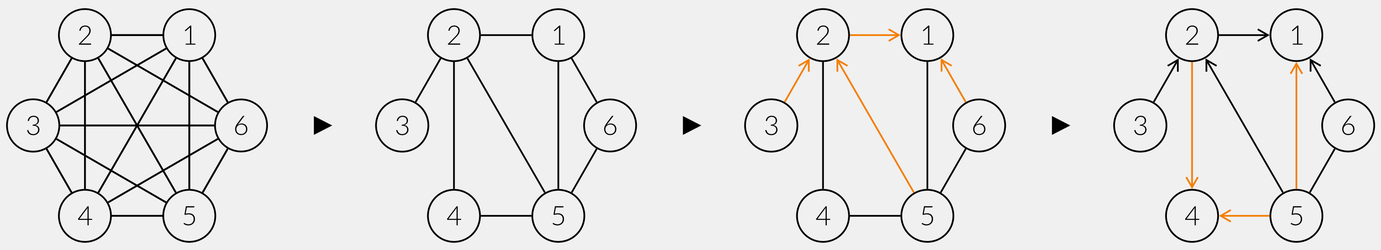
Constraint-based Causal Structure Learning with Consistent Separating Sets

Honghao Li, Vincent Cabeli, Nadir Sella, Hervé Isambert

Institute Curie, UMR 168, 26 Rue d'Ulm, 75005, Paris

Introduction

- **Network Inference**: Recover (causal) relationship (gene regulation, cancer treatment, etc.) from real data (biological experiment, clinical database).
- **PC-stable algorithm[1]** is a constraint-based network inference algorithm based on conditional independence tests.



- For each edge removed from the skeleton, its separating set may not be consistent with respect to the final graph.

Consistent: $(1 \perp\!\!\!\perp 4 \mid 2,5), (1 \perp\!\!\!\perp 3 \mid 2,5), (3 \perp\!\!\!\perp 4 \mid 2,5), (4 \perp\!\!\!\perp 6 \mid 5)$;
inconsistent type I: $(2 \perp\!\!\!\perp 6 \mid 3)$ There is no path between 2 and 6 that goes through 3;
inconsistent type II: $(3 \perp\!\!\!\perp 6 \mid 1)$ Vertex 1 is a child of vertex 6 and is not a neighbor of vertex 3.

Motivation and objective

Weakness of PC-algorithm: Lack of robustness against sampling noise for finite dataset. This can lead to

- tendency to uncover spurious conditional independences: false conditional independence;
- false orientation based on erroneous skeleton.

These errors can be shown by comparing the learnt graph with the true graph, or often, in the absence of the latter, by the **inconsistent separating sets** in the learnt graph.

Focusing on the inconsistency of separating set, we want to

- Make sure all separating sets used to remove an edge remain consistent with respect to the final graph;
- Retain the same level of performance (in terms of precision and recall) with respect to original PC algorithm;
- Resonable time complexity.

Definitions and notations

Given a graph $\mathcal{G}(\boldsymbol{V}, \boldsymbol{E})$ and a set of variables $\{X, Y, Z\} \subseteq \boldsymbol{V}$, let γ_{XY}^{Z} denote a path between X and Y that goes through Z.

Definition 1 (Consistent set[3]).

$$\text{Consist}(X, Y \mid \mathcal{G}) = \{\, Z \in \text{adj}(X) \setminus \{Y\} \mid 1.\ \text{at least one path } \gamma_{XY}^{Z} \text{ exists in } \mathcal{G};\ 2.\ Z \text{ is not a child of } X \text{ in } \mathcal{G} \,\}$$

Note that for an undirected graph, the second condition is always satisfied.

Empirical evaluation

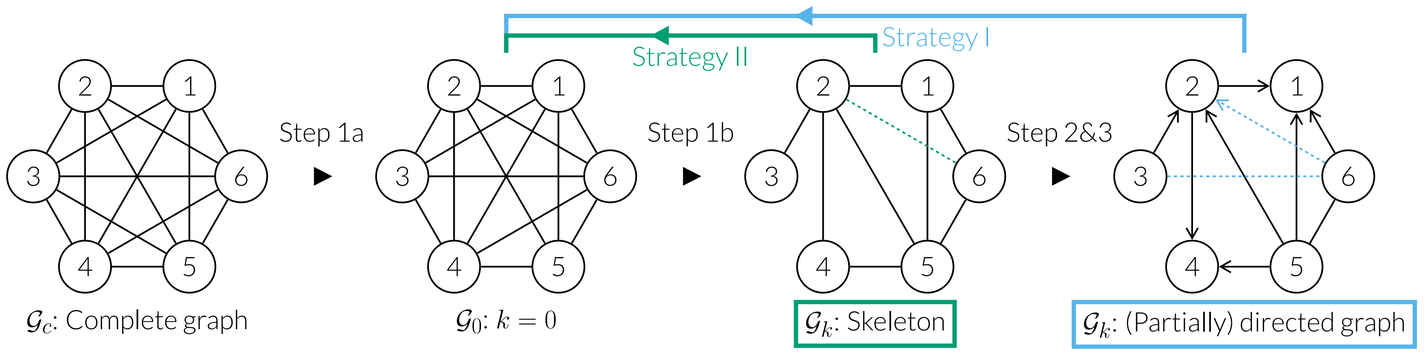
We conducted a series of benchmark structure learning simulations to

- quantify the fraction of inconsistent separating sets predicted by the original PC-stable algorithm (Figure 1);
- compare the performance of the original PC-stable, orientation-consistent PC-stable (Strategy I) and skeleton-consistent PC-stable (Strategy II) algorithms for different significance levels α, in terms of the precision and recall of the adjacencies found in the inferred graph with respect to the true skeleton (Figure 2).

More specifically:

- In figure 1, the data sets were generated with TETRAD[4] as scale-free DAGs with 50 nodes using a preferential attachment model and orienting its edges based on a random topological ordering;
- In figure 2, additional data sets were generated from the standard benchmarks Hepar2 and Barley from the BNlearn repository[5];
- All reconstructions were performed with (modified) pcalg[2]'s PC-stable implementation.

Algorithms: two strategies[3]



Step 1a: Remove edges with unconditional independence;
Step 1b: Remove edges with conditional independence;
Step 2&3: Identify V-structures & Propagation;
Dashed edges mark the difference between two successive iterations.

$k \leftarrow 0$
repeat
  $k \leftarrow k+1$; $\mathcal{G}_k \leftarrow \mathcal{G}_0$
  for all edges (X, Y) in $\mathcal{G}_k$ **do**
    if $(X \perp\!\!\!\perp Y \mid Z)$ **and** $Z \subseteq \text{Consist}(X, Y \mid \mathcal{G}_{k-1})$ **then**
      Remove (X, Y)
    end if
  end for
  Identify V-structures
  Propagation
until loop detected, *i.e.*, $\exists n > 0, \mathcal{G}_{k-n} = \mathcal{G}_k$
$\mathcal{G} \leftarrow \bigcup (\mathcal{G}_j)_{j=k-n}^{k}$

Strategy I : orientation-consistent

$k \leftarrow 0$
repeat
  $k \leftarrow k+1$; $\mathcal{G}_k \leftarrow \mathcal{G}_0$
  for all edges (X, Y) in $\mathcal{G}_k$ **do**
    if $(X \perp\!\!\!\perp Y \mid Z)$ **and** $Z \subseteq \text{Consist}(X, Y \mid \mathcal{G}_{k-1})$ **then**
      Remove (X, Y)
    end if
  end for
until loop detected, *i.e.*, $\exists n > 0, \mathcal{G}_{k-n} = \mathcal{G}_k$
$\mathcal{G} \leftarrow \bigcup (\mathcal{G}_j)_{j=k-n}^{k}$
Identify V-structures
Propagation
for all removed edges (X, Y) in $\mathcal{G}$ **do**
  $\text{Sepset}(X, Y \mid \mathcal{G}) \leftarrow \text{Sepset}(X, Y \mid \mathcal{G}_k)$
  if $\text{Sepset}(X, Y \mid \mathcal{G}) \notin$ **Consist**$(X, Y \mid \mathcal{G})$ **then**
    Add undirected edge (X, Y) to $\mathcal{G}$
  end if
end for

Strategy II : skeleton-consistent

Results

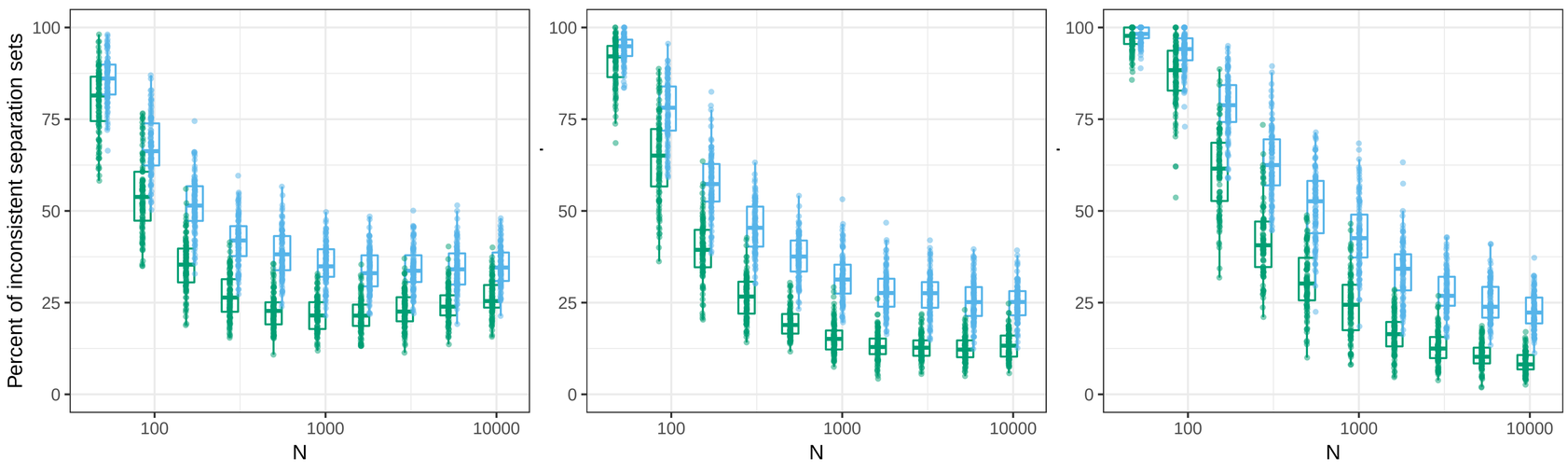


Figure 1. **Sepset inconsistency of the original PC-stable algorithm.** The fraction of inconsistent separating sets with respect to the skeleton (red) or CPDAG (blue) obtained with the original PC-stable algorithm with a fixed $\alpha = 0.05$ is displayed for increasing sample size N. Data-sets were generated from 50 scale-free graphs of 50 nodes and $d(G) = 1.6$ with different parent-child interaction strengths: strong (left), medium (middle) and weak (right).

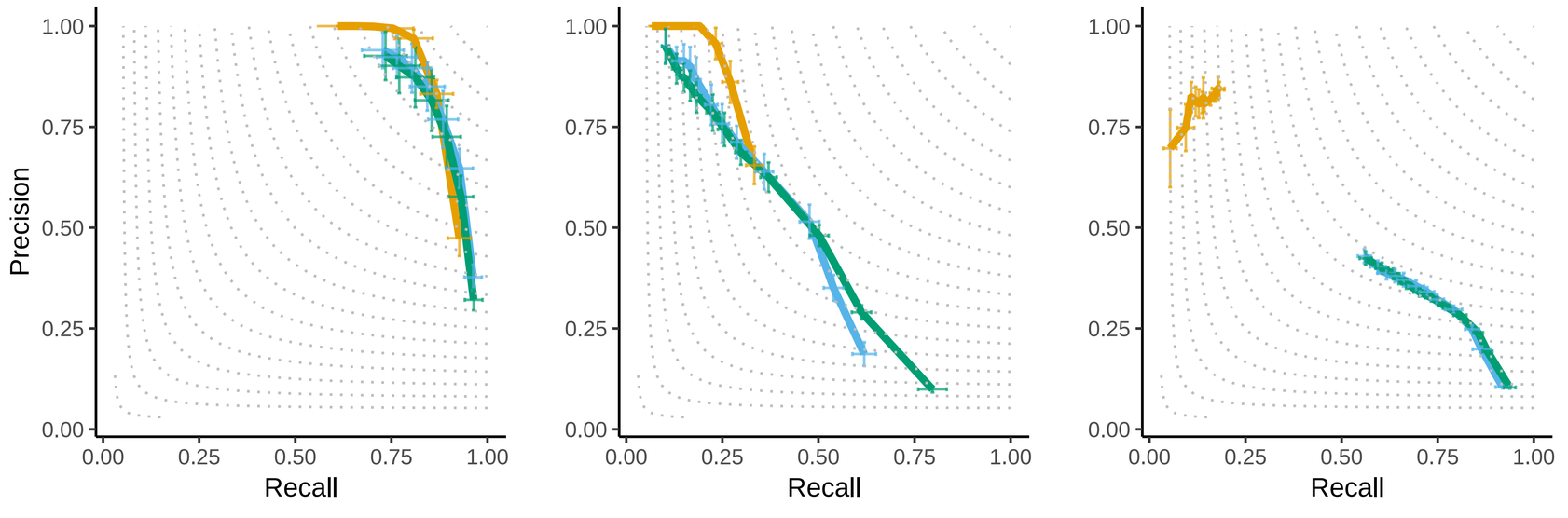


Figure 2. **Precision-recall curves for original PC-stable (yellow), skeleton-consistent PC-stable (green) and orientation-consistent PC-stable (blue)** Mean performances and standard deviations (error bars) obtained over 100 networks are shown for 7 values of the (conditional) independence significance threshold α between 10^{-5} and 0.2. Data-set with N=500 samples were generated from the same graph as in Figure 1 middle with medium interactions (left); Data-sets with N=1000 samples were generated for the standard benchmarks Hepar2 (middle) and Barley (right).

Conclusion

- We propose and implement simple modifications of the PC algorithm also applicable to any PC-derived constraint-based methods, in order to enforce the consistency of the separating sets of discarded edges with respect to the final graph, which is an actual shortcoming of constraint-based approaches;
- Enforcing separating set consistency is shown to significantly improve the sensitivity (recall) of constraint-based methods, while achieving equivalent or better overall structure learning performance;
- Ensuring the consistency of separating sets improves the interpretability of the inferred graph;
- One can either use separating set consistency of the skeleton to help determine the orientations (Strategy I) or use separating set consistency taking into account orientations to help reject inconsistent separating sets (Strategy II).

References

[1] Diego Colombo and Marloes H. Maathuis.
Order-independent constraint-based causal structure learning.
Journal of Machine Learning Research, 15:3741–3782, 2014.

[2] Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H Maathuis, and Peter Bühlmann.
Causal inference using graphical models with the R package pcalg.
J. Stat. Softw., 47(11):1–26, 2012.

[3] Honghao Li, Vincent Cabeli, Nadir Sella, and Hervé Isambert.
Constraint-based causal structure learning with consistent separating sets.
In *Advances in Neural Information Processing Systems 33*. Curran Associates, Inc., 2019.

[4] Richard Scheines, Peter Spirtes, Clark Glymour, Christopher Meek, and Thomas Richardson.
The tetrad project: Constraint based aids to causal model specification.
Multivariate Behavioral Research, 33(1):65–117, 1998.

[5] M. Scutari.
Learning Bayesian Networks with the bnlearn R Package.
Journal of Statistical Software, 35(3):1–22, 2010.