A Hybrid Causal Search Algorithm for Latent Variable Models

Juan Miguel Ogarrio & Peter Spirtes & Joe Ramsey
Department of Philosophy
Carngie Mellon University
Pittsburgh, PA

Abstract

Existing score-based causal model search algorithms such as *GES* (and a speeded up version, *FGS*) are asymptotically correct, fast, and reliable, but make the unrealistic assumption that the true causal graph does not contain any unmeasured confounders. There are several constraint-based causal search algorithms (e.g *RFCI*, *FCI*, or *FCI+*) that are asymptotically correct without assuming that there are no unmeasured confounders, but often perform poorly on small samples. We describe a combined score and constraint-based algorithm, *GFCI*, that we prove is asymptotically correct. On synthetic data, *GFCI* is only slightly slower than *RFCI* but more accurate than *FCI*, *RFCI* and *FCI+*.

1. Introduction

One of the major difficulties with inferring causal directed acyclic graphs (causal DAGs) or Markov equivalence classes of causal DAGs from observational data is accounting for the possibility that there are latent confounders among the measured variables. Many of the algorithms that have been developed for constructing DAGs or Markov equivalence classes of DAGs assume that there are no latent confounders, e.g. *PC* (Spirtes et al., 2001), *GES* (Chickering, 2002), and many others. Relatively few causal graphical model algorithms allow for the possibility of latent confounders. State of the art algorithms for searching for causal graphs with latent confounders suffer from a number of difficulties: some cannot be applied to large numbers of variables, require large sample sizes for accuracy, or lack asymptotic guarantees of correctness. We will describe an algorithm, *Greedy Fast Causal Inference* (*GFCI*) that is a combination of several different causal inference algorithms. *GFCI* has asymptotic guarantees of correctness and is more accurate on small sample sizes than current state of the art alternatives.

Section 2 describes current state of the art causal search algorithms for graphs with latent variables; Section 3 describes the assumptions and the output of the *GFCI* algorithm; Section 4 describes the algorithm; Section 5 describes a simulation study; and Section 6 is a discussion and summary.

2. State of the Art Algorithms

Among DAG search algorithms that do allow for the possibility of latent variables, there are some important limitations on their performance. The *Bayesian Structural EM* (Friedman, 1998), and *Information Bottleneck* (Elidan and Friedman, 2005) algorithms interleave a structure search with parameter estimation. However, they are only heuristic searches, and their output depends upon which initial starting DAG is given as input. Hoyer et al. (Hoyer et al., 2008) use *overcomplete ICA* to search for *canonical* DAGs which give the same predictions about manipulations as the correct latent variable DAG. However, *overcomplete ICA* is limited in both its accuracy if there are a large number of latent variables, and the number of measured variables that it can be applied to. The *BuildPureClusters* (Silva et al., 2006), *FOFC*, and *FTFC*(Kummerfeld et al., 2014) algorithms all make assumptions about the graphical structure of the true DAG. The *DM* algorithm (Murray-Watters and Glymour, 2015) also only works on a restricted class of true latent variable DAGs, and assumes that it is known which variables are input and output variables. The *Answer Set Programxfming causal discovery algorithm* (Hyttinen et al., 2013) uses a *maxSAT* solver to answer questions about causal DAGs that may have selection bias, latent variables, and feedback, but is currently feasible for only a dozen or so variables.

FCI (Spirtes et al., 1999), *RFCI* (Colombo et al., 2012), and *FCI*+ (Claasen et al., 2013) are constraint-based algorithms that all output a class of graphs that are Markov equivalent over the measured variables. They output a graphical object called a *Partial Ancestral Graph* (*PAG*) that represents the features common to all of the DAGs in the equivalence class. (These algorithms can also allow for the possibility of selection bias, but in this paper we will assume that there is no selection bias). In the large sample limit, given the *Causal Markov Assumption* (Spirtes et al., 2001), the *Causal Faithfulness Assumption* (Spirtes et al., 2001), i.i.d. sampling, and no feedback, they are guaranteed to output a PAG that contains the true latent variable DAG. One major limitation of *FCI* and related algorithms is that their small sample performance is often poor. The output tends to contain too few adjacencies, and incorrect orientations, especially far too many bi–directed edges (Colombo et al., 2012).

3. Assumptions and Output

We make the following assumptions and introduce the following terminology (Spirtes et al., 1999). We use standard graph terminology for directed graphs (e.g. parent, d-separation, etc.) We will consider several different kinds of directed graphs: directed acyclic graphs, patterns, and partial ancestral graphs, with different kinds of edges (explained below): $\rightarrow$, $\leftrightarrow$, $\circ\!\!\rightarrow$, $\circ\!\!-\!\!\circ$, —. In any of these kinds of graphs, a *directed path* from *A* to *B* is a sequence of vertices $\langle X_1, \ldots, X_n \rangle$ in which for $1 \leq i < n$, there is a directed edge from X_i to X_{i+1} ($X_i \rightarrow X_{i+1}$). *A* is a *parent* of *B* in *G* iff there is a directed edge $A \rightarrow B$; **Parents**(*A*,*G*) is the set of parents of *A*. If there is an acyclic directed path from *A* to *B* or *B* = *A* then *A* is an *ancestor* of *B*, and *B* is a *descendant* of *A*. If **Z** is a set of variables, *A* is a *descendant* of **Z** if and only if it is a descendant of a member of **Z**. If **X** is a set of vertices in a directed acyclic graph *G*, let **Descendants**(*G*,*X*) be the set of all descendants of members of **X** in *G*. (If the context makes clear what graph is being referred to, we will simply write **Descendants**(*X*).) Three vertices $\langle A, B, C \rangle$ are a *triple* in a pattern, PAG or DAG if both *A* and *B* are adjacent, and *B* and *C* are adjacent in the pattern, PAG or DAG; they are *unshielded* if *A* and *C* are not adjacent. An edge in a graph between *A* and *B* is *into B*, if there is an arrow head at the *B* end of the edge (i.e. $A \rightarrow B$, $A \leftrightarrow B$, or $A \circ\!\!\rightarrow B$). *B* is a *collider* on $\langle A, B, C \rangle$ if the edge between *A* and *B* and the edge between *B* and *C* are both into *B*.

A *causal directed graph G* over a set of vertices **V** for a given population is a directed graph in which the vertices are random variables, and there is a directed edge from *A* to *B* ($A \rightarrow B$) iff *A* is a direct cause of *B* relative to **V** in the population, i.e. there is some possible experimental manipulation of *A* that changes the probability distribution of *B* when all of the other variables in **V** are held fixed. We will assume that the causal directed graph is acyclic, i.e. that there is no feedback, that there is no selection bias, and samples are i.i.d.

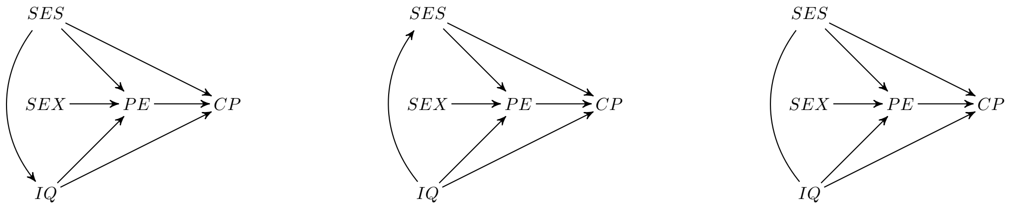
A probability distribution *P* satisfies the *local Markov condition* for a DAG *G* if and only if for each vertex *W* in *G*, *W* is independent of $\mathbf{V} \setminus (\mathbf{Descendants}(W) \cup \mathbf{Parents}(W))$ conditional on $\mathbf{Parents}(W)$.

We assume that causal graphs are related to probability distributions by the following assumptions (Spirtes et al., 2001). For each population, let **V** be a set of random variables such that every variable that is a direct cause (relative to **V** and the population) of two members of **V** is also in **V**, *G* be the causal directed acyclic graph over **V** for the population, and *P* be the probability distribution over the vertices in **V** in the popultion.

Local Causal Markov Assumption: *P* satisfies the local Markov condition for causal graph *G*, or equivalently, $\mathbf{A}$ is dependent on $\mathbf{B}$ conditional on $\mathbf{C}$ in *P* only if $\mathbf{A}$ is d-connected to $\mathbf{B}$ conditional on $\mathbf{C}$ in *P* in *G*.

Causal Faithfulness Assumption: The only conditional independencies that hold in the population are those entailed by the *Local Causal Markov Assumption*, or equivalently, $\mathbf{A}$ is d-connected to $\mathbf{B}$ conditional on $\mathbf{C}$ in *G* only if $\mathbf{A}$ is dependent on $\mathbf{B}$ conditional on $\mathbf{C}$ in *P*.

Two DAGs G_1 and G_2 are Markov equivalent if and only if they have the same set of vertices and the same set of d-separation relations. The set of DAGs that are Markov equivalent to *G* is designated **Equiv**(*G*). A *pattern* (or *PDAG* or *CPDAG*) represents features common to a Markov equivalence class of DAGs. A pattern *PAT represents* a DAG *G* (or **Equiv**(*G*)) iff (i) *A* and *B* are adjacent in *PAT* iff *A* and *B* are adjacent in *G*, and (ii) the edge between *A* and *B* is oriented as $A \rightarrow B$ in *PAT* iff $A \rightarrow B$ in every DAG in **Equiv**(*G*), and is oriented *A* — *B* iff $A \rightarrow B$ in some DAG in **Equiv**(*G*), and $A \leftarrow B$ in some other DAG in **Equiv**(*G*). Figure 1 shows an example of a DAG *G* (Figure 1a), a Markov equivalent DAG (Figure 1b), and a pattern that represents *G* and **Equiv**(*G*) (Figure 1c).



(a) DAG

(b) Markov Equivalent DAG

(c) Pattern

Figure 1: Markov Equivalence Class and Pattern

When a DAG has a set of vertices **V** that can be partitioned into the observed variables **O** and the latent variables **L**, we designate the graph as $G(\mathbf{O}, \mathbf{L})$. Two DAGs $G_1(\mathbf{O}, \mathbf{L_1})$ and $G_2(\mathbf{O}, \mathbf{L_2})$ whose vertices both contain the same observed subset **O** are Markov equivalent over **O** if and only if they have the same set of d-separation relations among vertices in **O**. The Markov equivalence class over **O** of $G(\mathbf{O}, \mathbf{L})$ is **O-equiv**$(G(\mathbf{O}, \mathbf{L}))$.

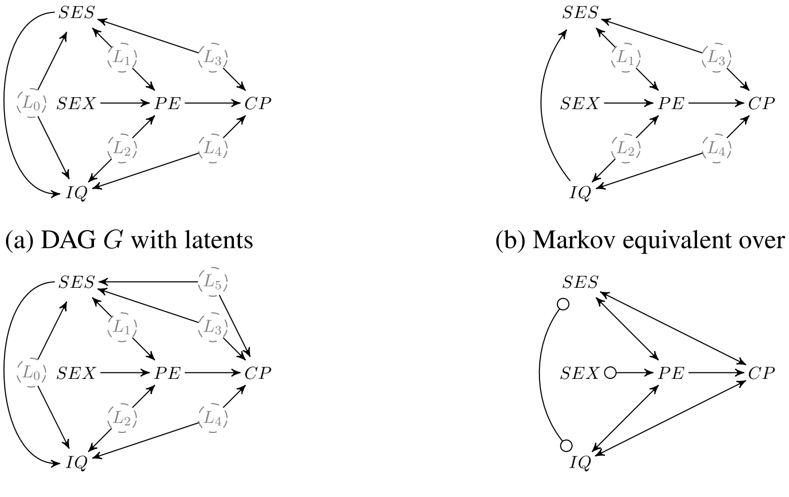
A partial ancestral graph *PAG* (without selection bias) represents a DAG $G(\mathbf{O}, \mathbf{L})$ or an equivalence class **O-equiv**$(G(\mathbf{O}, \mathbf{L}))$if and only if:

- The set of variables in *PAG* is **O**.
- If there is any edge between *A* and *B* in *PAG*, it is one of the following kinds: $A \rightarrow B$, $A \circ\!\!\rightarrow B$, $A \leftrightarrow B$, or $A \circ\!\!-\!\!\circ B$.

- *A* and *B* are adjacent in *PAG* if and only if for every subset $\mathbf{Z}$ of $\mathbf{O} \setminus \{A, B\}$ *A* is d-connected to *B* conditional on $\mathbf{Z}$ in every DAG in **O-Equiv**$(G(\mathbf{O}, \mathbf{L}))$.
- An edge between *A* and *B* in *PAG* is oriented as $A \rightarrow B$ only if *A* is an ancestor of *B* in every DAG in **O-Equiv**$(G(\mathbf{O}, \mathbf{L}))$.
- An edge between *A* and *B* in *PAG* is oriented as $A \leftrightarrow B$ only if *B* is not an ancestor of *A* and *A* is not an ancestor of *B* in any DAG in **O-Equiv**$(G(\mathbf{O}, \mathbf{L}))$.
- An edge between A and B in *PAG* is oriented as $A \circ\!\!\rightarrow B$ only if B is not an ancestor of A in any DAG in **O-Equiv**$((G(\mathbf{O}, \mathbf{L})))$, and A is an ancestor of B in some but not all DAGs in **O-Equiv**$(G(\mathbf{O}, \mathbf{L}))$.
- An edge between A and B in *PAG* is oriented as $A \circ\!\!-\!\!\circ B$ only if B is an ancesttor of A in some but not all DAGs in **O-Equiv**$((G(\mathbf{O}, \mathbf{L})))$, and B is an ancestor of A in some but not all DAGs in **O-Equiv**$(G(\mathbf{O}, \mathbf{L}))$.

$\mathbf{A}$ is *d-separated* from $\mathbf{B}$ conditional on $\mathbf{C}$ in a PAG if $\mathbf{A}$ is d-separated from $\mathbf{B}$ conditional on $\mathbf{C}$ in every DAG represented by the PAG.

Figure 2 shows a directed acyclic graph $G(\mathbf{O}, \mathbf{L})$, where $\mathbf{O} = \{SEX, SES, IQ, PE, CP\}$ (Figure 2a); some members of the Markov equivalence class over **O** (Figures 2b and 2c); and a PAG that represents **O-equiv**(*G*(**O**,**L**)) (Figure 2d).



(a) DAG G with latents

(b) Markov equivalent over **O**

(c) Markov equivalent over **O**

(d) PAG over **O**

Figure 2: Markov Equivalence Class and PAG

4. THE *GFCI* (*GRFCI*, *GRFCI+*) ALGORITHMS

4.1 *FCI*, *RFCI*, *FCI+*

FCI, *FCI+* and *RFCI* all take as input a sample from an i.i.d. distribution, optional background knowledge (e.g. time order), and output a PAG (in the case of *RFCI* a slightly modified object called a *RFCI–PAG*). For the purposes of this paper, they are all similar enough that we will only briefly describe *FCI*. Under the Causal Markov and Causal Faithfulness assumptions, the no feedback assumption, and the i.i.d. sampling assumption, in the large sample limit, the *PAG* that *FCI* ouputs is guaranteed to represent the **O**-equivalence class of the true causal DAG, where **O** is the set of observed variables. (The RFCI–PAG is in principle somewhat less informative than the PAG output by *FCI* in certain unusual cases, but in practice *RFCI* is much faster than *FCI*, and the differences in output are minimal (Colombo et al., 2012)).

Theorem 1 *Given the Causal Markov Assumption, the Causal Faithfulness Assumption, a causal system represent by a DAG, and i.i.d. sampling, in the large sample limit, the FCI algorithm outputs a PAG that represents the true causal DAG. ((Spirtes et al., 1999))*

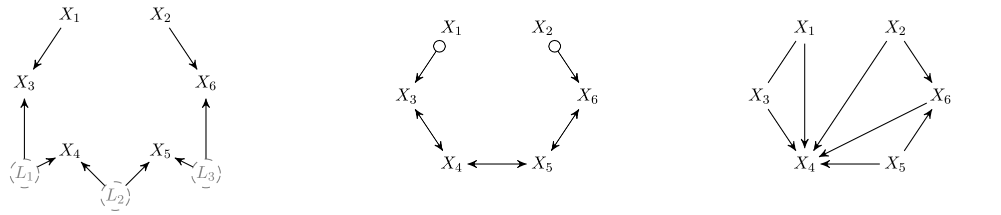
4.2 *GES*, *FGS*

In contrast to *FCI*, the Greedy Equivalence Search (*GES*) is a score-based algorithm that outputs a pattern. Given a locally consistent score such as the Bayesian Information Criterion (BIC), *GES* outputs a pattern that represents the true causal DAG. ((Chickering, 2002))

Theorem 2 *Given the Causal Markov Assumption, the Causal Faithfulness Assumption, a causal system represented by a DAG, no latent confounders, a locally consistent score, and i.i.d. sampling, in the large sample limit, the* GES *algorithm outputs a pattern that represents the true causal DAG.*

In practice, we actually use the *Fast Greedy Search* (*FGS*) algorithm (Ramsey, 2015), which is a modification of *GES* that uses the same scores and search algorithm and has the same output as *GES*, but has different data structures and greatly speeds up *GES* by caching information about scores that are calculated in the course of the search (Ramsey, 2015). Since they have the same output, henceforth we will refer to the output of *GES* and *FGS* interchangeably.

If there is a latent confounder, then *GES* may include extra edges that are not in the true causal PAG; in addition the orientations are sometimes incorrect. We use *FCI* as a post-processor for *GES* in order to remove the extra adjacencies, and correct the orientations in the output of *GES*. For example, if the true graph is in Figure 3a, and the true PAG is in Figure 3b, the output of *GES* (and *FGS*) is in Figure 3c. Figure 3c contains three adjacencies not in the true causal PAG: $X_1 \rightarrow X_4, X_2 \rightarrow X_4$, and $X_6 \rightarrow X_4$. The reason for the extra adjacencies in the output of *FGS* as compared to the true PAG is that the output of *FGS* is Markov to the true distribution, and there is no DAG without latent variables and the same adjacencies as the true PAG that is Markov to a marginal distribution faithful to the true DAG; since the output of *FGS* cannot add extra variables or represent hidden counfounders, in order to make the output Markov to the true distribution, it adds extra adjacencies. And since the output of *FGS* cannot represent latent confounders and possible



(a) True DAG with latents (b) True PAG over **O** (c) Output of *GES* or *FGS*

Figure 3: Output of *FGS* when given a data from a system with latents

latent confounders (represented by $\leftrightarrow$ in the true PAG and $\circ\!\!\rightarrow$ respectively), *FGS* also misorients some of the edges. However, the output of *FGS* does correctly orient unshielded colliders and non-colliders.

4.3 Greedy Fast Causal Inference (*GFCI*)

Due to space limitations, we will not describe the entire algorithm – instead we will describe the changes that *GFCI* makes to the version of *FCI* described in (Spirtes et al., 1999). *FCI* has the following stages. Step A initializes a graph *Q* to a complete undirected graph (that is subsequently modified by following steps into the ultimate output). Steps B and D search for edges to remove from *Q* by finding pairs of adjacent variables in *Q* that are independent conditional on some subset of the other variables, with step D requiring a partial orientation of edges in step C. Step E removes any orientations that were imposed during step C. Step F orients unshielded colliders $A \circ\!\!\rightarrow B \leftarrow\!\!\circ C$ if the conditioning set that led to the removal of the edge between A and C does not contain B. Step G uses more orientation rules to orient more edges until no more orientations are possible. Steps in *GFCI* that modify corresponding steps in FCI are labeled with a prime, e.g. A'. In *GFCI*, **Sepset**(*A,B*) records the conditioning set that led to the removal of an adjacency between *A* and *B* if there is one (as is always the case in *FCI*); or if there was no such conditioning set because the edge was removed in the *FGS* stage of *GFCI*, and **Sepset**(*A,B*) is needed for an orientation rule, **Sepset**(*A,B*) is set to the results of a search for a conditioning set that makes *A* and *B* independent. **Possible-D-Sep**(*X,Z*) is defined in (Spirtes et al., 1999).

Algorithm 1: Greedy Fast Causal Inference (*GFCI*)

```
Data: $Data$
Result: $PAG$
Run FGS on data and obtain output pattern PAT.
A'. Form an undirected graph Q using the adjacencies in PAT.
B: For all adjacencies in Q, apply step B of FCI to remove an adjacency between $X$ and $Y$ if
there is an independence between $X$ and $Y$ conditioning on some set M that is a subset of
adjacencies of $X$ or a subset of adjacencies of $Y$; record M in Sepset(X,Y) and Sepset(Y,X).
C': If $\langle X, Y, Z\rangle$ is an unshielded triple in Q, then orient it as $X \circ\!\!\rightarrow Y \leftarrow\!\!\circ Z$ if it is an
unshielded collider in PAT, or it is shielded in PAT and Sepset(X,Z) does not contain $Y$.
D: For all adjacencies in Q, apply step D of FCI to remove an adjacency between X and Z if
there is an independence between $X$ and $Z$ conditioning on a subset M of
Possible-D-Sep(X,Z) or Possible-D-Sep(Z,X); record M in Sepset(X,Z) and Sepset(Z,X) .
E: Apply step E of FCI to unorient all of the edges in Q that remain.
F':If $\langle X, Y, Z\rangle$ is an unshielded triple in Q, then orient it as $X \circ\!\!\rightarrow Y \leftarrow\!\!\circ Z$ if it is an
unshielded collider in PAT, or it is shielded in PAT and Sepset(X,Z) does not contain $Y$.
G: Apply further orientations from step G of FCI.
```

4.4 Sketch of Proof of Correctness

Lemma 3 *In the large sample limit of an i.i.d sample from a distribution P, if P is Markov and Faithful to DAG G with observed variables* **O***, and G is represented by partial ancestral graph PAG* *over **O**, then every pair of vertices X and Y adjacent in PAG is also adjacent in the output pattern PAT of GES.*

Proof Suppose that an edge $A \rightarrow B$ is in *PAG*. By assumption, the distribution is faithful to *PAG*, and so in every DAG represented by *PAG*, *A* is d-connected to *B* conditional on every subset of the observed variables, and hence by the Causal Faithfulness Assumption, *A* is dependent on *B* conditional on every subset of the observed variables. Lemma 7 in Chickering(2002) shows that a Bayesian scoring criterion increases the score of an arbitrary graph *G* by adding $A \rightarrow B$ as long as *A* and *B* are dependent conditional on the parents of *B* in *G*. Hence *GES* would add the edge $A \rightarrow B$ in the course of its forward search, and not remove it in the course of its backward search. Hence $A \rightarrow B$ is in *PAT*. ■

Discriminated colliders are shielded colliders that are oriented by discriminating paths; see (Spirtes et al., 1999), which also contains proofs of lemmas 4 and 5.

Lemma 4 *If A and B are not adjacent in a DAG, pattern or a PAG, than A and C are d-separated by some subset of variables in the DAG, pattern or PAG.*

Lemma 5 *If $\langle A, B, C\rangle$ is an unshielded or discriminated collider (non-collider) in a DAG, pattern or a PAG, then every subset of variables in the DAG, pattern or PAG that d-separates A and C does not (does) contain B.*

Lemma 6 *In the large sample limit of an i.i.d sample from a distribution P, if P is Markov and Faithful to PAG with observed variables **O**, PAT is the output of GES (or FGS), then every unshielded collider in PAT that is a triple in PAG is an unshielded collider in PAG, and every unshielded non-collider in PAT that is a triple in PAG is an unshielded non-collider in PAG.*

Proof Chickering(2002) proved that given i.i.d. samples from a distribution *P*, the output of *GES* is Markov to *P* in the large sample limit. If an unshielded triple in $\langle A, B, C\rangle$ in *PAT* is a triple in *PAG*, then it is an unshielded triple in *PAG*, because by Lemma 3, there is a non-adjacency between *A* and *C* in *PAT* only if there is a non-adjacency between *A* and *C* in *PAG*. By Lemmas 4 and 5 every unshielded triple $\langle A, B, C\rangle$ in *PAT* is an unshielded collider iff some subset $\mathbf{S}$ of vertices not containing *B* d-separates *A* and *C*. *PAT* is Markov to *P*, so *A* is independent of *C* conditional on $\mathbf{S}$ in *P*, and if $\langle A, B, C\rangle$ is a triple in *PAG*, it is an unshielded triple in *PAG*. Since *P* is Faithful to *PAG*, $\mathbf{S}$ d-separates *A* and *C* in *PAG*, and hence *B* is a collider in *PAG*. By Lemmas 4 and 5 $\langle A, B, C\rangle$ is an unshielded non-collider in *PAG* iff some subset $\mathbf{S}$ of vertices containing *B* d-separates *A* and *C*. *PAT* is Markov to *P*, so *A* is independent of *C* conditional on $\mathbf{S}$ in *P*, and if $\langle A, B, C\rangle$ is a triple in *PAG*, it is an unshielded triple in *PAG*. Since *P* is Faithful to *PAG*, $\mathbf{S}$ d-separates *A* and *C* in *PAG*, and hence *B* is a non-collider in *PAG*. ■

Theorem 7 *Given the Causal Markov Assumption, the Causal Faithfulness Assumption, a causal system represent by a DAG, and i.i.d. sampling, in the large sample limit, the GFCI algorithm outputs a PAG that represents the true causal DAG.*

Proof First consider the adjacencies. The pattern output by the *FGS* part of *GFCI* does not remove too many edges by Lemma 3. The *FCI* part of *GFCI* does not remove too many edges by the Causal Faithfulness Assumption. The *FCI* part of *GFCI* does not remove too few edges, because *GFCI* considers all subsets of adjacencies to A, adjacencies to B, **Possible-D-Sep**(*A*,*B*), and **Possible-D-Sep**(*B*,*A*), which is sufficient to remove all edges that should be removed.

Next consider the orientations. The orientations are completely determined by the adjacencies, the unshielded colliders and non-colliders and the discriminated colliders and non-colliders. Lemmas 4 and 5 shows that the orientation of unshielded colliders and non-colliders from the output of the *FGS* portion of *GFCI* is correct, and all of the other unshielded or discriminated colliders or non-colliders are oriented by the same rules as *FCI*, which has been proved correct. ■

The complexity of the *FCI* part of *GFCI* is exponential in the number of variables. In practice, with 1000 variables and up to 2000 edges randomly generated, and as many as 200 latent variables, the median time was 6.39 seconds with a maximum of 29 seconds.

5. Simulations

We conducted simulations to test the performance of *GFCI* compared with that of *FCI*, *RFCI* and *FCI+*. In order to speed up *GFCI* we only tested whether to remove edges from the pattern output by *FGS* if they were part of a clique with at least 3 vertices. We conjecture that this is always correct, but even if it is not, it affects the performance only by at most a small amount.

The variables were given an order, and then a random pair of variables was chosen and assigned an edge from the earlier to the later variable, if the edge did not already exist. Latent variables were selected at random from a list of all nodes that are common causes of pairs of observed variables. Every node X was assigned a random variable of the form $\sum_i a_i P_{Xi} + \epsilon_X$. $\{P_{Xi}\}_i$ is the set of parents of X; a_i is a coefficient picked uniformly from $\pm[.2, 1.5]$ and ϵ_X is a Gaussian random variable with mean 0 and variance picked uniformly from $[1, 3]$. Data samples of different sizes were generated by obtaining values for the exogenous variables and passing the values down through the linear model. The samples were then used to construct a covariance matrix which was used as input to all of the algorithms.

GFCI has two tuning parameters parameters. `depth`, which limits the size of the parent set of a variable was set to 1000. `penaltyDiscount` is a constant that multiplies the penalty term in the BIC score used by *GES* and was set to 4 based on experience with *FGS* in other contexts. The turning parameters for *FCI*, *RFCI* and *FCI+* were all set to the same values. All of the versions of *FCI*, including *GFCI*, take a tuning parameter *alpha*, that is the alpha level of a Fisher's z statistical test of conditional independence employed by the algorithms. We ran each algorithm at three different alpha levels: 0.01, 0.05 and 0.1. `depth` determines the maximum size of possible conditioning sets used in the adjacency search, and `maxPathLength` determines the maximum length of discriminating paths in the final steps of the orientation. Both were initially set to 3, as larger values tend to prove intractable for dense graphs. `completeRulesetUsed` was set to false to eliminate application of orientation rules that only apply when there is selection bias. We also ran a theoretically more accurate version of *RFCI* which places no restriction on the size of possible d-separating sets or path length, called *uRFCI*. The versions of all the algorithms used in testing were slight modifications of those available through the Tetrad software package (available at http://www.phil.cmu.edu/tetrad/current.html).

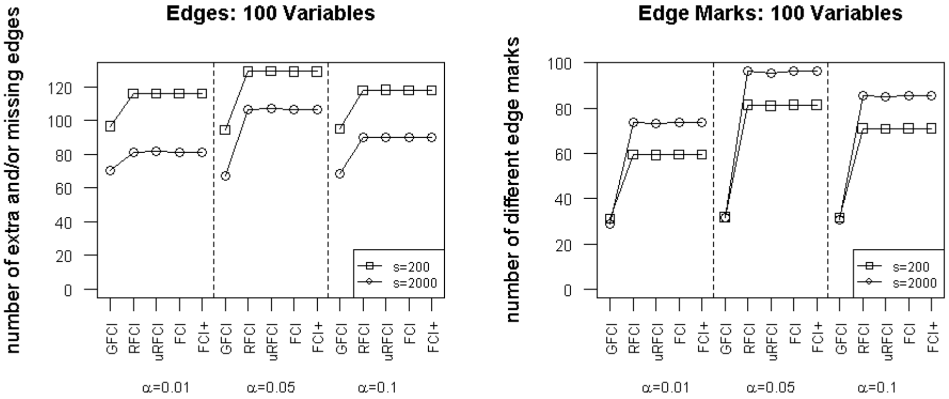


Figure 4: Average estimation errors of the various algorithms in the 100 variable cases, across different alpha levels for the independence test.

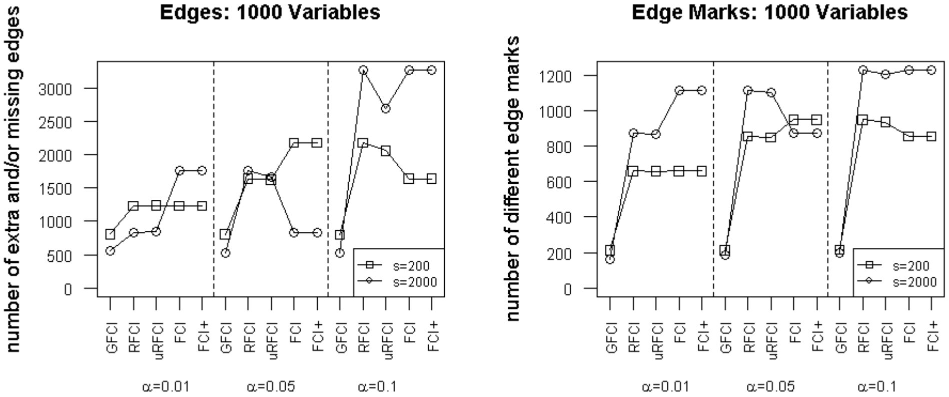


Figure 5: Average estimation errors of the various algorithms in the 1000 variable cases, across different alpha levels for the independence test.

Graphs were generated with 100 and 1000 nodes. Edge ratios were set to 1 and 2. Latent ratios were also set to .05 and .20. Once a DAG and model were fixed, data was generated with sample size of either 200 or 2000. Each run of our simulations consisted of running all of the algorithms (*GFCI*, *RFCI*, *uRFCI*, *FCI*, *FCI+*) for one of the 48 possible configurations above. For each configuration we tried to collect 100 trials. For the 1000 variable case, we first ran *GFCI* and the bounded version of *RFCI*. To avoid running into the difficulty of having the DAG to PAG conversion take too long, we split the simulations into batches consisting of the 24 configurations of 1000 node parameters each with a maximum of 20 trials. If the DAG to PAG conversion failed to finish within 12 hours, we would terminate the batch and start the next one. A total of 14 batches over the course of a week were necessary. 2 batches finished completely; 11 batches were interrupted during the DAG to PAG conversion; 1 batch was interrupted by a memory overflow error while running *RFCI*. Although

GFCI finished running in this data set, we have excluded that trial from our results. *FCI*, *FCI+* and *uRFCI* were all run later, using the same graphs, model and data sets as *GFCI* and bounded *RFCI*.

To measure the accuracy of the algorithms, we recorded in Table 3 the precision and recall for adjacencies, each kind of edge endpoint (arrowhead or arrow tail), and each kind of edge ($\circ\!\!-\!\!\circ$, $\circ\!\!\rightarrow$, $\rightarrow$, $\leftrightarrow$) We present the results for an alpha level of 0.01 - the other alpha levels gave similar, but generally slightly worse results. With respect to all of the parts of Table 3 except for the recall of double-headed arrows, and except for a few combination of the graph generation parameters, *GFCI* has better precision and recall than *FCI*, *FCI+* and *uRFCI*. In the few cases where *GFCI* is not at least tied for best precision or recall (e.g. tail precision for 1000 variables, 200 latents, 1000 edges, 2000 samples), *GFCI* has almost the same precision as the best algorithm (.41 versus .42) and has a higher F1 score (precision * recall/ precision + recall). For bi–directed edges the other algorithms tend to have better recall of bi–directed edges. This comes at the expense, however, of much worse precision. This suggests that the non–*GFCI* algorithms tend to add too many bi–directed edges. For reasons of space, we present only the results for alpha = 0.01 and 1000 variables. We also present results for the same measures as those in (Colombo et al., 2012) at three different alpha levels; the total number of adjacency mistakes, and the total number of edge endpoint mistakes. The results are shown in FIgures 4 and 5. Again, *GFCI* was substantially better than the other algorithms in each case.

The running times for *GFCI*, *RFCI*, *uRFCI*, and *FCI+* for the 1000 variable case had means 8.862, 7.188, 375.000, and 13.190 respectively, and standard deviations 5.489, 12.111, 1314.978, and 4.86 respectively.

6. Discussion

The increased accuracy of *GFCI*, a hybrid of a constraint-based and a score-based algorithm, over pure constraint-based algorithms is consistent with previous research. The first part of *FCI* is very similar to the *PC* algorithm, and studies have indicated that *GES* has superior performance to the *PC* algorithm. (Nandy et al., 2016). For example, *FCI* has a bias towards orienting triples as unshielded colliders when there is conflicting information about whether a given triple of variables should be considered an unshielded collider or not. If there are 6 variables (X_1 through X_6) adjacent to Y, and no edges between the X's, then there are 15 pairs of unshielded triples containing Y. If just three of those triples, such as $\langle X_1, Y, X_2\rangle$, $\langle X_3, Y, X_4\rangle$, and $\langle X_5, Y, X_6\rangle$ are oriented as colliders by *FCI*, that entails that the other 12 are also oriented as unshielded colliders (because all of the X_i – Y edges will have arrowheads at Y), even if statistically, the evidence is against 12 of those triples being colliders. In contrast, *GES* would consider all of the possible orientations of the 6 edges, and find which one scores best.

One of the disadvantages of $GFCI$ is that the *FCI*-family of algorithms requires only a consistent test of conditional independence in order to be asymptotically correct. However *GFCI* requires a locally consistent score in its *GES* part, which is known only for a few distributions such as multivariate normal or multinomial distributions.

7. Acknowledgements

This research was supported by grant U54HG008540 awarded by the National Institutes of Health. We thank anonymous referees for helpful suggestions.

DAG								Adj.		Arrow		Tail		○—○		○→		→		↔	
Nodes	Lat.	Edges	Samples	α	Alg.	Runs	Time	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
1000	50	1000	200	0.01	GFCI	116	4.776s	**0.98**	**0.81**	**0.96**	**0.70**	**0.78**	**0.77**	**0.83**	**0.82**	**0.92**	**0.71**	**0.78**	**0.77**	**0.76**	**0.02**
					RFCI	116	0.274s	0.77	0.71	0.37	0.56	0.50	0.26	0.61	0.30	0.42	0.22	0.50	0.26	0.02	**0.02**
					uRFCI	116	1.858s	0.77	0.71	0.37	0.56	0.50	0.26	0.61	0.31	0.41	0.22	0.50	0.26	0.02	**0.02**
					FCI	116	0.02s	0.77	0.71	0.37	0.56	0.50	0.26	0.61	0.30	0.42	0.22	0.50	0.26	0.02	**0.02**
					FCI+	116	3.007s	0.77	0.71	0.37	0.56	0.50	0.26	0.61	0.30	0.41	0.22	0.50	0.26	0.02	**0.02**
1000	50	1000	2000	0.01	GFCI	116	5.739s	**1.00**	**0.94**	**0.98**	**0.88**	**0.86**	**0.96**	**0.98**	**0.96**	**0.99**	**0.90**	**0.86**	**0.96**	**0.74**	**0.33**
					RFCI	116	0.387s	0.74	0.93	0.40	**0.88**	0.60	0.69	0.94	0.35	0.65	0.38	0.60	0.69	0.04	0.07
					uRFCI	116	2.216s	0.74	0.93	0.40	**0.88**	0.60	0.69	0.94	0.36	0.65	0.38	0.60	0.69	0.04	0.07
					FCI	116	0.053s	0.74	0.93	0.40	**0.88**	0.60	0.69	0.94	0.36	0.65	0.38	0.60	0.69	0.04	0.07
					FCI+	116	4.112s	0.74	0.93	0.40	**0.88**	0.60	0.69	0.94	0.35	0.65	0.38	0.60	0.69	0.04	0.07
1000	200	1000	200	0.01	GFCI	116	2.795s	**0.98**	**0.48**	**0.95**	**0.29**	**0.50**	**0.53**	**0.68**	**0.72**	**0.83**	**0.41**	**0.50**	**0.54**	**0.85**	0.01
					RFCI	116	0.171s	0.70	0.45	0.35	**0.29**	0.39	0.17	0.56	0.27	0.38	0.15	0.39	0.17	0.07	0.05
					uRFCI	116	1.494s	0.70	0.45	0.35	**0.29**	0.39	0.17	0.55	0.27	0.38	0.15	0.39	0.17	0.07	0.05
					FCI	116	0.016s	0.70	0.45	0.35	**0.29**	0.39	0.17	0.56	0.27	0.38	0.15	0.39	0.17	0.07	0.05
					FCI+	116	2.109s	0.70	0.45	0.35	**0.29**	0.39	0.17	0.56	0.27	0.38	0.15	0.39	0.17	0.07	0.05
1000	200	1000	2000	0.01	GFCI	116	4.981s	**1.00**	0.69	**0.95**	0.52	0.41	**0.83**	**0.90**	**0.85**	**0.94**	**0.62**	0.41	**0.83**	**0.68**	0.14
					RFCI	116	0.319s	0.71	**0.71**	0.40	**0.59**	**0.42**	0.52	0.89	0.25	0.59	0.23	**0.42**	0.52	0.14	**0.17**
					uRFCI	116	2.051s	0.71	0.70	0.40	**0.59**	**0.42**	0.52	0.89	0.25	0.59	0.23	**0.42**	0.52	0.14	**0.17**
					FCI	116	0.085s	0.71	**0.71**	0.40	**0.59**	**0.42**	0.52	0.89	0.25	0.59	0.23	**0.42**	0.52	0.14	**0.17**
					FCI+	116	3.268s	0.71	**0.71**	0.40	**0.59**	**0.42**	0.52	0.89	0.25	0.59	0.23	**0.42**	0.52	0.14	**0.17**
1000	50	2000	200	0.01	GFCI	111	11.328s	**0.98**	**0.78**	**0.96**	**0.70**	**0.84**	**0.81**	**0.80**	**0.76**	**0.89**	**0.69**	**0.84**	**0.82**	**0.50**	0.01
					RFCI	111	0.678s	0.95	0.50	0.55	0.40	0.64	0.18	0.35	0.20	0.41	0.11	0.64	0.18	0.02	**0.04**
					uRFCI	111	2.439s	0.95	0.49	0.55	0.40	0.64	0.18	0.35	0.20	0.41	0.11	0.64	0.18	0.02	**0.04**
					FCI	111	0.036s	0.95	0.50	0.54	0.40	0.64	0.18	0.35	0.20	0.41	0.11	0.64	0.18	0.02	**0.04**
					FCI+	111	4.136s	0.95	0.50	0.55	0.40	0.64	0.18	0.35	0.20	0.41	0.11	0.64	0.19	0.02	**0.04**
1000	50	2000	2000	0.01	GFCI	111	15.273s	**1.00**	**0.88**	**0.98**	**0.82**	**0.92**	**0.93**	**0.97**	**0.93**	**0.98**	**0.86**	**0.92**	**0.93**	**0.69**	0.16
					RFCI	111	3.14s	0.96	0.79	0.59	0.73	0.62	0.47	0.87	0.16	0.74	0.16	0.62	0.47	0.04	**0.18**
					uRFCI	111	5.877s	0.97	0.78	0.60	0.72	0.62	0.47	0.86	0.16	0.74	0.16	0.62	0.48	0.04	0.17
					FCI	111	0.174s	0.96	0.79	0.59	0.73	0.62	0.47	0.87	0.16	0.74	0.16	0.62	0.47	0.04	**0.18**
					FCI+	111	8.856s	0.96	0.79	0.59	0.73	0.62	0.47	0.87	0.16	0.74	0.16	0.62	0.47	0.04	**0.18**
1000	200	2000	200	0.01	GFCI	104	8.217s	**0.97**	**0.42**	**0.94**	**0.28**	**0.70**	**0.62**	**0.66**	**0.68**	**0.81**	**0.41**	**0.70**	**0.63**	**0.72**	0.01
					RFCI	104	0.466s	0.94	0.27	0.55	0.18	0.61	0.13	0.31	0.19	0.38	0.07	0.61	0.13	0.07	**0.03**
					uRFCI	104	1.907s	0.94	0.27	0.55	0.17	0.61	0.13	0.31	0.19	0.38	0.07	0.61	0.13	0.07	**0.03**
					FCI	104	0.03s	0.94	0.27	0.55	0.18	0.61	0.13	0.31	0.19	0.38	0.07	0.61	0.13	0.07	**0.03**
					FCI+	104	2.851s	0.94	0.27	0.55	0.18	0.61	0.13	0.31	0.19	0.38	0.07	0.61	0.13	0.07	**0.03**
1000	200	2000	2000	0.01	GFCI	104	19.375s	**0.99**	**0.53**	**0.96**	**0.40**	**0.71**	**0.78**	**0.89**	**0.79**	**0.94**	**0.54**	**0.71**	**0.78**	**0.72**	0.08
					RFCI	104	2.596s	0.96	0.49	0.61	0.39	0.58	0.34	0.78	0.12	0.69	0.09	0.58	0.35	0.16	**0.14**
					uRFCI	104	5.979s	0.96	0.48	0.61	0.38	0.58	0.35	0.77	0.12	0.70	0.09	0.58	0.35	0.15	0.13
					FCI	104	0.269s	0.96	0.49	0.61	0.39	0.58	0.34	0.77	0.12	0.69	0.09	0.58	0.34	0.16	**0.14**
					FCI+	104	7.066s	0.96	0.49	0.61	0.39	0.58	0.35	0.78	0.12	0.69	0.09	0.58	0.35	0.16	**0.14**

Table 1: Lists the average running time and the average precision and recall for the different accuracy measurements for every algorithm in every parametrization with 1000 variables and $\alpha = 0.01$. The bold values represent the best value in that column for that parametrization.

References

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

T. Claasen, J. M. Mooij, and T. Heskes. Learning sparse causal models is not np-hard. In P. S. Ann Nicholson, editor, *Proceedings of the Twenty-Ninth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, volume 29, pages 172–181. AUAI Press, 2013.

D. Colombo, M. H. Maathuis, M. Kalisch, and T. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *Annals of Statistics*, 40(1):294–321, 2012.

G. Elidan and N. Friedman. Learning hidden variable networks: The information bottleneck approach. *J Mach Learn Res*, 6:81–127, 2005.

N. Friedman. The bayesian structural em algorithm. In G. Cooper and S. Moral, editors, *UAI'98 Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 129–137, Madison, WI, 1998. Morgan Kaurmann.

P. O. Hoyer, S. Shimizu, A. Kerminen, and M. Palviainen. Estimation of causal effects using linear non-gaussian causal models with hidden variables. *International Journal of Approximate Reasoning*, 49(2):362–378, 2008.

A. Hyttinen, P. Hoyer, F. Eberhardt, and M. Jarvisalo. Discovering cyclic causal models with latent variables: A general sat-based procedure. In P. S. Ann Nicholson, editor, *Proceedings of the Twenty-Ninth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, volume 29, pages 301–310, Corvallis, Oregon, 2013. AUAI Press.

E. Kummerfeld, J. Ramsey, R. Yang, P. Spirtes, and R. Scheines. Causal clustering for 2-factor measurement models. In T. Calders, F. Esposito, E. Hllermeier, and R. Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 34–49. Springer, 2014.

A. Murray-Watters and C. Glymour. What is going on inside the arrows? discovering the hidden springs in causal models. *Philosophy of Science*, 82(4):pp. 556–586, 2015.

P. Nandy, A. Hauser, and M. Maathuis. High-dimensional consistency in score-based and hybrid structure learning. 2016. URL `http://arxiv.org/pdf/1507.02608v3.pdf`.

J. D. Ramsey. Scaling up greedy equivalence search for continuous variables. *CoRR*, abs/1507.07749, 2015. URL `http://arxiv.org/abs/1507.07749`.

R. Silva, R. Scheines, C. Glymour, and P. Spirtes. Learning the structure of linear latent variable models. *J Mach Learn Res*, 7:191–246, 2006.

P. Spirtes, T. RIchardson, and C. Meek. Causal discovery in the presence of latent variables and selection bias. In G. Cooper and C. Glymour, editors, *Computation, Causality, and Discovery*, pages 211–252. AAAI Press, 1999.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search, Second Edition (Adaptive Computation and Machine Learning).* The MIT Press, 2001.