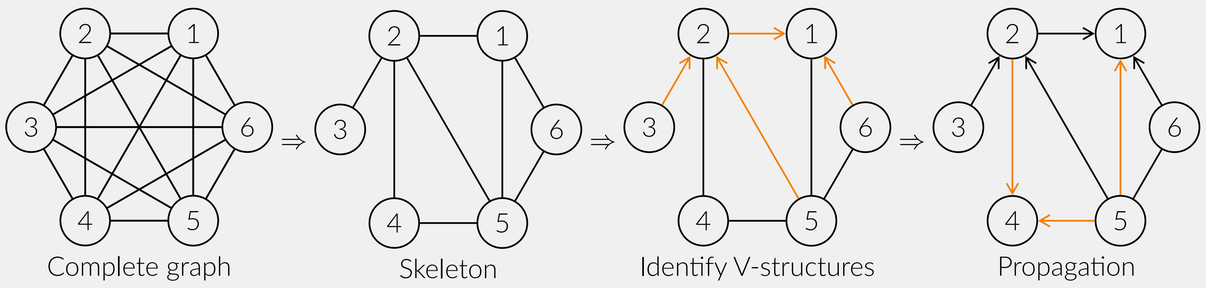
PC algorithm with consistent separating sets

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: $(3 \perp\!\!\!\perp 6 \mid 1)$ The vertex 1 is a descendant of vertex 6 and 3.

Definitions and notations

Given a (directed) 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 (Skeleton-consistent sets)**.** The set of consistent vertices with respect to (X, Y) and the skeleton of $\mathcal{G}$ is defined as:

$$\textbf{Conskel}(X, Y \mid \mathcal{G}) = \{ Z \in \boldsymbol{V} \setminus \{ X, Y \} \mid \text{at least one path } \gamma_{XY}^{Z} \text{ exists} \}.$$

Definition 2 (Orientation-consistent sets)**.** The set of consistent vertices with respect to (X, Y) and $\mathcal{G}$ is defined as:

$$\textbf{Consist}(X, Y \mid \mathcal{G}) = \{ Z \in \textbf{Conskel}(X, Y \mid \mathcal{G}) \mid Z \text{ is a non-descendant of } X \text{ or } Y. \}$$

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, skeleton-consistent PC-stable (Strategy I) and orientation-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:

- The underlying DAGs were generated with TETRAD[3] as scale-free DAGs with 50 nodes using a preferential attachment model and orienting its edges based on a random topological ordering;
- Data-sets were simulated with linear structural equation models for three settings: strong, medium and weak interactions;
- Reconstruction benchmarks were performed with (modified) pcalg[2]'s PC-stable implementation.

Motivation and objective

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

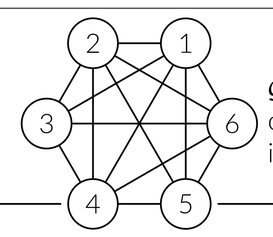
- tendency to uncover spurious conditional independences: false conditional independence;
- false orientation based on erroneous skeleton.

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.

Algorithms: two strategies



$\mathcal{G}_{k=0}$: edges between pairs of nodes that are unconditionally independent are removed ($(3 \perp\!\!\!\perp 5)$)

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 $Z \subseteq$ Conskel$(X, Y \mid \mathcal{G}_{k-1})$ **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

Strategy I : skeleton-consistent

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 $Z \subseteq$ 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 II : orientation-consistent

for all removed edges (X, Y) in $\mathcal{G}$ **do**
  Sepset$(X, Y \mid \mathcal{G}) \leftarrow$ Sepset$(X, Y \mid \mathcal{G}_k)$
  if 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

Results

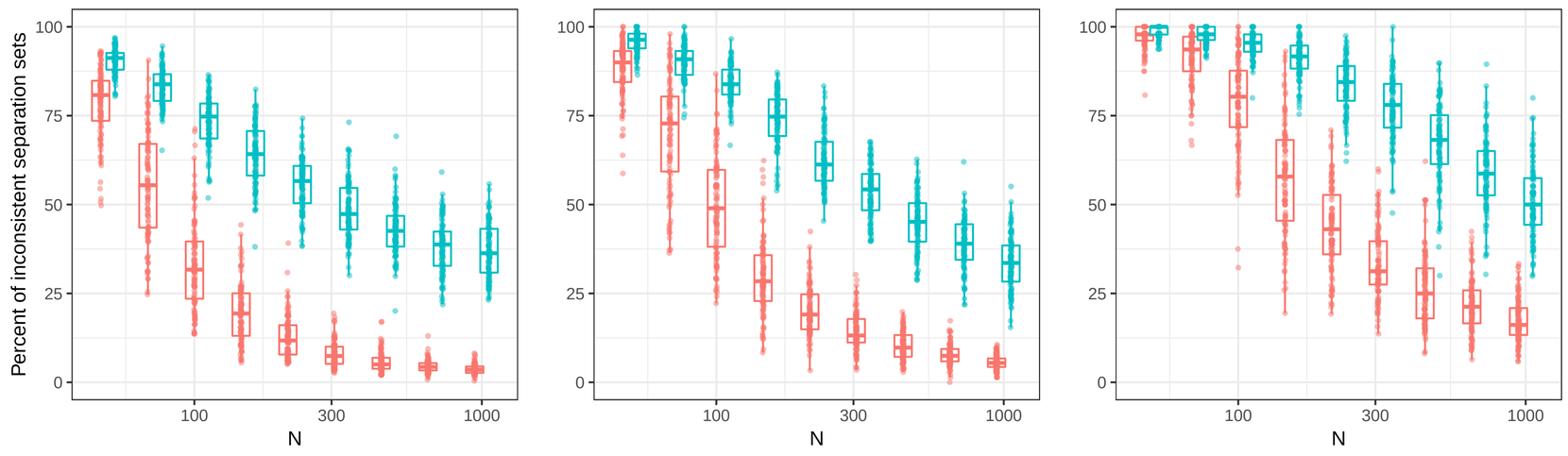


Figure 1. **Sepset inconsistency of the original PC-stable algorithm.** In each subplot the fraction of inconsistent separating sets with respect to the skeleton (red) or CPDAG (blue) obtained with the original PC-stable algorithm with a fixed $\alpha = 0.05$ is displayed for increasing sample size N. 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).

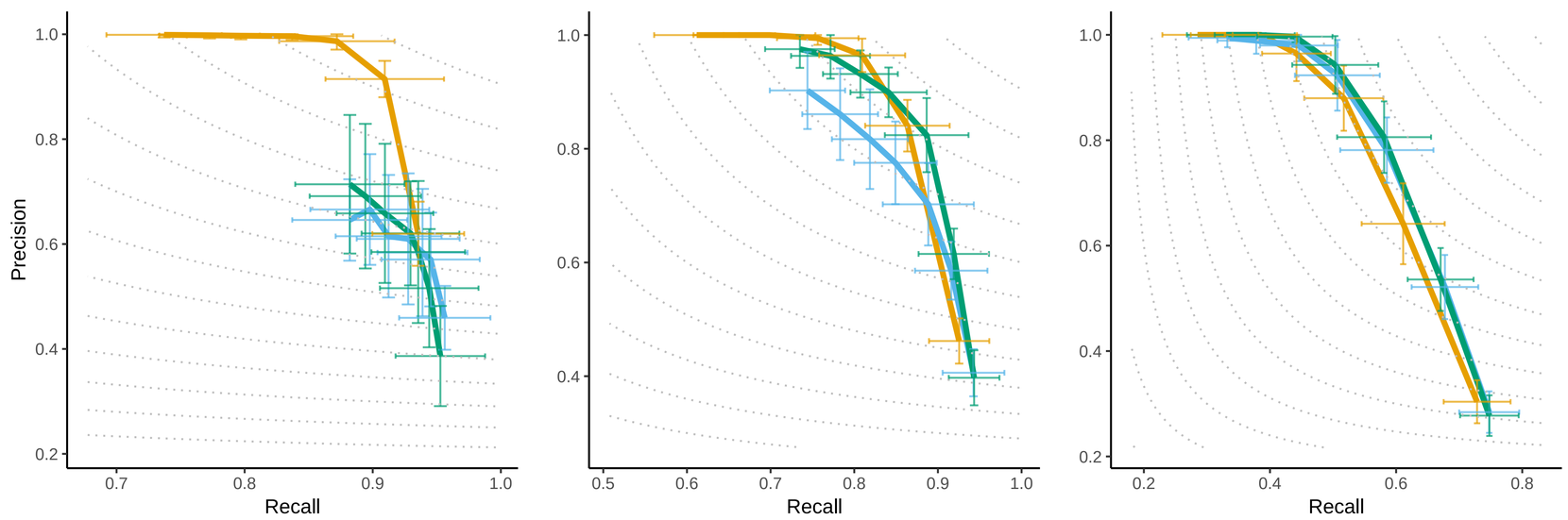


Figure 2. **Precision-recall curves for the original PC-stable (yellow), skeleton-consistent PC-stable (blue) and orientation-consistent PC-stable (green)** The mean performances and standard deviations (error bars) obtained over 20 networks are shown for 7 values of the (conditional) independence significance threshold α between 10^{-5} and 0.2. Data-sets with N=500 samples were generated from the same graphs as in Figure 1 with strong (left), medium (middle) and weak (right) interactions.

Conclusion

- We propose and implement simple modifications of the PC algorithm, which are also applicable to any PC-derived constraint-based methods, in order to enforce the consistency of the separating sets of removed edges with respect to the final graph, which is an actual shortcoming of constraint-based approaches;
- Enforcing sepset consistency is shown to significantly improve the sensitivity of constraint-based methods, while achieving equivalent or better overall structure learning performance,
- One can either use sepset consistency of the skeleton to help determine the orientations (Strategy I) or use sepset consistency taking into account orientations to help reject inconsistent sepsets (Strategy II). The former approach tends to yield better performance with the setting of the PC-stable algorithm used here but this is expected to be dependent on the specific settings used for conditional independence test, orientation and propagation rules, in different constraint-based methods.

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.
Journal of Statistical Software, 47(11):1–26, 2012.

[3] 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.