An Instance-Specific Algorithm for Learning the Structure of Causal Bayesian Networks Containing Latent Variables

Fattaneh Jabbari* Gregory F. Cooper*,†

Abstract

Almost all of the existing algorithms for learning a causal Bayesian network structure (CBN) from observational data recover a structure that models *the causal relationships that are shared by the instances in a population.* Although learning such population-wide CBNs accurately is useful, it is important to learn CBNs that are specific to each instance in domains in which different instances may have varying causal structures, such as in human biology. For example, a breast cancer tumor in a patient (instance) is often a composite of causal mechanisms, where each of these individual causal mechanisms may appear relatively frequently in breast-cancer tumors of other patients, but the particular combination of mechanisms is unique to the current tumor. Therefore, it is critical to discover the specific set of causal mechanisms that are operating in each patient to understand and treat that particular patient effectively.

We previously introduced an instance-specific CBN structure learning method that builds a causal model for a given instance T from the features we know about T and from a training set of data on many other instances [12]. However, that method assumes that there are no latent (hidden) confounders, that is, there are no latent variables that cause two or more of the measured variables. Unfortunately, this assumption rarely holds in practice. In the current paper, we introduce a novel instance-specific causal structure learning algorithm that uses partial ancestral graphs (PAGs) to model latent confounders. Simulations support that the proposed instance-specific method improves structure-discovery performance compared to an existing PAG-learning method called GFCI, which is not instance-specific. We also report results that provide support for instance-specific causal relationships existing in real-world datasets.

1 Introduction

Causal Bayesian networks (CBNs) have been used extensively for discovering causal knowledge from observational data [16, 20]. During the past few decades, several algorithms have been developed to infer CBN structures from observational data. Almost all of the existing CBN structure learning approaches are intended to discover a CBN structure that represents the causal relationships that are shared by the instances in a population; we call such a model a *population-wide CBN model*. However, in many domains, like human biology, different instances of the population may have varying causal structures.

For example, consider a complex disease like cancer in which each individual tumor in a patient (instance) is derived by many distinct underlying causal mechanisms. Each of these causal mechanisms may appear relatively frequently in other patients, but the *joint set of mechanisms* in the current patient's tumor is unique to that patient. Understanding and identifying the particular set of causal mechanisms that are driving a cancerous tumor in the current patient will likely lead to more effective therapies for the current patient. A population-wide CBN would at best recover the more common causal mechanisms operating in a population of tumors, and consequently, would fail to capture the particular causal mechanisms that are deriving a tumor in each patient. Instead of learning a single CBN model for all the instances in a population, our goal is to construct a specialized CBN structure for a given instance (e.g., a patient) by leveraging the features (i.e., the variable values) of the given instance and a training set of data on many other instances; we call such a model an *instance-specific CBN model*.

We previously introduced a fully Bayesian instance-specific structure learning method, called IGES [12], that searches the space of CBNs to build a model that is specific to an instance T by guiding the search from the features we know about T and from a training set of data on many other instances. The IGES method assumes that there are no latent confounders (i.e., it makes the causal sufficiency assumption). However, relying on the causal sufficiency assumption could be a major drawback since this assumption is unrealistic in many practical applications. In the current paper, we introduce a novel instance-specific causal structure learning algorithm that uses partial ancestral graphs (PAGs) to learn CBNs that can model for the possi-

*Intelligent Systems Program, University of Pittsburgh.

†Department of Biomedical Informatics, University of Pittsburgh.

bility of latent confounders. We hypothesize that such an instance-specific learning approach will model the causal relationships for T better than does a population-wide one. We evaluate this hypothesis using simulated and real data. The remainder of this paper first provides related work in Section 2 and the relevant background in Section 3. Section 4 describes the proposed instance-specific structure learning method for CBNs that contain latent confounders. Section 5 gives a quantitative assessment of the method using simulated and real data. Finally, Section 6 concludes the paper.

2 Related Work

A Bayesian network (BN) provides a compact representation to encode the conditional independence constraints that hold among a set of variables, which results in modular parameterization of complex high-dimensional systems. An ordinary BN structure, however, is unnecessarily strict, meaning that each independence constraint is included in the BN structure only when it holds for all combinations of values of the variables involved. A more expressive form of conditional independence, called context-specific independence (CSI), was introduced in [1] to allow local structures in the conditional probability distributions of the variables. A CSI relationship indicates that an independence constraint holds between a child variable and some but not necessarily all combinations of values of its parents (e.g., if $X \perp\!\!\!\perp Y|Z$ holds only when $Z = 1$, then this means that $P(X|Y, Z = 1) = P(X|Z = 1)$ but $P(X|Y, Z = 0) \neq P(X|Z = 0)$).

A number of algorithms have been investigated to build more flexible BN structures that are able to represent CSI relationships either implicitly (i.e., by using different representations of the conditional probability distributions) or explicitly (i.e., by altering the BN structure representation to encode CSI structures). One of the earliest methods was introduced by [7] that augments a BN structure with tree-structured conditional probability tables (CPTs). Later, [4] developed a more generalized version using decision-graph CPTs. These types of structured CPTs are used to partition the outcome space of the parents of a variable to characterize the regularities that exist in its CPT, which correspond to CSIs. Recently, [23] used Boolean functions to define the interactions among the parents of a variable to learn the local structures. [8] developed a graph-based method that represents CSI relationships using similarity networks and Bayesian multinets. Also, the method in [14] learns multiple BNs jointly from multiple datasets using integer linear programming. [17] introduced a modified representation of BNs to model the CSI structures in a single BN model. This representation adds labels to the edges of a BN to specify the contexts in which local structures exist; such graphs are called labeled directed acyclic graphs (LDAGs). [10] developed a constraint-based and an exact score-based CBN structure learning algorithm for LDAGs.

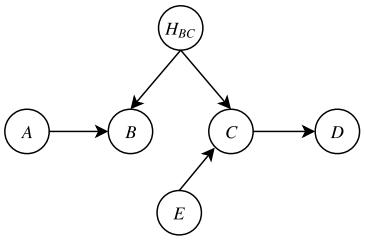
The methods in the previous paragraph try to capture all possible local structures in a single model by modifying either the representation of the graph structure or the local probability distributions. One major drawback of these approaches is the computational complexity overhead that the new representations add to the CBN structure learning task, which is already an NP-hard problem [2]. Additionally, none of these methods learns a CBN model that is specialized for a given test instance (e.g., a given patient), which is the main goal of the current paper. [13] introduced a distance matching regularizer to learn instance-specific regression models for each instance. However, this method learns a predictive model (not a CBN model), and it does not specifically capture the CSIs.

We previously introduced a score-based instance-specific CBN structure learning method, called IGES [12]. The IGES method searches directly for a CBN model that is tailored for a given test instance, rather than searching for all (or at least many) possible instance-specific models and then choosing the one that matches the current test instance, which is generally much more complicated. Recently, [5] introduced a tumor-specific causal inference algorithm using bipartite CBNs in which causes are at one level and effects are at another. This method also assumes that each effect variable has one and only one cause. Although these assumptions are reasonable for that application, they restrict generality of this method. Both IGES and tumor-specific causal inference methods assume no latent confounders among the variables. In this paper, we introduce a novel method that can discover causal graphs that model latent confounders.

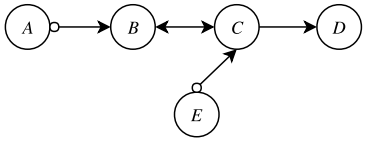
3 Background

Directed acyclic graphs (DAGs) and their Markov equivalence class (i.e., DAGs that have the same d-separation properties [22]) are the canonical graphical models that qualitatively represent the causal relationships among a set of variables when assuming that the true causal graph does not contain any latent confounders. This assumption is unrealistic and rarely holds in practice. Partial ancestral graphs (PAGs) [18] are graphical models that represent a Markov equivalence class of DAGs in the presence of latent variables. Conditional independence relationships in PAGs are represented using an expanded set of edge marks. Figure 1b shows an example of a PAG that models the

relationships among observed variables in the ground-truth DAG given in Figure 1a, where A to E are measured variables and H_{BC} is a latent variable. In Figure 1b, the subgraph $B \leftrightarrow C$ represents that B and C are both caused by one or more latent variables (i.e., they are confounded by a latent variable). The subgraph $C \rightarrow D$ represents that C is a cause of D. The subgraph $A \circ\!\!\rightarrow B$ represents that either A causes B, A and B are confounded by a latent variable, or both. Another edge possibility, which does not appear in the example, is $X \circ\!\!-\!\!\circ Y$, which is compatible with the true causal model having X as cause of Y, Y as a cause of X, a latent confounder of X and Y, or some acyclic combination of these three alternatives.



(a) The data-generating CBN $\mathcal{G}_1$.



(b) The ground-truth PAG $\mathcal{G}_2$ that is learnable in the large sample limit.

Figure 1: The PAG in (b) is learnable in the large sample limit from observational data generated by the causal model in (a), where H_{BC} is a latent confounder and the other variables are measured.

Major types of algorithms for learning a causal structure from observational data include score-based, constraint-based, and hybrid approaches. Score-based methods involve two main components: a scoring function and a search algorithm. Given a dataset of samples, and possibly prior knowledge or belief, a score is derived for each candidate causal structure, which reflects the goodness of fit of the causal structure given the dataset. The score is then incorporated into a search algorithm, which is often a greedy heuristic, to find the highest scoring causal structure in the hypothesis space of the possible structures. A constraint-based causal structure learning algorithm searches for an equivalence class of causal structures (CBNs), all of which entail a particular set of conditional independence constraints that are judged to hold in a dataset of samples, based on the results of statistical independence tests applied to the dataset. Constraint-based algorithms typically select a sufficient subset of constraints to test. Finally, several hybrid algorithms have been developed to combine the strengths of constraint-based and score-based methods in several ways.

In this paper, we use a variant of a hybrid causal structure learning algorithm called GFCI [15] that learns PAGs from data. In the following sections, we provide an overview of GFCI and the Bayesian independence test that we use to score constraints when applying GFCI.

3.1 Greedy Fast Causal Inference (GFCI). GFCI [15] is hybrid search algorithm that combines a score-based method, called greedy equivalence search (GES) [3], and a constraint-based method, called fast causal inference (FCI) [21]. It does so because GES is fast and effective at finding the variables that are directly dependent (i.e., have some type of edge between them) and FCI is effective at determining the specific edge to form a PAG. In the following paragraphs, we first briefly describe the GES and FCI methods, and then, we explain how these methods are combined in GFCI.

GES performs a two-stage search over the space of equivalence class of CBNs without latent variables (i.e., patterns). During the forward phase, it adds single edges to the current graph to generate the neighbor states. It then greedily changes the current graph with the neighbor state that leads to the greatest improvement in the score; this phase stops when no further improvement can be achieved. Similarly, during the backward phase, it removes single edges from the current graph to generate all possible neighbor states and replaces the current graph with the highest scoring neighbor state; it continues until no further improvement can be achieved and returns the resultant graph. The Bayesian information criterion (BIC) score [19] is often used to learn a CBN structure when variables follow a Gaussian distribution and the BDeu score [9] is frequently used for multinomial variables, although other scores are possible. For more information about this method see [3].

FCI is a two-phase algorithm that searches over PAGs. FCI starts off with a fully connected undirected graph. For each pair of adjacent variables $X - Y$, if it finds a subset $\boldsymbol{Z}$ that makes X and Y independent conditioned on $\boldsymbol{Z}$ (i.e., $X \perp\!\!\!\perp Y | \boldsymbol{Z}$), it deletes the edge and stores $\boldsymbol{Z}$. In the second phase, it applies a set of orientation rules to orient the endpoints based on the results of the first phase. As is typical of constraint-based causal discovery algorithms, FCI outputs a single graph

structure (PAG) and does not provide any information about the uncertainty of the edges in the structure.

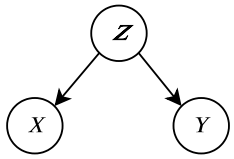
During the first step, GFCI applies GES to obtain a CBN structure; the GES search replaces the adjacency search of the FCI algorithm. The graph obtained from GES may contain extra edges and incorrect orientations if the model includes latent confounders. In the second step, GFCI uses the FCI algorithm to prune the extraneous edges and correct the orientations by performing a sequence of conditional independence tests, similar to the orientation phase of FCI. The GFCI algorithm outputs the correct PAG with probability 1.0 in the large sample limit, under assumptions [15]. However, GFCI still suffers from the same problem as most constraint-based approaches: It outputs a single PAG structure, without providing any quantification regarding how likely it is to be correct, relative to alternative PAGs. In the following section, we explain how to address this problem, using a Bayesian method for scoring independence constraints, which we introduced in [11].

3.2 Bayesian Scoring of Constraints. We previously introduced a hybrid PAG learning approach, called Bayesian scoring of constraints (BSC) [11]. BSC uses a Bayesian method to perform an independence test (described below) that can be incorporated into any search that requires independence testing (e.g., FCI or GFCI), rather than using a frequentist significance testing. We can compute the posterior probability of a PAG as the joint posterior probability of all the independence constraints that characterize that PAG [11], which is the major advantage of the BSC method.

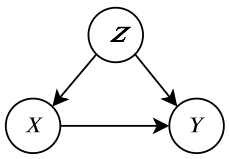
Let D be a dataset that is generated from a ground-truth CBN model, and let $r = (X \perp\!\!\!\perp Y|\boldsymbol{Z})$ be an arbitrary conditional independence constraint, where X and Y are variables of dataset D, and $\boldsymbol{Z}$ is a subset of variables of D that excludes X and Y. The goal is to determine whether this independence constraint holds given D, using a Bayesian scoring method. Let D_r be parts of D that are about r (i.e., data that corresponds to X, Y, and $\boldsymbol{Z}$). The posterior probability of r using Bayes' rule is as follows:

$$P(r|D_r) = \frac{P(D_r|r) \cdot P(r)}{P(D_r|r) \cdot P(r) + P(D_r|\bar{r}) \cdot P(\bar{r})} \quad . \tag{3.1}$$

Assuming that data are discrete, as we do in this paper, we can use the BDeu score [9] for deriving marginal likelihoods, i.e. $P(D_r|r)$ and $P(D_r|\bar{r})$, in Equation (3.1). To derive $P(D_r|r)$ (i.e., X is independent of Y given $\boldsymbol{Z}$), we score the BN structure that is shown in Figure 2a, in which $\boldsymbol{Z}$ is a set of parents for X and Y. Similarly, to compute $P(D_r|\bar{r})$ (i.e., X and Y are dependent given $\boldsymbol{Z}$), we score the BN structure that is shown in Figure 2b[1]. We assume that the prior probability of the constraint being true versus not true in Equation (3.1) is equally likely, and thus, we drop the terms $P(r)$ and $P(\bar{r})$ in that equation.



(a) The BN structure that corresponds to independence (i.e., $r = (X \perp\!\!\!\perp Y|\boldsymbol{Z})$).



(b) The BN structure that corresponds to dependence (i.e., $\bar{r} = (X \not\perp\!\!\!\perp Y|\boldsymbol{Z})$).

Figure 2: Independence and dependence structures that are used to score a constraint.

4 Instance-Specific GFCI (IS-GFCI)

In this section, we describe a novel instance-specific PAG structure learning algorithm that applies the idea of instance-specific modeling to GFCI. Instance-specific GFCI (IS-GFCI) takes as input a set D of observational training instances and a test instance T, and it returns as output an instance-specific PAG PAG_{IS}. This algorithm operates in two steps. In the first step (line 1 in Algorithm 1), it applies the population-wide GFCI algorithm (see Section 3.1) using the training set D. GFCI initially learns a population-wide CBN. More particularly, it learns an equivalence class of CBNs, which is a CBN in which for some edges the orientation may not be known, in which case the edge is represented as undirected (e.g., X — Y); such a generalized CBN is called a pattern. We denote this CBN as PAT_{pop}. Then, it performs additional conditional independence tests to obtain a population-wide PAG, which we denote as PAG_{pop}; we denote the set of constraints that correspond to PAG_{pop} by $\boldsymbol{r}_{pop}$, which can be obtained from PAG_{pop}. In the second step (line 2 in Algorithm 1), it applies GFCI with an instance-specific scoring function, called IS-Score, that we introduced in [12] (see below) and a novel instance-specific BSC test, called IS-BSC (see below), to find an instance-specific PAG, PAG_{IS} given D, T, and the population-wide models (i.e., PAT_{pop} and PAG_{pop}); we use the name GFCI2 to denote this application of GFCI. Algorithm 1 shows the high-level procedure of IS-GFCI algorithm.

[1] We also considered all possible graph structures that correspond to dependence and independence when scoring a constraint $r = (X \perp\!\!\!\perp Y|\boldsymbol{Z})$. On a selection of test cases, we found that it did not have a major effect on the results. Therefore, we used these two structures since it is simpler and more efficient for modeling a constraint.

Algorithm 1 IS-GFCI (D, T)

Input: dataset D, test case T
Output: A population-wide PAG PAG_{pop} and an instance-specific PAG PAG_{IS}.

1: PAT_{pop}, $PAG_{pop} \leftarrow$ GFCI(D)
2: PAT_{IS}, $PAG_{IS} \leftarrow$ GFCI2(D, T, PAT_{pop}, PAG_{pop})
3: return PAG_{pop} and PAG_{IS}

IS-Score is a Bayesian scoring function that is decomposable at the parent-child level (i.e., it computes the score for each variable X given its parents denoted by set $\boldsymbol{Z}$). IS-Score has two components for variable X. The first component scores the instance-specific parents of X, denoted by $\boldsymbol{Z}_{IS}$. To do so, IS-Score uses the cases in data D that match the current test case T. These cases are selected based on the values of $\boldsymbol{Z}_{IS}$ in T; let $\boldsymbol{j}$ be this set of values, and let $D_{Z_{IS}=j}$ be the instances that match T. IS-Score uses $D_{Z_{IS}=j}$ to score $[\boldsymbol{Z}_{IS} = \boldsymbol{j}] \rightarrow X$. The second component scores the population-wide parents of X, denoted by $\boldsymbol{Z}_{pop}$. IS-Score uses the remaining instances in D (i.e., $D_{Z_{IS}\neq j}$) to score the population-wide model $\boldsymbol{Z}_{pop} \rightarrow X$. The overall score for variable X is the product of these two scores. See [12] for a detailed description of the IS-Score procedure. This score is used in performing a GES-like greedy search, called IGES, which is described in detail in [12].

The IS-BSC procedure (defined below) scores an instance-specific independence constraint $r_{IS}^{(X,Y)} = (X \perp\!\!\!\perp Y | \boldsymbol{Z} = \boldsymbol{j})$. In such a constraint, the conditioning set $\boldsymbol{Z}$ takes specific values $\boldsymbol{j}$ that correspond to the values of $\boldsymbol{Z}$ in the given test case T. The basic idea behind IS-BSC is to find those cases in D in which $\boldsymbol{Z} = \boldsymbol{j}$ and use them to score the instance-specific constraint $r_{IS}^{(X,Y)}$. In essence, those instances in D form a cluster that are similar to instance T in the context of $\boldsymbol{Z} = \boldsymbol{j}$; we use that cluster to determine whether the independence constraint holds between (X, Y). Since those instances are being used to score this instance-specific constraint between (X, Y), in order to avoid duplicate scoring, they can no longer be used to also score the population-wide constraints in $\boldsymbol{r}_{pop}$ that contain independence queries about variables (X, Y) regardless of their conditioning sets; let $\boldsymbol{r}_{pop}^{(X,Y)}$ denote this subset of constraints. Therefore, the scores for constraints $\boldsymbol{r}_{pop}^{(X,Y)}$ must be adjusted accordingly.

More specifically, let $D_{Z=j}$ denote the instances in D in which $\boldsymbol{Z} = \boldsymbol{j}$ and $D_{Z\neq j}$ denote the remaining instances in D (line 1 in Algorithm 2). We use $D_{Z=j}$ to score an instance-specific independence constraint of the form $r_{IS}^{(X,Y)}$ (line 3 in Algorithm 2). Then, we use $D_{Z\neq j}$ to re-score the population-wide constraints $\boldsymbol{r}_{pop}^{(X,Y)}$ that are about (X, Y) (line 4 in Algorithm 2). The overall score for $r^{(X,Y)}$ is given as the product of the instance-specific score of $r_{IS}^{(X,Y)}$ and the population-wide scores of $\boldsymbol{r}_{pop}^{(X,Y)}$ (line 5 in Algorithm 2). The following equation derives the posterior probability of $r^{(X,Y)}$ using this method:

$$P(r^{(X,Y)}|D) = P(r_{IS}^{(X,Y)}|D_{Z=j}) \cdot P(\boldsymbol{r}_{pop}^{(X,Y)}|D_{Z\neq j}) \ , \tag{4.2}$$

where the computation of the terms on the right hand side can be done as described in Section 3.2. Algorithm 2 provides pseudo-code for IS-BSC method.

Algorithm 2 IS-BSC(D, T, $r_{IS}^{(X,Y)}$, $\boldsymbol{r}_{pop}$)

Input: training dataset D, test case T, an instance-specific constraint of the form $r_{IS}^{(X,Y)} = (X \perp\!\!\!\perp Y | \boldsymbol{Z} = \boldsymbol{j})$, a constraint set $\boldsymbol{r}_{pop}$ from a population-wide PAG
Output: the posterior probability of independence constraint $r^{(X,Y)}$

1: Derive $D_{Z=j}$ and $D_{Z\neq j}$ from D and the values $\boldsymbol{j}$ of $\boldsymbol{Z}$ in T
2: Derive the constraints $\boldsymbol{r}_{pop}^{(X,Y)} \in \boldsymbol{r}_{pop}$ that are about (X, Y)
3: $P(r_{IS}^{(X,Y)}|D_{Z=j}) \leftarrow$ Score the constraint $r_{IS}^{(X,Y)}$ using $D_{Z=j}$
4: $P(\boldsymbol{r}_{pop}^{(X,Y)}|D_{Z\neq j}) \leftarrow$ Score the constraints $r \in \boldsymbol{r}_{pop}^{(X,Y)}$ using $D_{Z\neq j}$
5: $P(r^{(X,Y)}|D) \leftarrow P(r_{IS}^{(X,Y)}|D_{Z=j}) \cdot P(\boldsymbol{r}_{pop}^{(X,Y)}|D_{Z\neq j})$
6: return $P(r^{(X,Y)}|D)$

5 Experiments

This section describes the experimental methods and results that we used to investigate the performance of the instance-specific GFCI (IS-GFCI) versus GFCI, which is a state-of-the-art, non-instance-specific PAG-learning algorithm. To do so, we used both simulated and real data, which are described below in Sections 5.1 and 5.2, respectively.

5.1 Simulation Experiments. To investigate the performance of IS-GFCI versus GFCI, we conducted simulation studies to generate data as follows.

1. We created random BNs with $V = \{10, 20\}$ discrete variables where each variable has 2, 3, or 4 categories, which is chosen randomly. The number of edges are $E = \{2V, 4V, 6V\}$. To generate a BN, we first cre-

ate an arbitrary ordering of variables; then we randomly add edges to BN in a forward direction until obtaining the specified number of edges. The BNs generated in this way have a power-law-type distribution over the number of parents, with some variables having many more than the average number of parents.

2. The BN structures were then parametrized to include context-specific independence (CSI) in the conditional probability tables of the variables that have more than one parent, where any such variable includes at least one CSI.

3. We randomly set $L = 20\%$ of variables to be latent (i.e., hidden). These variables were chosen at random from a list of all variables that are common causes of two or more of the measured variables.

4. We used each BN and its parameters, to generate a training dataset D with $N = \{500, 1000, 5000\}$ cases and a test dataset with $M = \{100\}$ cases; we refer to each instance in the test dataset as a case T. For $E = \{2V, 4V, 6V\}$, the average fraction of variables that exhibit CSI in the $M = 100$ instances of each simulated test set are 0.28, 0.38, and 0.40, respectively.

5. We used the training dataset D generated in step (4) to learn a population-wide BN structure using the GFCI algorithm (Section 3.1). GFCI uses a score-based (i.e., GES) and a constraint-based (i.e., FCI) in its two steps. For GES, we used the BDeu score [9] with prior equivalence sample size (PESS) of 1.0 to learn a population-wide pattern PAT_{pop}. For the independence testing used in FCI, we applied BSC (Section 3.2) with 0.5 decision threshold (i.e., if $P(r|D) \geq 0.5$ for a constraint $r = (X \perp\!\!\!\perp Y|\boldsymbol{Z})$, then BSC returns *true* for r, otherwise, it returns *false*). The final output of GFCI is a PAG model; we refer to this model as PAG_{pop}.

6. For each test instance T, we used T and the training dataset D generated in step (4) to learn an instance-specific BN structure using the IS-GFCI algorithm described in Section 4. Similar to GFCI, IS-GFCI uses a score-based (i.e., IGES [12]) and a constraint-based (i.e., FCI with IS-BSC independence test) method in its two steps. For IGES, we used PAT_{pop}, which is learned in the population-wide search, as the population-wide model; also, we set PESS = 1.0 and the structure prior $\kappa = 0.5$, where $0 < \kappa \leq 1$ is a penalty factor that is used when computing the prior probabilities of the instance-specific CBN structure; it penalizes the structural difference between the population-wide and instance-specific CBNs (see [12] for more details). For the FCI part, we used IS-BSC with the constraints that correspond to PAG_{pop}, which is learned in the population-wide search; we also set the decision point to 0.5 when performing independence tests. The final output of IS-GFCI is a PAG model for each test instance T; we refer to this model as PAG_{IS}.

7. Finally, we computed evaluation measures (described below) to compare the structure recovery performance of GFCI versus IS-GFCI. To do so, we obtained the ground-truth PAG structure (steps 1-3) of each test instance T considering the existing CSIs in T; we refer to this graph as PAG_{truth}. We compared PAG_{pop} and PAG_{IS} versus PAG_{truth} for each test case and reported the average of measures over $M = 100$ test cases.

For each simulation setting mentioned above, steps (1) through (7) were repeated for 10 randomly generated BNs and the performance results were averaged. The evaluation measures we used include precision (P) and recall (R) for edge adjacency and arrowhead orientation, which are calculated as follows:

$$\text{Adjacency P} = \frac{\#\text{correctly predicted adjacencies}}{\#\text{predicted adjacencies}}$$

$$\text{Adjacency R} = \frac{\#\text{correctly predicted adjacencies}}{\#\text{true adjacencies}}$$

$$\text{Arrowhead P} = \frac{\#\text{correctly predicted arrowheads}}{\#\text{predicted arrowheads}}$$

$$\text{Arrowhead R} = \frac{\#\text{correctly predicted arrowheads}}{\#\text{true arrowheads}}$$

For precision and recall evaluation measurements, we derived three subtypes: (1) using the subset of variables that include CSIs (denoted by *IS* subscript), (2) using the remaining variables that do not include CSI (denoted by *other* subscript), and (3) using all variables (without a subscript).

Tables 2a, 2b, and 2c show the adjacency and arrowhead P/R results (for all the three subtypes) for the simulated BNs with $V = \{10, 20\}$ variables and $E = \{2V, 4V, 6V\}$ edges, when using $N = 500$, $N = 1000$ and $N = 5000$ cases, respectively. For $N = 500$, IS-GFCI and GFCI perform similar in terms of adjacency P, but adjacency R is often higher when using GFCI. However, IS-GFCI performs better in terms of arrowhead P/R for $N = 500$ (Table 2a). As Table 2b indicates, when using $N = 1000$ training cases, IS-GFCI almost always performs better in terms of adjacency and arrowhead P/R for *IS* subtype, and overall. In this case, GFCI performs slightly better in terms of these measures for *other* subtype. Also, IS-GFCI almost always performs better in terms of adjacency recall and arrowhead recall with $N = 5000$ training samples.

We also used structural Hamming distance (SHD) to compare the structural differences between PAG_{pop} and PAG_{IS} versus PAG_{truth}. The SHD of each PAG from PAG_{truth} includes three types of edge modifications: added, deleted, and reversed edges, where sum of all these edge modifications is overall SHD; we call

this *strict* SHD (S-SHD). We also defined a *lenient* version of SHD (L-SHD), which allows general edges that include circle endpoints to be compatible with their specializations. For example, the L-SHD between $A \circ\!\!\rightarrow B$ and $A \rightarrow B$ is 0 because these edges are compatible. However, the L-SHD between $A \rightarrow B$ and $B \circ\!\!\rightarrow A$ is 1.

Table 1 shows the SHD results for the simulated BNs with $V = \{10, 20\}$ variables and $E = \{2V, 4V, 6V\}$ edges, using $N = \{500, 1000, 5000\}$ cases. GFCI usually performs better when using $N = 500$ cases. Both methods performed similarly in terms of SHD when using $N = 1000$ training samples; however, IS-GFCI has better SHD performance than GFCI when the sample size increases to $N = 5000$. In this case, the SHD difference gets larger as the graph has more variables and edges (e.g., $V = 20$, $E = 80$, and $N = 5000$ and $V = 20$, $E = 120$, and $N = 5000$).

Table 1: Average strict SHD (S-SHD) and lenient SHD (L-SHD) results over BNs with V variables and E edges. The best results in each cell are shown in bold (the lower the better).

		$N = 500$		$N = 1000$		$N = 5000$	
V, E	Method	S-SHD	L-SHD	S-SHD	L-SHD	S-SHD	L-SHD
10, 20	IS-GFCI	10.88	9.21	9.41	7.17	**8.91**	**6.66**
	GFCI	**10.56**	**8.29**	**9.01**	**6.33**	9.29	6.94
10, 40	IS-GFCI	17.03	13.98	16.93	**12.95**	**15.74**	**11.46**
	GFCI	**16.25**	**13.41**	**16.34**	13.45	17.48	12.61
10, 60	IS-GFCI	18.44	15.39	17.50	**13.43**	**19.95**	**14.06**
	GFCI	**17.23**	**15.19**	**17.18**	13.81	20.52	14.96
20, 40	IS-GFCI	25.25	21.38	**24.22**	19.46	**22.04**	**17.14**
	GFCI	**24.37**	**19.52**	25.16	**19.41**	24.07	18.72
20, 80	IS-GFCI	**52.42**	**45.99**	**50.10**	**42.96**	**49.09**	**39.63**
	GFCI	53.13	46.11	51.55	44.12	51.79	45.04
20, 120	IS-GFCI	67.47	60.61	66.12	**58.24**	**62.83**	**52.00**
	GFCI	**67.08**	**60.07**	**66.02**	59.11	63.31	56.54

5.2 Real Data Experiments. We also evaluated the proposed IS-GFCI method on multiple real-world datasets from the UCI repository [6]. The datasets we used were the Breast Cancer, Primary Tumor, Lymphography, SPECT Heart, and Audiology datasets. Table 3 shows the number of cases (N) and variables (V) in each of these datasets. We performed leave-one-out cross-validation on each of the datasets. For a given dataset D, we selected a single instance T and used it as the test instance; we used all the remaining instances as the training set D_{train}. Given each T, we learned an instance-specific PAG_{IS} model for T using IS-GFCI. We repeated this procedure for every instance in D. We also learned a population-wide PAG_{pop} model for all the instances in D using GFCI.

Since we do not know the true causal relationships in these datasets, we compare the average of structural differences between PAG_{IS} and PAG_{pop}, which are shown in Table 3. The results indicate that as the data includes more variables, the structural differences increase between PAG_{pop} and PAG_{IS}. Since we do not know the true causal structures for the real datasets, we cannot determine whether IS-GFCI or GFCI is performing better in learning the causal structures. The results do show, however, that instance-specific causal structure frequently exists when we learn PAGs from real-world data. In future studies, we plan to evaluate the extent to which instance-specific causal structures are correct when learning from real data for which the true causal structure is known. The simulation results reported above provide preliminary support that those studies will be positive.

6 Conclusions

The instance-specific IGES method that we introduced in [12] builds a causal model for a given instance assuming causal sufficiency, but this assumption rarely holds in practice. In the current paper, we introduced an instance-specific PAG-learning algorithm called IS-GFCI that outputs a PAG that is specific to a given instance T (e.g, a patient) by guiding causal model search based on the attributes of T. The approach we used to develop IS-GFCI is quite general and can be readily applied to develop an instance-specific version of other graphical causal discovery methods.

The empirical results we obtained on simulated data for discovering the instance-specific PAG structure of each test instance T indicate that when fewer samples are available (i.e., $N = 500$), IS-GFCI performs similar to GFCI in terms of adjacency P, but better than GFCI in terms of arrowhead P/R. However, IS-GFCI performs better in terms of adjacency and arrowhead P/R when the sample size increases to $N = 5000$. In terms of SHD, we found that GFCI performs better when using $N = 500$ training cases, where the differences are due to missing edges by IS-GFCI. We conjecture that the missing edges are weak enough to make instance-specific detection difficult without more samples. In that regard, we found that both methods perform similarly when using $N = 1000$ cases, whereas IS-GFCI performs better when the sample size increases to $N = 5000$.

The IS-GFCI method can be extended in numerous ways, including the following: (a) develop an instance-specific score to learn BN structures that contain other types of variables (e.g., continuous or a mixture of continuous and discrete variables), (b) develop more informative structure and parameter prior probabilities, and (c) extend the experimental evaluations. In summary, the current paper provides support that the IS-GFCI approach is a promising approach for discovering instance-specific PAG structures relative to a population-wide method, and thus, further investigation of the approach is warranted.

Table 2: Average adjacency and arrowhead precision (P) and recall (R) results over BNs with V variables and E edges. We derived three subtypes for P/R measurements using: (1) the subset of variables that include CSIs (denoted by *IS* subscript), (2) the remaining variables that do not include CSI (denoted by *other* subscript), and (3) all variables (without a subscript). The best results in each cell are shown in bold.

(a) Results when using $N = 500$ training cases.

		Adjacency						Arrowhead					
V, E	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10, 20	IS-GFCI	**0.94**	0.98	**0.96**	0.42	0.33	0.34	**0.24**	**0.36**	**0.38**	**0.10**	0.05	0.04
	GFCI	0.87	**1.00**	0.94	**0.52**	**0.43**	**0.43**	0.02	0.25	0.20	0.01	0.05	0.04
10, 40	IS-GFCI	0.89	1.00	0.93	0.36	0.30	0.31	**0.10**	**0.20**	**0.15**	**0.10**	**0.10**	**0.10**
	GFCI	0.89	1.00	0.93	**0.40**	**0.31**	**0.34**	0.01	0.02	0.02	0.04	0.01	0.01
10, 60	IS-GFCI	**0.94**	1.00	**0.96**	0.31	0.26	0.28	**0.23**	**0.19**	**0.22**	**0.21**	**0.04**	**0.10**
	GFCI	0.92	1.00	0.95	**0.37**	**0.27**	**0.30**	0.03	0.11	0.05	0.09	0.02	0.04
20, 40	IS-GFCI	0.78	0.97	0.90	0.35	0.36	0.36	**0.35**	0.68	**0.63**	0.18	**0.19**	**0.17**
	GFCI	**0.79**	**1.00**	0.90	**0.45**	**0.40**	**0.42**	0.24	**0.86**	0.58	**0.24**	0.18	0.16
20, 80	IS-GFCI	0.86	1.00	0.92	0.26	**0.23**	**0.24**	**0.51**	0.65	**0.60**	**0.22**	**0.11**	**0.12**
	GFCI	0.86	1.00	0.92	**0.28**	0.20	0.23	0.35	**0.68**	0.48	0.16	0.06	0.07
20, 120	IS-GFCI	0.90	0.99	0.93	0.23	0.16	0.19	**0.52**	**0.50**	**0.50**	**0.23**	**0.07**	**0.09**
	GFCI	**0.91**	**1.00**	**0.94**	**0.24**	0.16	**0.20**	0.43	0.45	0.41	0.16	0.05	0.07

(b) Results when using $N = 1000$ training cases.

		Adjacency						Arrowhead					
V, E	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10, 20	IS-GFCI	**0.93**	0.99	**0.96**	0.52	0.46	0.48	**0.32**	**0.50**	**0.50**	0.25	**0.15**	**0.18**
	GFCI	0.89	**1.00**	0.95	**0.63**	**0.54**	**0.57**	0.31	0.39	0.36	0.25	0.09	0.14
10, 40	IS-GFCI	0.86	0.99	0.92	0.39	**0.38**	**0.37**	**0.21**	**0.34**	**0.31**	**0.20**	**0.19**	**0.20**
	GFCI	**0.90**	**1.00**	**0.94**	**0.40**	0.30	0.33	0.10	0.17	0.14	0.11	0.06	0.07
10, 60	IS-GFCI	**0.83**	1.00	**0.92**	0.36	**0.36**	**0.36**	**0.22**	**0.31**	**0.23**	0.18	**0.13**	**0.14**
	GFCI	0.80	1.00	0.90	0.36	0.32	0.34	0.19	0.24	0.20	**0.21**	0.09	0.12
20, 40	IS-GFCI	**0.85**	0.95	**0.90**	0.44	0.45	0.43	**0.61**	**0.73**	**0.70**	**0.34**	**0.26**	**0.26**
	GFCI	0.77	**1.00**	0.88	**0.48**	**0.46**	**0.45**	0.28	0.68	0.48	0.31	0.21	0.21
20, 80	IS-GFCI	**0.86**	**0.98**	**0.90**	0.33	**0.23**	**0.27**	0.42	0.64	0.57	**0.27**	**0.13**	**0.15**
	GFCI	0.81	0.97	0.87	**0.34**	0.19	0.25	**0.51**	**0.68**	**0.63**	0.24	0.08	0.10
20, 120	IS-GFCI	0.87	0.97	0.91	**0.24**	**0.18**	**0.21**	**0.41**	**0.55**	**0.50**	**0.23**	**0.09**	**0.11**
	GFCI	0.87	**0.99**	0.91	0.22	0.14	0.18	0.33	0.47	0.41	0.15	0.05	0.06

(c) Results when using $N = 5000$ training cases.

		Adjacency						Arrowhead					
V, E	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10, 20	IS-GFCI	**0.92**	0.96	**0.95**	0.64	0.50	0.56	**0.54**	0.52	**0.65**	**0.48**	**0.34**	**0.38**
	GFCI	0.81	**1.00**	0.90	**0.66**	**0.54**	**0.58**	0.30	**0.60**	0.45	0.35	0.25	0.26
10, 40	IS-GFCI	**0.87**	0.99	**0.91**	**0.53**	**0.43**	**0.48**	**0.30**	**0.41**	**0.38**	0.42	**0.28**	**0.30**
	GFCI	0.74	**1.00**	0.83	0.45	0.39	0.42	0.22	0.28	0.26	0.42	0.23	0.26
10, 60	IS-GFCI	**0.85**	0.99	0.92	**0.43**	**0.41**	**0.42**	**0.18**	**0.30**	**0.27**	**0.29**	**0.24**	**0.24**
	GFCI	0.84	**1.00**	0.92	0.39	0.34	0.36	0.12	0.23	0.17	0.24	0.14	0.14
20, 40	IS-GFCI	**0.90**	0.98	**0.94**	0.50	**0.56**	**0.52**	**0.49**	0.78	**0.72**	0.46	**0.35**	**0.34**
	GFCI	0.73	0.98	0.86	**0.52**	0.51	0.50	0.27	**0.85**	0.63	**0.51**	0.29	0.29
20, 80	IS-GFCI	**0.89**	0.98	**0.93**	**0.42**	**0.35**	**0.37**	**0.49**	0.72	0.63	**0.41**	**0.26**	**0.28**
	GFCI	0.83	**1.00**	0.90	0.29	0.23	0.25	0.42	**0.80**	**0.64**	0.27	0.11	0.13
20, 120	IS-GFCI	**0.88**	0.98	**0.93**	**0.33**	**0.27**	**0.30**	**0.51**	0.64	0.59	**0.39**	**0.18**	**0.21**
	GFCI	0.82	**1.00**	0.89	0.27	0.17	0.21	0.44	**0.77**	**0.62**	0.25	0.08	0.10

Table 3: Average strict SHD (S-SHD) and lenient SHD (L-SHD) distance between PAG_{pop} and PAG_{IS} using leave-one-out cross-validation on UCI datasets. N and V denote the number of cases and variables, respectively.

UCI dataset	N	V	Added	Removed	Reoriented	S-SHD	L-SHD
Breast Cancer	286	10	0.60	1.70	1.51	3.81	2.30
Primary Tumor	339	18	4.47	0.83	2.14	7.43	5.57
Lymphography	148	19	5.41	1.81	2.62	9.84	7.40
SPECT Heart	267	23	10.08	2.84	13.77	26.69	12.93
Audiology	200	70	32.47	2.67	10.03	45.16	35.63

Acknowledgements

Research reported in this publication was supported by grant U54HG008540 from the National Institutes of Health (NIH), grant IIS-1636786 from the National Science Foundation (NSF), grant #4100070287 from the Pennsylvania Department of Health (PA DOH), and grant PA-18-02-01 (ASKE) from the Defense Advanced Research Projects Agency (DARPA). The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH, NSF, PA DOH, or DARPA.

References

[1] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, pages 115–123. Morgan Kaufmann Publishers Inc., 1996.

[2] D. M. Chickering. Learning Bayesian networks is NP-complete. In *Learning from Data*, pages 121–130. Springer, 1996.

[3] D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2003.

[4] D. M. Chickering, D. Heckerman, and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 80–89. Morgan Kaufmann Publishers Inc., 1997.

[5] G. Cooper, C. Cai, and X. Lu. Tumor-specific causal inference (TCI): A Bayesian method for identifying causative genome alterations within individual tumors. *bioRxiv*, page 225631, 2018.

[6] D. Dua and C. Graff. UCI machine learning repository, 2017.

[7] N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In *Learning in Graphical Models*, pages 421–459. Springer, 1998.

[8] D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence*, 82(1-2):45–74, 1996.

[9] D. Heckerman. A tutorial on learning with Bayesian networks. In *Learning in Graphical Models*, pages 301–354. Springer, 1998.

[10] A. Hyttinen, J. Pensar, J. Kontinen, and J. Corander. Structure learning for Bayesian networks over labeled DAGs. In *International Conference on Probabilistic Graphical Models*, pages 133–144, 2018.

[11] F. Jabbari, J. Ramsey, P. Spirtes, and G. Cooper. Discovery of causal models that contain latent variables through Bayesian scoring of independence constraints. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 142–157. Springer, 2017.

[12] F. Jabbari, S. Visweswaran, and G. F. Cooper. Instance-specific Bayesian network structure learning. *Proceedings of machine learning research*, 72:169, 2018.

[13] B. J. Lengerich, B. Aragam, and E. P. Xing. Personalized regression enables sample-specific pan-cancer analysis. *Bioinformatics*, 34(13):i178–i186, 2018.

[14] C. J. Oates, J. Q. Smith, S. Mukherjee, and J. Cussens. Exact estimation of multiple directed acyclic graphs. *Statistics and Computing*, 26(4):797–811, 2016.

[15] J. M. Ogarrio, P. Spirtes, and J. Ramsey. A hybrid causal search algorithm for latent variable models. In *Conference on Probabilistic Graphical Models*, pages 368–379, 2016.

[16] J. Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 2009.

[17] J. Pensar, H. Nyman, T. Koski, and J. Corander. Labeled directed acyclic graphs: a generalization of context-specific independence in directed graphical models. *Data Mining and Knowledge Discovery*, 29(2):503–533, 2015.

[18] T. Richardson, P. Spirtes, et al. Ancestral graph Markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

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

[20] P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, prediction, and search*. MIT Press, 2000.

[21] P. Spirtes, C. Meek, and T. Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh conference on Uncertainty in Artificial Intelligence*, pages 499–506. Morgan Kaufmann Publishers Inc., 1995.

[22] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 255–270. Elsevier Science Inc., 1991.

[23] Y. Zou, J. Pensar, and T. Roos. Representing local structure in Bayesian networks by Boolean functions. *Pattern Recognition Letters*, 95:73–77, 2017.