
A Single Iterative Step for Anytime Causal Discovery

Raanan Y. Rohekar
Intel Labs
raanan.yehezkel@intel.com

Yaniv Gurwicz
Intel Labs
yaniv.gurwicz@intel.com

Shami Nisimov
Intel Labs
shami.nisimov@intel.com

Gal Novik
Intel Labs
gal.novik@intel.com

Abstract

We present a sound and complete algorithm for recovering causal graphs from observed, non-interventional data, in the possible presence of latent confounders and selection bias. We rely on the causal Markov and faithfulness assumptions and recover the equivalence class of the underlying causal graph by performing a series of conditional independence (CI) tests between observed variables. We propose a single step that is applied iteratively, such that the independence and causal relations entailed from the resulting graph, after any iteration, is correct and becomes more informative with successive iteration. Essentially, we tie the size of the CI condition set to its distance from the tested nodes on the resulting graph. Each iteration refines the skeleton and orientation by performing CI tests having condition sets that are larger than in the preceding iteration. In an iteration, condition sets of CI tests are constructed from nodes that are within a specified search distance, and the sizes of these condition sets is equal to this search distance. The algorithm then iteratively increases the search distance along with the condition set sizes. Thus, each iteration refines a graph, that was recovered by previous iterations having smaller condition sets—having a higher statistical power. We demonstrate that our algorithm requires significantly fewer CI tests and smaller condition sets compared to the FCI algorithm. This is evident for both recovering the true underlying graph using a perfect CI oracle, and accurately estimating the graph using limited observed data.

1 Introduction

Causality plays an important role in many sciences, such as social sciences, epidemiology, and finance. (Pearl 2010; Spirtes 2010). Understanding the underlying mechanisms is crucial for tasks such as explaining a phenomenon, predicting, and decision making. Pearl (2009) provided a machinery for automating the process of answering interventional and (retrospective) counterfactual queries even when only observed data is available, and determining if a query cannot be answered given the available data type (identifiability). This requires a knowledge about the true underlying causal structure; however, in many real-world situations, this structure is unknown. There is a large body of literature on recovering causal relations from observed data—causal discovery (Spirtes et al. 2000; Peters et al. 2017; Rohekar et al. 2018; Yehezkel and Lerner 2009; Cooper and Herskovits 1992; Chickering 2002; Shimizu et al. 2006; Hoyer et al. 2009), differing in the assumptions upon which they rely. In this work we assume a directed acyclic graph (DAG) for the underlying causal structure and focus on learning it from observational data. Furthermore, we assume the causal Markov and faithfulness assumptions, and consider recovering the structure by performing a series of

conditional independence (CI) tests (Spirtes et al. 2000). In this setting the true DAG is statistically indistinguishable from many other DAGs. Moreover, when considering the possible presence of latent confounders and selection bias (no causal sufficiency), the true DAG cannot be recovered. Instead, Richardson and Spirtes (2002) proposed the Maximal ancestral graph (MAG), which represents independence relations among observed variables, and the partial ancestral graph (PAG), which is a Markov equivalence class of MAGs—a set of MAGs that cannot be ruled out given the observed independence relations.

Recently, causal identification was demonstrated for PAG models (Jaber et al. 2018; Jaber et al. 2019), which is a more practical use of these models. That is, using only observed data and no prior knowledge on the underlying causal relations, some identification and causal queries can be answered.

In this paper, we address the problem of learning a PAG (latent confounders and selection bias may be present) such that interrupting the learning process results in a correct PAG, that is, all the entailed independence and causal relations are correct, although it can be less informative. This anytime property is important in many real-world settings where it is desired to recover as many causal relations as possible under limited available data and compute power.

2 Related Work

Causal discovery in the potential presence of latent confounders and selection bias requires placing additional assumptions. In this paper we assume the causal Markov assumption (Pearl 2009), faithfulness (Spirtes et al. 2000), and a DAG structure for the underlying causal relations. In this setting, several causal discovery algorithms have been proposed, FCI (Spirtes et al. 2000), RFCI (Colombo et al. 2012), FCI+ (Claassen et al. 2013), and GFCI (Ogarrio et al. 2016). Limitations of FCI have been reported previously where it tends to erroneously exclude many edges that are in the true underlying graph, and it requires many conditional independence (CI) tests with large condition sets. The GFCI algorithm employs a greedy score-based approach to improve the accuracy for small data sizes (small sample). However, it requires additional assumptions for justifying the score function that it uses. The RFCI algorithm alleviates computational complexity by avoiding the last stage. This stage requires many CI tests having large condition sets. Although it is sound (outputs correct causal information) it is not complete (some MAGs in the equivalence class can be ruled out given the data).

Similarly to the FCI and FCI+ algorithms, we consider a procedure that is sound and complete in the large sample limit (or when a perfect conditional independence oracle is used). However, these algorithms, for a MAG, treat nodes that are m -separated by adjacent nodes differently from nodes that are m -separated by a minimal separating set that includes nodes outside the neighborhood. In contrast, we treat all possible d -separating sets in a similar manner, and employ an iterative procedure, similar to the one used by the PC algorithm (Spirtes et al. 2000). This allows our method to be interrupted at any iteration, returning a correct graph, similarly to the anytime FCI algorithm. (Spirtes 2001). We call the proposed algorithm iterative casual discovery, ICD.

3 Anytime Iterative Discovery of Causal Relations

We will start by describing notations, definitions and assumptions. Then, we will describe the proposed ICD algorithm and prove its correctness. We conclude by discussing its computational complexity.

3.1 Preliminaries

A causal DAG, \mathcal{D} , over nodes $\mathbf{V} = \mathbf{O} \cup \mathbf{S} \cup \mathbf{L}$ is denoted by $\mathcal{D}(\mathbf{O}, \mathbf{S}, \mathbf{L})$, where \mathbf{O} , \mathbf{S} , and \mathbf{L} represent disjoint sets of observed, selection, and latent variables, respectively. We use a ancestral graph (Richardson and Spirtes 2002) to model the conditional independence relations among the observed variables \mathbf{O} in the causal DAG \mathcal{D} . This class of graphical models is useful since for every causal DAG there exists a unique maximal ancestral graph (MAG). In this setting, our method is aimed at recovering the MAG of the true underlying DAG.

We assume the causal Markov and faithfulness assumption, and for CI testing we use either a perfect oracle (independence is derived directly from d-separation in the true DAG), or a statistical independence test (e.g., partial correlation) using observed, non-experimental data. In this setting, the MAG cannot be fully recovered and only a Markov equivalence class, represented by a partial ancestral graph (PAG), can be recovered. In this graph, an invariant edge-mark is denoted by an empty circle ‘—o’. That is, there exists at least one MAG in the equivalence class that has a tail edge-mark and at least one MAG that has an arrowhead at that edge tip.

Definition 1 (O-equivalence (Spirtes et al. 2000)). *Two DAGs, $\mathcal{D}_i(\mathbf{O}, \mathbf{S}_i, \mathbf{L}_i)$ and $\mathcal{D}_j(\mathbf{O}, \mathbf{S}_j, \mathbf{L}_j)$ are said to be **O-equivalent** if and only if*

$$\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | (\mathbf{Z} \cup \mathbf{S}_i) \text{ in } \mathcal{D}_i \iff \mathbf{X} \perp\!\!\!\perp \mathbf{Y} | (\mathbf{Z} \cup \mathbf{S}_j) \text{ in } \mathcal{D}_j,$$

for disjoint subsets \mathbf{X} , \mathbf{Y} , and \mathbf{Z} of \mathbf{O} .

That is, the observed d-separation relations in two O-equivalent DAGs, \mathcal{D}_i and \mathcal{D}_j , are identical. Spirtes (2001) defines equivalence considering an *n-oracle* for testing conditional independence. It returns “dependence” if the condition set size is larger than n , otherwise d-separation is tested and returned. Thus, *n-O-equivalence* class is defined by using the *n-oracle* instead of d-separation in Definition 1.

In an MAG \mathcal{M} , node X is in **D-Sep**(A, B) if and only if $X \neq A$ and there is an undirected path between A and X such that every node on the path, except the endpoints, is a collider and is an ancestor of A or B (Spirtes et al. 2000). Spirtes et al. (2000) defined a super-set of D-Sep that can be determined from the skeleton of the PC algorithm and its identified v-structures. This super-set, denoted **Possible-D-Sep**(A, B) (note the capitalization of ‘D’), includes all the nodes connected by a path to A , where every node on this path, except the end-points, is a collider or part of a triangle, hiding its orientation. It is important to note that this definition of **Possible-D-Sep** assumes that all conditional independence relations $X_i \perp\!\!\!\perp X_j | \mathbf{Z}$, such that $\mathbf{Z} \in \text{Parents}_{\mathcal{M}}(\{X_i, X_j\})$ are identified. A definition of a smaller super-set of **D-Sep**, called **Possible-D-Sep**_{path}, was given by Colombo et al. (2012), where there is a path connecting every node $Z \in \text{Possible-D-Sep}_{\text{path}}(A, B)$ to B . We take special interest in the path that connects any $Z \in \text{Possible-D-Sep}(A, B)$ to A and define it as follows.

Definition 2 (PDS-path). *A possible-D-Sep-path (PDS-path) from A to Z with respect to B , denoted $\Pi_B(A, Z)$, is an undirected path $\langle A, \dots, Z \rangle$ such that*

1. for every sub-path $\langle U, V, W \rangle$ of $\Pi_B(A, Z)$, V is a collider or $\{U, V, W\}$ forms a triangle, and
2. W is on a path between A and B .

3.2 Proposed Algorithm

We propose a single step that is called iteratively, for recovering the underlying equivalence class, represented by a PAG. Each iteration is parameterized by r ($r \in \{0, \dots, |\mathbf{O}| - 1\}$), and given a PAG returned by the previous iteration with $r - 1$, each pair of connected nodes A and B are tested for independence conditioned on a set $\mathbf{Z} \subset \mathbf{O}$, such that

1. $|\mathbf{Z}| = r$, and
2. the length of the shortest PDS-path from $Z \in \mathbf{Z}$ to either A or B is less than r .

By parameter r , we bind the condition set size to its distance from the tested nodes. That is, the condition set size is bounded by the shortest PDS-path length connecting its nodes to the tested nodes (A and B). Only nodes that are at most r edges away, on a PDS-path, from A or B are considered. Then, identified v-structures are oriented and the iteration concludes by repeatedly applying orientation rules until no more edges can be oriented (Spirtes et al. 2000; Zhang 2008). The full procedure is given in Algorithm 1

For the first iteration, $r = 0$ and condition sets of CI tests are empty. In the second iteration $r = 1$ and only nodes that are adjacent to the tested nodes are included in the condition set (PDS-path length limit is one edge). Only in succeeding iterations nodes that are not adjacent to the tested nodes may be included.

Algorithm 1: Anytime Iterative Discovery of Causal Relations (ICD algorithm)

Input:

\mathcal{G} : Initial PAG over observed variables \mathbf{O} (default: \mathcal{G} is fully connected with o—o edges)
 r_0 : Initial condition set size for CI tests (default: $r_0 = 0$)
 n : Desired n -representing PAG, largest condition set size to use (default: $n = |\mathbf{O}| - 2$)
 Ind: a conditional independence oracle

Output:

A PAG for n - \mathbf{O} -equivalence class

```

1 Function Main( $\mathcal{G}, r_0, n$ ):
2   for  $r$  in  $\{r_0, \dots, n\}$  do
3      $(\mathcal{G}, done) \leftarrow$  CDIteration( $\mathcal{G}, r$ )
4     if  $done = \text{True}$  then
5       break
6   return  $\mathcal{G}$ 

7 Function CDIteration( $\mathcal{G}, r$ ):
8    $done \leftarrow \text{True}$ 
9   for  $(X * \text{---} * Y)$  in Edges( $\mathcal{G}$ ) do
10     $\{\mathbf{Z}_i\}_i \leftarrow$  PDSep_r( $X, Y, \mathcal{G}, r$ )       $\triangleright$  returns sets of  $r$  nodes in range  $r$ 
11    if  $\{\mathbf{Z}_i\}_i \neq \emptyset$  then
12       $done \leftarrow \text{False}$ 
13      for  $\mathbf{Z}'$  in  $\{\mathbf{Z}_i\}_i$  do
14        if Ind( $X, Y | \mathbf{Z}'$ ) then
15          remove edge  $(X * \text{---} * Y)$  from  $\mathcal{G}$ 
16          break
17  orient edges in  $\mathcal{G}$ 
18  return  $(\mathcal{G}, done)$ 

```

It is important to note that the result of $\text{PDSep}_r(X, Y, \mathcal{G}, r)$, in Algorithm 1-line 10, is an ordered set of possible separating sets $\{\mathbf{Z}_i\}_i$, where each \mathbf{Z}_i contains exactly r nodes. The order in which these sets are assigned to \mathbf{Z}' in Algorithm 1 is crucial for reducing the total number of CI test that are performed. A trivial example for an inefficient order is having a condition set consisting of only parents of A or B last in $\{\mathbf{Z}_i\}_i$. We propose ordering this set according to the average of the shortest PDS-path lengths connecting each node in \mathbf{Z}_i . That is, for a $\mathbf{Z} \subset \text{Possible-D-Sep}_{\text{path}}(X, Y)$, the following value is calculated,

$$\hat{d}_X(\mathbf{Z}) = \frac{1}{|\mathbf{Z}|} \sum_{W \in \mathbf{Z}} \min(|\Pi_Y(X, W)|). \quad (1)$$

Then, the possible separating sets, \mathbf{Z}_i are ordered according to this value. Nevertheless, note that the correctness of the algorithm is invariant to this order.

3.3 Correctness

Our proof relies on the following Lemmas.

Lemma 1. *Let \mathcal{G} be a PAG n -representing $\mathcal{D}(\mathbf{O}, \mathbf{S}, \mathbf{L})$. If $A \perp\!\!\!\perp B | [\mathbf{Z}]$, for $A, B \in \mathbf{O}$ and $\mathbf{Z} \subset \mathbf{O}$, then any node $V \in \mathbf{O}$ on the shortest PDS-path in \mathcal{G} from either A or B to $Z \in \mathbf{Z}$ is also in \mathbf{Z} .*

In words: the shortest path connecting a node to A or B consists of a subset of nodes of the minimal separating set (denoted by square brackets).

Proof. It is easy to see that a PDS-Path $\Pi_B(A, Z)$ in a PAG resulting by any iteration of our algorithm has the following properties:

1. every sub-path $\langle A, \dots, V \rangle$ of $\Pi_B(A, Z)$ is a PDS-Path $\Pi_B(A, V)$ (recursion)
2. every node on $\Pi_B(A, Z)$ is in **Possible-D-Sep**_{path}(A, B)

Thus, a condition set that includes node V should also include all the nodes on the shortest PDS-path. \square

Lemma 2. *Let \mathcal{G} be a PAG n -representing $\mathcal{D}(\mathbf{O}, \mathbf{S}, \mathbf{L})$. The number nodes on the shortest PDS-path connecting every node in \mathbf{Z} to A or B is less than n .*

Proof. From Lemma 1 it directly follows that the condition set size is at least as the length of the shortest PDS-path. \square

Theorem 1. *Let $\mathcal{M}(\mathbf{O})$ be a MAG for a causal DAG $\mathcal{D}(\mathbf{O}, \mathbf{L}, \mathbf{S})$ and let Ind be a conditional independence oracle that returns d -separation relation for \mathbf{O} in \mathcal{D} . If Algorithm 1 is called with Ind as the conditional independence oracle, then every casual feature entailed by the returned PAG \mathcal{G} is also entailed by \mathcal{M} .*

Proof. Spirtes (2001) proposed an anytime algorithm for causal discovery. He proved that a graph constructed only from CI tests having a condition set up to r is correct, although less informative. The skeleton of the graph includes edges only between those nodes that cannot be d -separated conditioned on any subset (not just the parents) having a size bounded by r . He proved that relying on such skeleton ensures that any edge-mark that is oriented by identifying a v -structure and by applying orientation rules is invariant (present in any MAG in the equivalence class). It follows from Lemma 2 that after iteration r of Algorithm 1, every edge between nodes that are d -separated by a minimal subset bounded by size r was removed. Thus, from the proof anytime FCI, after refining the skeleton at iteration r , any edge-mark that is oriented by identifying a v -structure and by applying orientation rules is invariant is correct. \square

3.4 Complexity and Efficiency Analysis

The proposed algorithm consists of two main mechanisms for improving efficiency. First, It gradually performs CI tests of increasing condition set sizes, similarly to the PC algorithm. This is done by including all subsets of **Possible-D-Sep** having a specific size. Secondly, after iteration r all edges are removed between nodes that are m -separated in the underlying MAG by a minimal set of size at most r . Thus, we can complete orienting the graph (using v -structures and orientation rules). Moreover, the definition of **Possible-D-Sep** relies on a strong assumption that the only edges that were removed, are between nodes that are m -separated given their parents in the MAG. In our case, edges between nodes that are m -separated given any minimal subset of size r are removed. This allows considering super-sets of **D-Sep** that are smaller than **Possible-D-Sep**, resulting in fewer CI tests. We expect this to be more dominant for CI tests with large condition sets.

Finally, we note that the computational complexity of the algorithm is $O(n^{2k})$, where n is the number of nodes and k is the maximal in-degree. This is because for every pair of nodes A and B , the number of CI tests is bounded by the number of subsets of **Possible-D-Sep**(A, B) and **Possible-D-Sep**(A, B). That is, a complexity that is polynomial in the number of nodes n given the maximal in-degree k .

4 Experimental Evaluation

We compare the proposed ICD algorithm to the prominent FCI algorithm (Spirtes et al. 2000)—both are anytime, sound, and complete. We empirically compare the complexity, run-time and structural accuracy of the algorithms using two types of CI tests: 1) a perfect, always correct, CI oracle, and 2) a statistical independence test evaluated from finite-size data sets. We follow a procedure, similar to the one described by Colombo et al. (2012), for creating random DAGs and data sets. Specifically, we generate DAGs having $n \in \{15, 20, 25, 35\}$ nodes with a connectivity factor of $\rho = 2$. For sampling a

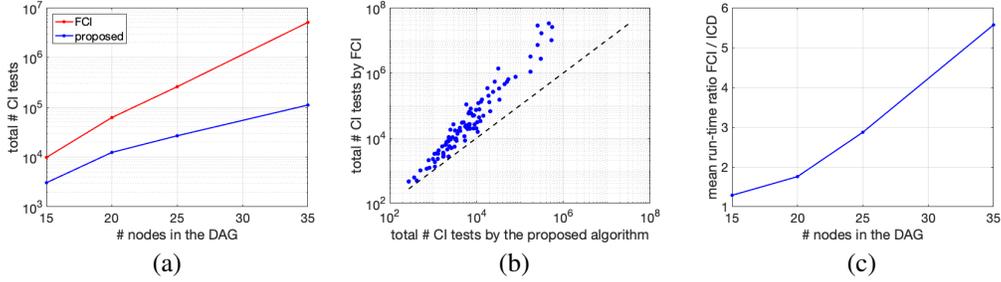


Figure 1: Total number of CI tests and run-time. (a) Average total number of CI tests as a function of the graph size (Y -axis is logarithmic). (b) A scatter plot using all DAGs in the experiment (both X and Y axes are logarithmic). For all the 100 tested DAGs, the proposed ICD algorithm requires fewer CI tests for recovering the true underlying PAG compared to FCI. (c) Mean run-times ratio. Run-time of FCI is divided by the ICD run-time (both are implemented and run on the same platform).

random DAG, having n variables and connectivity factor ρ , an adjacency matrix \mathbf{A} for variables $\mathbf{O} \cup \mathbf{L}$ of DAG $\mathcal{D}(\mathbf{O}, \mathbf{L}, \mathbf{S} = \emptyset)$ is created by independent realization of Bernoulli($\rho/(n-1)$) in the upper triangle. If the resulting DAG is unconnected, we repeat until a connected DAG is sampled. Then, a weight matrix \mathbf{W} for the graph edges is created by sampling from Uniform($[-2, -0.5] \cup [0.5, 2]$) for each non-zero element in \mathbf{A} . Finally, a statistical model is created by setting conditional probabilities $p(X_i | \text{Pa}_{\mathbf{A}}(X_i)) = \mathbf{W}_{(\cdot, i)} \mathbf{A}_{(\cdot, i)}^T + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, 1)$. For each DAG, we sample half of the parentless nodes that have at least two children and set them to be latent set (\mathbf{L}). The remaining nodes are the observed set (\mathbf{O}), and selection bias is not simulated ($\mathbf{S} = \emptyset$).

4.1 Learning the True PAG using a Perfect Independence Oracle

The proposed ICD algorithm is proved to recover the PAG that corresponds to the true underlying DAG when using a perfect oracle for the CI tests. A perfect oracle is implemented such that it returns d -separation relations in the true DAG. In this experiment, we measure the number of unique¹ CI tests and their condition set sizes that are required for recovering the true PAG, and compare it to those that are required by the FCI algorithm.

First, we evaluate the total number of CI tests and run-times for different graph sizes. We randomly generate 25 dags for each of the graphs sizes: 15 nodes, 20 nodes, 25 nodes, and 35 nodes. From Figure 1 (a), we find that compared to FCI, the total number of CI tests required by the proposed method increases more slowly with the graph size than the FCI algorithm (note the logarithmic Y -axis). From Figure 1 (b) it is evident that the proposed algorithm requires fewer CI tests compared to FCI, for all the 100 tested graphs, and that this advantage of ICD is more evident for graphs that require a larger number of CI tests. From Figure 1 (c) we note that the ratio between the run-times of FCI and ICD increases with the graph size.

Next, we analyze the condition set sizes of the required CI tests. In many CI tests for real-world applications, the statistical power decreases and the computational complexity grows exponentially with the condition set size. In Figure 2 for each tested graph size, we depict the average number of required CI tests per condition set size. It is evident that the saving in number of CI tests, compared to FCI, increases with the graph size.

Lastly, we provide a visualization for the distribution of the ratio between the number of CI tests of FCI and ICD Figure 3. It is evident that the ratio increases with the graph size. Moreover, this distribution is skewed towards larger condition sets and that this skew increases with the graph size.

Finally, we note that in our implementation, the significant reduction in the number of CI tests by ICD compared to FCI, led to a proportional reduction in run-time.

¹We implemented a caching mechanism to ensure that the same CI test is not evaluated and counted more than once.

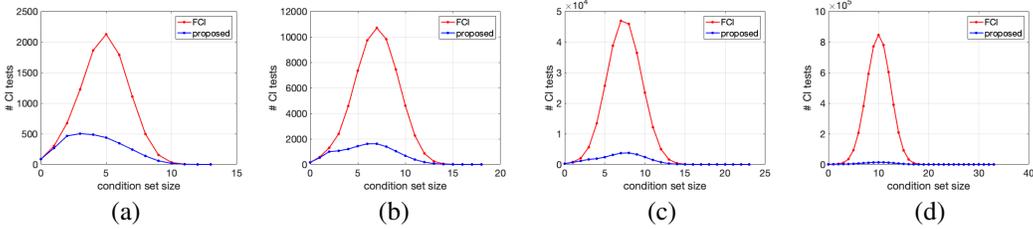


Figure 2: Average number of CI tests per condition set size for different graph sizes. (a) 15 nodes, (b) 20 nodes, (c) 25 nodes, (d) 35 nodes. The proposed method provides a greater saving in CI tests with large condition sets for larger graphs.

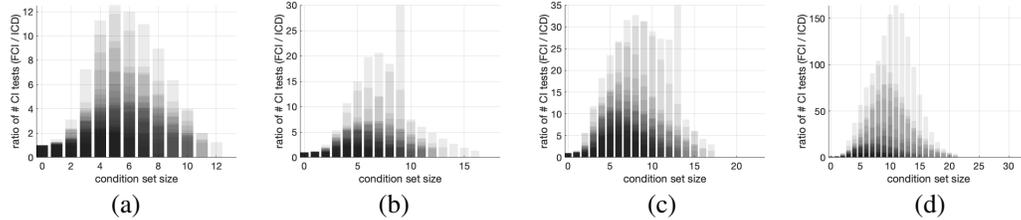


Figure 3: A visualization of the distribution (shading) of the ratio between the number of CI tests required by FCI and the proposed ICD algorithm. For each DAG in the experiment, a semi-transparent bar-plot is created by dividing the FCI number of CI tests (per condition set size) by the ICD number of CI tests. Greater ratios mean fewer CI tests are required by the proposed method compared to FCI for recovering the true underlying PAG. All plots are layered one on top of the other. Thus, dark areas are proportional to high probability for that condition set size at that ratio. Number of graph nodes: (a) 15 nodes, (b) 20 nodes, (c) 25 nodes, (d) 35 nodes. It can be viewed that the distribution is skewed (the shading decreases more gradually) towards larger condition sets.

4.2 Learning a PAG from Observed Data

In this experiment, we use the partial-correlation for testing conditional independence (CI) at $\alpha = 0.01$, and examine two aspects: accuracy of the learned PAG and the required number of CI tests. For evaluating partial correlation we use data (samples for the observed variables \mathbf{O}) in various sizes, $\ell \in [10, 1000]$. For each data size, we randomly sample 1000 DAGs, each having 15 nodes. We learn the PAGs using the proposed ICD algorithm and the FCI algorithm (both are sound and complete).

Required Number of CI Tests. We analyze the total number CI tests required by the proposed algorithm compared to the FCI algorithm. First, using 1000 learned DAGs, we compute the empirical cumulative distribution function F (CDF) for both algorithms on different data sizes. The empirical CDFs for data sizes 100, 200, 500, and 1000 is depicted in Figure 4. We mark $F(t') = 0.9$ with dashed horizontal line and with vertical line we mark t' in 90%. That is, in 90% of the tests (learned DAGs) the algorithm required at most t' CI tests. For data sizes 100, 200, 500, and 1000, the ratios between t' for our proposed method and FCI are 1.16, 1.75, 2.76, and 3.75, respectively. It is evident that the ratio increases rapidly for small data sets and starts to converge to some value (the ratio when using perfect CI oracle). We then evaluate statistical significance using the 2-sample Kolmogorov–Smirnov test. The p-values and the statistic are given in Figure 5 (a) and (b), respectively. The value of the Kolmogorov–Smirnov statistic for $\ell \geq 40$ is consistently high compared to its value for smaller data sizes. We conclude that for $\ell \geq 40$ the CDFs of the number of CI tests is different with statistical significance at $\alpha = 0.05$. Finally, In Figure 5 (c) we plot the number of CI tests as a function of data set size.

Structural Accuracy. We measure three types of structural errors: (‘extra-edges’) the number of edges present in the learned structure but are not in the true PAG, (‘missing-edges’) the number of edges that are in the true PAG but are missing from the learned PAG, and (‘wrong edge-marks’) the number of edge-marks in the learned graph that are different in the true PAG (for edges that exist in both graphs). Compared to the FCI algorithm and for $100 \leq \ell \leq 1000$, the proposed method had fewer, (up to 10% for 15 nodes, and 18% for 35 nodes), ‘missing-edges’ and a similar a number of

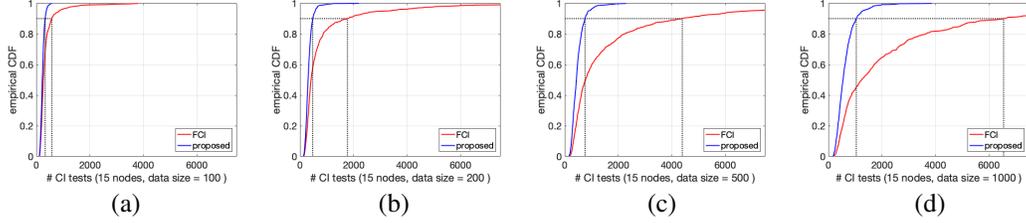


Figure 4: The empirical cumulative distribution function $F(t)$ (CDF) for the number of CI tests t . Evaluated by 1000 randomly generated DAGs. Horizontal line indicates $F(t') = 0.9$, vertical lines indicate the corresponding value t' . That is, 90% of experiment resulting in number of CI tests fewer than t' . Data size used for calculating CI tests: (a) 100, (b) 200, (c) 500, (d) 1000, and the ratios between t' for our proposed method and FCI are 1.16, 1.75, 2.76, and 3.75, respectively.

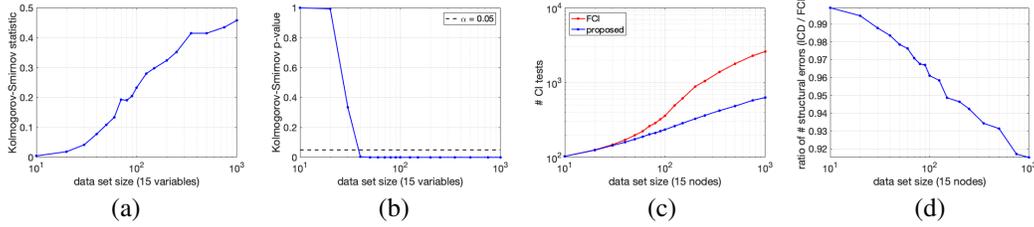


Figure 5: Structural error and the difference in the number of CI tests as a function of the data size. The FCI and ICD algorithms learned 1000 randomly generated DAGs with 15 nodes. The Kolmogorov-Smirnov test for comparing the difference in the CDFs of the number of CI tests: (a) statistic, and (b) p-value. (c) The mean number of CI tests. (d) The ratio between the number of structural errors of FCI and ICD.

'extra-edges' and 'wrong edge-marks'. We report the ratio in total number of structural errors for graphs having 15 nodes, as a function of the data set size in Figure 5 (d).

5 Conclusions

We proposed an anytime, sound, and complete causal discovery algorithm that consists of a single step that is applied iteratively. In every iteration, the skeleton is sought to be refined and edges are oriented using the same orientation rules used by the FCI algorithm. From the experimental results, the proposed method requires significantly fewer CI tests compared to FCI (along with a small improvement in accuracy), especially for large condition sets. This is reflected in significantly shorter run-times.

One important difference of the proposed ICD algorithm from FCI and its related algorithms is that, right from the outset, it considers nodes for the condition set, that are not in the neighborhood of the tested nodes. One might suspect that this could result in a high number of CI tests evaluated by the proposed algorithm compared to FCI. However, the reliance on complete orientation, which is missing in FCI, enables the proposed algorithm to reduce the number of nodes to consider for the condition sets.

We proposed an ordering for the execution of CI tests when testing independence between a pair of nodes. This ordering is proportional to the distance of the condition set from the tested pair. Although the correctness of our algorithm is invariant to this ordering, it enables further reducing the number of CI tests. In our future work we plan to investigate different criteria for ordering.

References

- Chickering, David Maxwell (2002). “Optimal structure identification with greedy search”. In: *Journal of machine learning research* 3.Nov, pp. 507–554.
- Claassen, Tom, Joris M Mooij, and Tom Heskes (2013). “Learning Sparse Causal Models is not NP-hard”. In: *Uncertainty in Artificial Intelligence*. Citeseer, p. 172.
- Colombo, Diego, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson (2012). “Learning high-dimensional directed acyclic graphs with latent and selection variables”. In: *The Annals of Statistics*, pp. 294–321.
- Cooper, Gregory F and Edward Herskovits (1992). “A Bayesian method for the induction of probabilistic networks from data”. In: *Machine learning* 9.4, pp. 309–347.
- Hoyer, Patrik O, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf (2009). “Nonlinear causal discovery with additive noise models”. In: *Advances in neural information processing systems*, pp. 689–696.
- Jaber, Amin, Jiji Zhang, and Elias Bareinboim (2018). “Causal Identification under Markov Equivalence”. In: *34th Conference on Uncertainty in Artificial Intelligence*. Association for Uncertainty in Artificial Intelligence (AUAI), pp. 978–987.
- Jaber, Amin, Jiji Zhang, and Elias Bareinboim (2019). “Causal identification under markov equivalence: Completeness results”. In: *International Conference on Machine Learning*, pp. 2981–2989.
- Ogarrio, Juan Miguel, Peter Spirtes, and Joe Ramsey (2016). “A hybrid causal search algorithm for latent variable models”. In: *Conference on Probabilistic Graphical Models*, pp. 368–379.
- Pearl, Judea (2009). *Causality: Models, Reasoning, and Inference*. Second. Cambridge university press.
- Pearl, Judea (2010). “An introduction to causal inference”. In: *The international journal of biostatistics* 6.2.
- Peters, Jonas, Dominik Janzing, and Bernhard Schölkopf (2017). *Elements of causal inference*. The MIT Press.
- Richardson, Thomas and Peter Spirtes (2002). “Ancestral graph Markov models”. In: *The Annals of Statistics* 30.4, pp. 962–1030.
- Rohekar, Raanan Y, Yaniv Gurwicz, Shami Nisimov, Guy Koren, and Gal Novik (2018). “Bayesian Structure Learning by Recursive Bootstrap”. In: *Advances in Neural Information Processing Systems (NeurIPS)*.
- Shimizu, Shohei, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen (2006). “A linear non-Gaussian acyclic model for causal discovery”. In: *Journal of Machine Learning Research* 7.Oct, pp. 2003–2030.
- Spirtes, Peter (2001). “An Anytime Algorithm for Causal Inference.” In: *AISTATS*.
- Spirtes, Peter (2010). “Introduction to causal inference”. In: *Journal of Machine Learning Research* 11.May, pp. 1643–1662.
- Spirtes, Peter, Clark Glymour, and Richard Scheines (2000). *Causation, Prediction and Search*. 2nd. MIT Press.
- Yehezkel, Raanan and Boaz Lerner (2009). “Bayesian network structure learning by recursive autonomy identification”. In: *Journal of Machine Learning Research (JMLR)* 10.Jul, pp. 1527–1570.
- Zhang, Jiji (2008). “On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias”. In: *Artificial Intelligence* 172.16-17, pp. 1873–1896.