHD) (Tsamardinos et al., 2006).

However, SHD is calculated based on the corresponding MEC, which may contain multiple DAGs. DAGs within the same MEC can be represented as a single Completed Partially Directed Acyclic Graph (CPDAG), where controversial directed edges within the MEC are converted into undirected edges. That is, before computing SHD, we need first convert the ground-truth DAG into its corresponding CPDAG, namely true CPDAG (see in Figure 5a). Then, the SHD between the output CPDAG G_1 from the GIES with the true CPDAG G_2 is given by Equation 5 (see an example in Figure 5b & 5c).

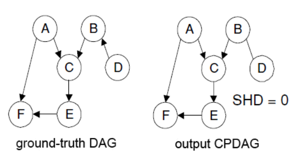
$$\mathrm{SHD}(G_1, G_2) = |E_1 \setminus E_2| + |E_2 \setminus E_1| + |\{(i,j) \in E_1 \cap E_2 : (i,j) \text{ is reversed in } G_2\}| \tag{5}$$

where:

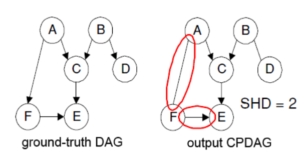
- G_1 and G_2 are the CPDAGs obtained from GIES and true CPDAG being compared.
- E_1 and E_2 are the sets of edges in G_1 and G_2, respectively.
- $|E_1 \setminus E_2|$ is the number of edges present in G_1 but not in G_2, i.e., *deletion*.
- $|E_2 \setminus E_1|$ is the number of edges present in G_2 but not in G_1, i.e., *addition.*.
- $|\{(i,j) \in E_1 \cap E_2 : (i,j) \text{ is reversed in } G_2\}|$ is the number of edges that are reversed in G_2 compared to G_1, i.e., *reverse*. Note that this includes reversed edges and edges that are undirected in one graph and directed in the other.



(a) Converting the ground-truth DAG into true CPDAG



(b) Comparing the output CPDAG to ground-truth DAG



(c) Comparing the output CPDAG to ground-truth DAG

Figure 5: Comparing the output CPDAG to ground-truth DAG is equivalent to comparing to true DAG

5 Algorithm implementation

By now, it should be clear that the role of interventional data in causal structural learning is twofold: (1) to distinguish between/among DAGs within the same MEC; and (2) to reduce the MEC searching space during the FES, BES, and/or turning phases in the GIES algorithm. However, it is still beneficial to implement this algorithm using both purely observational data and a combination of observational and interventional data. To proceed, we will first provide some information about the data used in this tutorial.

5.1 Data

5.1.1 Stimulating data from a given DAG

In the current tutorial, we used simulation data. The data was simulated according to the graph given in Figure 1a. Our simulated datasets are multivariate and normally distributed, whose relationships are linear. This procedure were realized via the $rmvDAG()$ function in *pcalg* R library.

5.1.2 Data used in the current tutorial

However, due to the requirement of this course, we used the simulated data generated by our coach. In greater detail, we received three datasets: the observational dataset, the interventional dataset 1 with manipulating X_3 and the interventional dataset 2 with manipulating X_5. In principle, the three dataset are generated according to the graph given in Figure 1a. However, in interventional datasets, the nodes X3 and X5 are manipulated, respectively. That is, in those dataset, the directed arrow ending up with the manipulated variables is gone (see Figures 1b and 1c, respectively). In additional, the ground-truth DAG given in Figure 1a coincides with its true CPDAG.

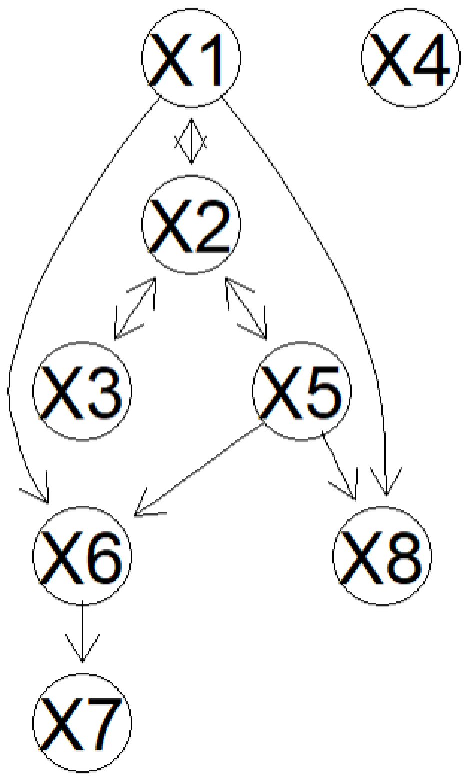
5.2 Data analysis

Before conducting the actual data analysis, we first defined two reformulated datasets for the algorithm implementation: (1) the observational dataset; and (2) the mixed dataset, which is a combination of the observational dataset and the two interventional datasets.

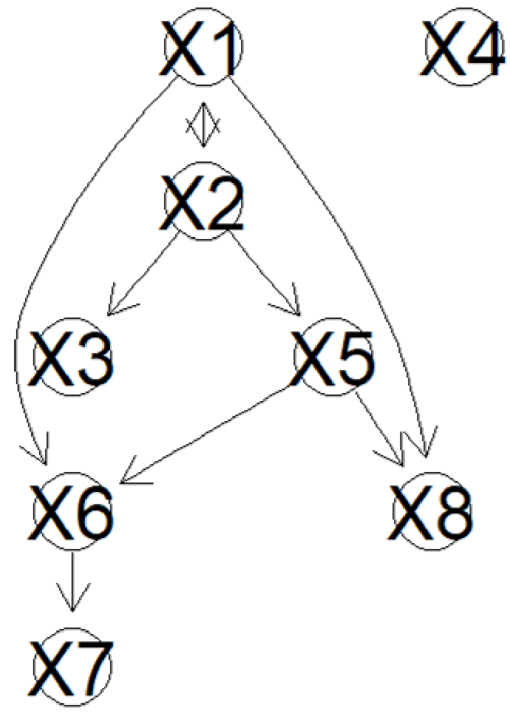
For the reformulated observational dataset, we first used the *new()* function from the *pcalg* library to create a *score* object. With the built *score* object, we applied the GIES algorithm to get the estimated CPDAG from the observational data by employing the *gies()* function from the *pcalg* library (Kalisch et al., 2022). The output CPDAG was compared with the true CPDAG in Figure 1a using the *shd()* function from the *pcalg* library. For the reformulated mixed dataset, the procedure was the same as described for the reformulated observational dataset, except that a different dataset was used (see details in Appendix).

6 Results

As shown in Figure 6a, the output of the GIES algorithm on the reformulated observational dataset has three undirected edges. This results in an SHD value of 3 when comparing the output CPDAG with true CPDAG (see Figure 1a). In contrast, the output of the GIES algorithm on the reformulated mixed dataset has only one undirected edge, leading to an SHD value of 1 when compared with true CPDAG (see Figure 1a).



(a) Output of GIES on observational data



(b) Output of GIES on observational data together with Interventional data when X3 and X5 are manipulated

Figure 6: The output of the GIES on different types of datasets

7 Discussion

In summary, we have found that the combined the observational dataset with interventional dataset helps to distinguish between DAGs within the same MEC. In greater detail, with the given datasets in this tutorial, when the GIES algorithm was applied solely to the observational dataset, the resulting CPDAG had three undirected edges. This ambiguity in the causal structure is reflected in a SHD value of 3 when comparing the output CPDAG with true CPDAG. The undetermined directions indicate that the algorithm could not fully resolve the causal relationships among the variables using observational data alone. This is expected, as observational data often lacks the necessary information to distinguish between/among certain DAGs within the same MEC.

In contrast, when the GIES algorithm was applied to the mixed dataset, which included both observational data and interventional data where variables $X3$ and $X5$ were manipulated, the resulting CPDAG had only one undirected edge. This improvement is reflected in a reduced SHD value of 1 when when comparing the output CPDAG with true CPDAG. The inclusion of interventional data provided additional information that helped to disambiguate the causal relationships, thereby reducing the uncertainty in the estimated causal structure.

These findings underscore the importance of interventional data in enhancing the precision of causal discovery algorithms. Interventions allow for the direct manipulation of variables, providing clearer insights into causal relationships that are not easily discernible from observational data alone. By reducing the search space within the MEC and providing more definitive causal directions, interventional data significantly improves the performance of the GIES algorithm.

8 Conclusion

The comparison between the outputs of the GIES algorithm on observational dataset versus mixed dataset demonstrates that incorporating interventional data can lead to more accurate and precise DAGs. This highlights the value of designing studies that include both observational and interventional components, especially in complex systems where causal relationships are not easily inferred from observational data alone.

9 Limitations & Future work

The limitations of the GIES algorithm in causal structural learning on interventional data include two points: (1) Although we used the BIC to quantify representation and MEC for the ground-truth causal structure, the GIES selected the MECs in a greedy manner. This approach ensures locally optimal solutions but not always globally optimal ones; (2) While incorporating interventional data with observational data can lead to more accurate and precise causal structures, the interventional data is only useful if the manipulated nodes in the CPDAG obtained from the observational data have undirected edges. In this tutorial, as shown in Figure 6a, the nodes $X3$ and $X5$ in the CPDAG obtained from the observational data have undirected edges. Therefore, combining the observational data with interventional data that manipulates $X3$ and $X5$ is beneficial. Conversely, using interventional data that manipulates, for example, $X7$ would not improve the accuracy and precision of the causal structures.

In addition, there are also other limits regarding to the choice of metrics for GIES performance. On the one hand, SHD accounts for both orientation and skeleton when comparing two CPDAGs but is limited to comparing the output CPDAG with true CPDAG. One the other hand, the output CPDAG of the GIES is an estimate based on the given dataset, whose generalization over different datasets requires further study.

References

Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of Machine Learning Research*, *3*, 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.

Kalisch, M., Hauser, A., Maathuis, M., & Mächler, M. (2022). An overview of the pcalg package for r. *R package version*, 2–7.

Meek, C. (1997). *Graphical models: Selecting causal and statistical models* [Doctoral dissertation, Carnegie Mellon University].

Pearl, J., et al. (2000). Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, *19*(2), 3.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, prediction, and search.* MIT Press.

Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, *65*, 31–78.

Appendix: R code

```
# ------ generate simulated data for the given DAGs "toy example"
library(igraph)
library(graph)
library(pcalg)
library(Rgraphviz)
adj_matrix = matrix(c(0, 1, 0, 0, 0, 1, 0, 1,
                      0, 0, 1, 0, 1, 0, 0, 0,
                      0, 0, 0, 0, 0, 0, 0, 0,
                      0, 0, 0, 0, 0, 0, 0, 0,
                      0, 0, 0, 0, 0, 1, 0, 1,
                      0, 0, 0, 0, 0, 0, 1, 0,
                      0, 0, 0, 0, 0, 0, 0, 0,
                      0, 0, 0, 0, 0, 0, 0, 0), nrow = 8, byrow = TRUE
                      )
colnames(adj_matrix) = paste0("X", 1:8); rownames(adj_matrix) = paste0("X", 1:8)
## Create igraph object
igraph_dag = graph_from_adjacency_matrix(adj_matrix, mode = "directed")
# Analyze or manipulate the DAG directly using igraph
plot(igraph_dag, main = "DAG using igraph")
DAG = igraph.to.graphNEL(igraph_dag)
# normal-distribution errors
sim_data_non_normal <- rmvDAG(n = 100, dag = DAG, errDist = "normal")
cpdag = dag2cpdag(adj_matrix) # get the CPDAG for the ground-truth DAG
rm(adj_matrix,igraph_dag)#, DAG

observationalData = read.csv("./Data/simuObs.csv")
targetV3 = read.csv("./Data/targetV3.csv")
targetV5 = read.csv("./Data/targetV5.csv")
IntData = rbind(observationalData, targetV3, targetV5)
IntTarget = list(integer(0), 3, 5)

# ----- using the observational data for causal structural discovery
observationalData = read.csv("./Data/simuObs.csv")
IntData = rbind(observationalData)
IntTarget = list(integer(0))

# define a score object
scoreObs = new("GaussL0penIntScore", data = IntData, targets = IntTarget,
               target.index = c(rep(as.integer(1), nrow(observationalData))))
giesObs.fit <- gies(scoreObs)

# ----- using the observational & interventional data for causal structural discovery

# define a score object
scoreInt = new("GaussL0penIntScore", data = IntData, targets = IntTarget,
               target.index = c(rep(as.integer(1), nrow(observationalData)),
                                rep(as.integer(2), nrow(targetV3)),
                                rep(as.integer(3), nrow(targetV5))))
giesInt.fit <- gies(scoreInt)

# ------- plot the output & compare it with the ground-truth DAG
par(mfrow=c(1,3))
plot(giesObs.fit$essgraph, main = "Estimated CPDAG under \n observational data")
plot(giesInt.fit$essgraph, main = "Estimated CPDAG under \n obserevtional & X3
& X5 \n being manipulatde")
```

```
plot(DAG, main = "True DAG")

essgraphObs = giesObs.fit$essgraph
adj_matrixObs = as(essgraphObs, "matrix")
graph_igraphObs = graph_from_adjacency_matrix(adj_matrixObs, mode = "directed")
graph_nelObs = igraph.to.graphNEL(graph_igraphObs)
shdObs.val = shd(graph_nelObs, DAG)
shdObs.val

essgraphInt = giesInt.fit$essgraph
adj_matrixInt = as(essgraphInt, "matrix")
graph_igraphInt = graph_from_adjacency_matrix(adj_matrixInt, mode = "directed")
graph_nelInt = igraph.to.graphNEL(graph_igraphInt)

shdInt.val = shd(graph_nelInt, DAG)
shdInt.val

shdInt.c = shd(graph_nelObs, graph_nelInt)
shdInt.c
```