TECHNICAL REPORT
R-33-L
June 2018

Budgeted Experiment Design for Causal Structure Learning

AmirEmad Ghassami [1] Saber Salehkaleybar [2] Negar Kiyavash [1] Elias Bareinboim [3]

Abstract

We study the problem of causal structure learning when the experimenter is limited to perform at most k non-adaptive experiments of size 1. We formulate the problem of finding the best intervention target set as an optimization problem, which aims to maximize the average number of edges whose directions are resolved. We prove that the corresponding objective function is submodular and a greedy algorithm suffices to achieve $(1-\frac{1}{e})$-approximation of the optimal value. We further present an accelerated variant of the greedy algorithm, which can lead to orders of magnitude performance speedup. We validate our proposed approach on synthetic and real graphs. The results show that compared to the purely observational setting, our algorithm orients the majority of the edges through a considerably small number of interventions.

1. Introduction

The problem of learning the causal relations underlying a complex system is of great interest in AI and throughout the empirical sciences. Causal systems are commonly represented by directed acyclic graphs (DAGs), where the vertices are random variables and an edge from variable X to Y indicates that variable X is a direct cause of Y (Pearl, 2009; Spirtes et al., 2000; Bareinboim & Pearl, 2016).

To uncover the causal relations among a set of variables, if restricted to work with only observational data from the variables, one can use a constraint-based algorithm such as IC, IC* (Pearl, 2009), and PC, FCI (Spirtes et al., 2000), or a score-based methods, including (Meek, 1997; Chickering, 2002; Tian et al., 2012; Solus et al., 2017). Such purely observational approaches reconstruct the causal graph up to the Markov equivalence class, and hence, the investigator is commonly left with some (or in some cases many) unresolved causal relations. Albeit, under some extra assumptions, in some compelling settings, full structure learning using merely observational data is feasible (Mooij et al., 2016; Shimizu et al., 2006; Hoyer et al., 2009; Peters & Bühlmann, 2012; Ghassami et al., 2017; Quinn et al., 2015; Sun et al., 2015).

On the other hand, it is well-understood that whenever the investigator can perform sufficient number of interventions, the causal graph representing the underlying system can be fully recovered. There is a growing body of research on learning causal structures using interventional data in causally sufficient (Eberhardt, 2007; Hauser & Bühlmann, 2012; Cooper & Yoo, 1999; He & Geng, 2008), and causally insufficient systems (with latent variables) (Kocaoglu et al., 2017b). An interventional structure learning approach requires performing a set of experiments, each intervening on a subset of the variables, and subsequently collecting data from the intervened system. In this setting, two natural questions arise:

1. What is the smallest required number of experiments in order to fully learn the underlying causal graph?
2. For a fixed number of experiments (budget), what portion of the causal graph is learnable?

The first problem has been addressed in the literature under different assumptions. (Eberhardt et al., 2005) obtained the worst case bounds on the number of required experiments. Connections between the experiment design problem and the problem of finding a separating system in a graph was studied in (Eberhardt, 2007; Hyttinen et al., 2013). Adaptive algorithms for experiment design were proposed in (Hauser & Bühlmann, 2014; Shanmugam et al., 2015). In (Shanmugam et al., 2015), the authors present information-theoretic lower bounds on the number of required experiments for both deterministic and randomized adaptive approaches. In (Kocaoglu et al., 2017a), the authors considered costs for intervening on each variable and derived an experiment design algorithm with minimum total cost that reconstructs the whole structure.

The second question mentioned above has received less attention. We address the second question herein. Specifically, we consider a setup with budget limitation of k experiments.

[1] Department of ECE, and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA [2] Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran [3] Department of Computer Science, Purdue University, West Lafayette, IN, USA. Correspondence to: AmirEmad Ghassami <ghassam2@illinois.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

In some applications (for instance, reconstructing gene regulatory networks from knockout data in biology), performing simultaneous interventions on multiple variables is not always feasible. To insure that our model is applicable to such settings, we set each experiment to contain exactly one intervention. Even when more interventions are allowed per experiment, this constraint allows us to lower bound the fraction of the causal graph that is learnable. This is a distinctive feature of our work since most of the literature assumes that the size of each experiment is larger than one, in some cases going as high as half of the variables. Note that our results cannot be designed by limiting the size of experiments in those approaches to one, as that would result in trivial designs for experiments, such as requiring to intervene on all the variables. The authors of (Kocaoglu et al., 2017a) also considered the case in which the number of experiments is limited. However, each experiment in their setup is allowed to include intervening on arbitrary number of variables. We put the budget restriction on the number of variables to be randomized, i.e., the number of interventions (as opposed to the number of experiments), which we believe is a more natural budgeting constraint[1].

Contributions. In our interventional structure learning algorithm, first an observational test, such as PC algorithm (Spirtes et al., 2000), is performed on the set of variables. This test learns the skeleton as well as the orientation of some of the edges of the causal graph. Based on the result of the initial observational stage, the complete set of k experiments is designed. The more formal description of the problem statement is provided in Section 2. Our main contributions are summarized bellow:

- We cast the problem of finding the best intervention target set as an optimization problem which aims to maximize the average number of edges whose directions are resolved.
- We prove that the corresponding objective function is monotonically increasing and submodular. This implies that a general greedy algorithm is a $(1 - \frac{1}{e})$-approximation.
- Since computing the objective function is, in general, intractable for a given set, we propose an unbiased estimator of this function, which for graphs with bounded degree, has a polynomial time complexity. For graphs with high degree, we provide another efficient, albeit slightly biased estimator.

We implement an accelerated variant of the general greedy algorithm through *lazy* evaluations, originally proposed by Minoux (Minoux, 1978). This algorithm can lead to orders of magnitude performance speedup. Using synthetic and real data, in Section 5, we show that the proposed approach recovers a significant portion of the edges by performing only a few interventions in the underlying causal system.

2. Problem Description

We denote an undirected graph with a pair $G = (V, E)$, and a directed graph with a pair $G = (V, A)$, where V is the vertex set, and E and A are sets of undirected and directed edges, respectively. A mixed graph $G = (V, E, A)$, comprises both undirected and directed edges.

We use the language of Structural Causal Models (SCM) (Pearl, 2009). Formally, a SCM is a 4-tuple $\langle U, V, F, P(U)\rangle$, where U is a set of exogenous (latent) variables and V is a set of endogenous (measured) variables. F represents a collection of functions $F = \{f_v\}$ such that each endogenous variable X_v is determined by a function $f_v \in F$, where f_v is a mapping from the respective domain of $U_v \cup PA_v$ to X_v, where $U_v \subseteq U$, $PA_v \subseteq V \backslash X_v$, and the entire set F forms a mapping from U to V. The uncertainty is encoded through a probability distribution over the exogenous variables, $P(U)$. Each SCM induces a causal graph G, where vertex v corresponds to endogenous variable X_v, and the arguments of the functions correspond to its parent set. We will refer to variables and their corresponding vertices interchangeably. We consider causally sufficient systems in which the exogenous variables are independent. We also assume the observational and experimental distributions are faithful (Spirtes et al., 2000). For a detailed discussion on the properties of structural models, we refer readers to (Pearl, 2009). We introduce next other definitions that will be used throughout the paper.

Definition 1. *Two causal DAGs G_1 and G_2 over V are Markov equivalent if they represent the same set of conditional independence assertions. The set of all graphs over V is partitioned into a set of mutually exclusive and exhaustive Markov equivalence classes (Koller & Friedman, 2009).*

Definition 2. *The essential graph of G, denoted by $Ess(G)$, is a mixed graph in which the directed edges are those that have the same direction in all elements of the Markov equivalence class of G, and the undirected edges are those whose direction differ in at least two elements of the class (Andersson et al., 1997).*

Definition 3. *A complete conditional independence-based (CCI) algorithm is an observational structure learning algorithm resulting in learning the Markov equivalence class of the ground truth DAG.*

[1] One less usual connection is with the literature concerned with the influence maximization problem. The goal in the latter is to find k vertices (seeds) in a given network such that under a specified influence model, the expected number of vertices influenced by the seeds is maximized (Kempe et al., 2003; Leskovec et al., 2007; Chen et al., 2009). Besides the interpretative differences, an important distinction between the two problems is that in maximum influence problem, the goal is to spread the influence to the vertices of the graph, while in budgeted experiment design problem, the goal is to pick the initial k vertices in a way that leads to discovering the orientation of as many edges as possible. Therefore, the optimal solution to these two problems for a given graph can be quite different (see the supplementary materials for an example).

We denote the underlying true causal structure (ground truth DAG) by G^*. After performing CCI on the data from G^*, we denote the revealed set of directed edges by $A(Ess(G^*))$ and the set of undirected edges by $E(Ess(G^*))$. In general, the size of Markov equivalence classes can be very large. For instance, as observed in (He et al., 2015), the sizes of classes with even sparse essential graphs grow approximately exponentially with the order of the graph. Interventions will allow us to differentiate among the different causal structures within a Markov equivalence class.

We use the same notion of ideal intervention as in (Eberhardt et al., 2005; Pearl, 2009). For an intervention on variable X, denoted by $do(X)$, the influence of all the variables on the target variable X is removed, and X is randomized by forcing values from an independent distribution on it. This intervention changes the joint distribution of all variables in the system for which X is a direct or indirect cause, and results in interventional joint distribution $P(V|do(X))$. In our terminology, an intervention is always on a single variable. An interventional structure learning algorithm consists of a set of k experiments[2] $\mathcal{E} = \{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_k\}$, where each experiment $\mathcal{E}_i$ contains m_i interventions, which are performed simultaneously, i.e., $\mathcal{E}_i = \{v_1^{(i)}, v_2^{(i)}, ..., v_{m_i}^{(i)}\}$, for $1 \leq i \leq k$. More precisely, in experiment $\mathcal{E}_i$ data is drawn from distribution $P(V|do(X_{v_1^{(i)}}, X_{v_2^{(i)}}, ..., X_{v_{m_i}^{(i)}}))$.

A structure learning algorithm may be adaptive, in which case the experiments are performed sequentially and the information obtained from the previous experiments is used to design the next one, or passive, in which case all the experiments are designed in one shot. The approach we take here is to first perform a CCI algorithm to learn the skeleton and the direction of the edges in $A(Ess(G^*))$, and then design the experiments in a passive manner[3] under a budget constraint. This approach gives the experimenter the ability to perform the experiments in parallel without the need to wait for the result of one experiment to choose the next one. For example, in the study of gene regulatory networks (GRNs), when the GRN of all cells are the same, experiments can be performed simultaneously on different cells.

We consider a setup in which the number of interventions is fixed at k, and seek to design a set of experiments that allows learning the directions of as many edges in $E(Ess(G^*))$ as possible. We focus on single-intervention experiment setup in which for all experiments, $m_i = 1$. Therefore, our experiment set is of the form $\mathcal{E} = \{\{v_1^{(1)}\}, \{v_1^{(2)}\}, ..., \{v_1^{(k)}\}\}$. To simplify the notation, we denote the set of intervention targets as $\mathcal{I} = \{v_1, ..., v_k\}$, where $v_i = v_1^{(i)}$. As shown in (Eberhardt, 2007; Hyttinen et al., 2013), observing the result of the null experiment, allows orientating the edge $\{u, v\} \in E(Ess(G^*))$, if there exists $\mathcal{E}_l \in \mathcal{E}$ such that $(u \in \mathcal{E}_l \,,\, v \notin \mathcal{E}_l)$ or $(v \in \mathcal{E}_l \,,\, u \notin \mathcal{E}_l)$. On the other hand, if for all experiments $\mathcal{E}_l \in \mathcal{E}$ both $u \in \mathcal{E}_l$ and $v \in \mathcal{E}_l$, the orientation of $\{u, v\}$ cannot be learned. An experiment in which both u and v are intervened on is called a zero information experiment for u and v. Our setup in which $m_i = 1$, for all $i \in \{1, ..., k\}$, avoids such zero information experiments. Specifically, we focus on the following problem: *If the experimenter is allowed to perform k experiments, each of size 1, what portion of the graph could, on average, be reconstructed?* We formalize the problem statement in the rest of this section.

As discussed earlier, we only intervene on a single variable in each experiment. Hence, due to avoiding the issue of zero information experiments (Eberhardt et al., 2005), an experiment with intervention on vertex v will lead to learning the orientation of all the edges intersecting with v. Therefore, the entire experiment set $\mathcal{I} \subseteq V$ leads to learning the orientation of all the edges intersecting with members of $\mathcal{I}$. We denote the set of these learned directed edges by $A_{G^*}^{(\mathcal{I})}$ (which clearly depends on the structure of the true DAG G^*). Note that after learning the edges in $A_{G^*}^{(\mathcal{I})}$, we could also possibly resolve the direction of more edges by applying the Meek rules (Verma & Pearl, 1992; Meek, 1997) to $A_{G^*}^{(\mathcal{I})} \cup A(Ess(G^*))$, set of all directed edges after intervention $\mathcal{I}$. Let $H = (V(H), E(H))$ denote the undirected subgraph of $Ess(G^*)$. For any given set of directed edges $\mathcal{A}$ from the true DAG G^*, define $R(\mathcal{A}, G^*)$ as the subset of $E(H)$, whose directions can be learned by applying Meek rules starting from the set of directed edges $\mathcal{A} \cup A(Ess(G^*))$. Using this notation, experiment $\mathcal{I}$ results in learning the direction of edges in $R(A_{G^*}^{(\mathcal{I})}, G^*)$.

Let $D(\mathcal{I}, G^*) = |R(A_{G^*}^{(\mathcal{I})}, G^*)|$, i.e., the cardinality of $R(A_{G^*}^{(\mathcal{I})}, G^*)$, and let $\mathcal{G}$ denote the set of all DAGs in the Markov equivalence class of G^*. As we do not know the ground truth DAG, and since there is no preference between the members of the Markov equivalence class, G^* is equally likely to be any of the DAGs in $\mathcal{G}$. Hence, the expected number of the edges recovered through the experiment $\mathcal{I}$ is

$$\mathcal{D}(\mathcal{I}) := \mathbb{E}_{G_i}[D(\mathcal{I}, G_i)] = \frac{1}{|\mathcal{G}|} \sum_{G_i \in \mathcal{G}} D(\mathcal{I}, G_i). \tag{1}$$

Thus, our problem of interest can be formulated as finding some intervention target set $\mathcal{I} \subseteq V$ of cardinality k that maximizes $\mathcal{D}(\mathcal{I})$:

$$\max_{\mathcal{I}: \mathcal{I} \subseteq V} \mathcal{D}(\mathcal{I}) \quad \text{s.t.} \quad |\mathcal{I}| = k. \tag{2}$$

[2] Note that in most of the related work, each intervention is what we refer to as experiment here. That is, each intervention can contain many variables that are randomized simultaneously, while in our terminology, as intervention is always on a single variable.

[3] This approach is referred to as the passive setup by (Shanmugam et al., 2015), while (Eberhardt et al., 2005) uses the term passive for a setting in which the interventions are selected without performing the null experiment.

This is a challenging optimization problem for two reasons: First, finding an optimal $\mathcal{I}$ requires a combinatorial search. Second, even for a given set $\mathcal{I}$, computing $\mathcal{D}(\mathcal{I})$ when the value of k or the cardinality of the Markov equivalence class is large, can be computationally intractable.

3. Proposed Approach

We start by defining monotonicity and submodularity properties for a set function.

Definition 4. *A set function $f : 2^V \to \mathbb{R}$ is monotonically increasing if for all sets $\mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq V$, we have $f(\mathcal{I}_1) \leq f(\mathcal{I}_2)$.*

Definition 5. *A set function $f : 2^V \to \mathbb{R}$ is submodular if for all subsets $\mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq V$ and all $v \in V \setminus \mathcal{I}_2$[4],*

$$f(\mathcal{I}_1 \cup \{v\}) - f(\mathcal{I}_1) \geq f(\mathcal{I}_2 \cup \{v\}) - f(\mathcal{I}_2).$$

Nemhauser et al. showed that if f is a submodular, monotonically increasing set function with $f(\emptyset) = 0$, then the set $\hat{\mathcal{I}}$ with $|\hat{\mathcal{I}}| = k$ found by the greedy algorithm satisfies $f(\hat{\mathcal{I}}) \geq (1 - \frac{1}{e}) \max_{\mathcal{I}:|\mathcal{I}|=k} f(\mathcal{I})$ (Nemhauser et al., 1978). That is, the greedy algorithm is a $(1 - \frac{1}{e})$-approximation algorithm. We will use this result in our proposed approach.

We will show that the set function $\mathcal{D}$ is monotonically increasing and submodular, and hence, the greedy algorithm is a $(1 - \frac{1}{e})$-approximation algorithm to the maximization problem (2).

Lemma 1. *The set function $\mathcal{D}$ defined in (1) is monotonically increasing, i.e., for sets $\mathcal{I}_1 \subseteq \mathcal{I}_2$, we have*

$$\mathcal{D}(\mathcal{I}_2) \leq \mathcal{D}(\mathcal{I}_1).$$

See the supplementary materials for the proof.

The following lemma plays a fundamental role in the proof of submodularity of the set function $\mathcal{D}$.

Lemma 2. *For sets $\mathcal{I}_1, \mathcal{I}_2 \subseteq V$,*

$$R(A_{G^*}^{(\mathcal{I}_1 \cup \mathcal{I}_2)}, G^*) = R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*).$$

See the supplementary materials for the proof.

Interpreting $R(A_{G^*}^{(\mathcal{I})}, G^*)$ as the information obtained by intervening on set $\mathcal{I}$, Lemma 2 indicates that the result of two simultaneous interventions does not generate any new information which was not provided by the union of the information of each of the interventions.

Theorem 1. *The set function $\mathcal{D}$ defined in (1) is a submodular function.*

Algorithm 1 General Greedy Algorithm

```
Input: Joint distribution over $V$, and budget $k$.
Obtain $Ess(G^*)$ by performing a CCI algorithm.
initiate: $\mathcal{I}_0 = \emptyset$
for $i = 1$ to $k$ do
    $v_i = \arg\max_{v \in V \setminus \mathcal{I}_{i-1}} \hat{\mathcal{D}}(\mathcal{I}_{i-1} \cup \{v\}) - \hat{\mathcal{D}}(\mathcal{I}_{i-1})$
    $\mathcal{I}_i = \mathcal{I}_{i-1} \cup \{v_i\}$
end for
Output: $\hat{\mathcal{I}} = \mathcal{I}_k$
```

Proof. Due to Lemma 1, it suffices to show that for $\mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq V$, and $v \in V$, we have $\mathcal{D}(\mathcal{I}_1 \cup \{v\}) - \mathcal{D}(\mathcal{I}_1) \geq \mathcal{D}(\mathcal{I}_2 \cup \{v\}) - \mathcal{D}(\mathcal{I}_2)$. First we show that for a given directed graph $G_i \in \mathcal{G}$ the function $D(\mathcal{I}, G_i)$ is a submodular function of $\mathcal{I}$. From Lemma 2, we have $R(A_{G_i}^{(\mathcal{I}_1 \cup \{v\})}, G_i) = R(A_{G_i}^{(\mathcal{I}_1)}, G_i) \cup R(A_{G_i}^{(\{v\})}, G_i)$. Therefore,

$$\begin{aligned} &D(\mathcal{I}_1 \cup \{v\}, G_i) - D(\mathcal{I}_1, G_i) \\ &\quad = |R(A_{G_i}^{(\mathcal{I}_1 \cup \{v\})}, G_i)| - |R(A_{G_i}^{(\mathcal{I}_1)}, G_i)| \\ &\quad = |R(A_{G_i}^{(\mathcal{I}_1)}, G_i) \cup R(A_{G_i}^{(\{v\})}, G_i)| - |R(A_{G_i}^{(\mathcal{I}_1)}, G_i)| \\ &\quad = |R(A_{G_i}^{(\{v\})}, G_i)| - |R(A_{G_i}^{(\mathcal{I}_1)}, G_i) \cap R(A_{G_i}^{(\{v\})}, G_i)|. \end{aligned}$$

Similarly,

$$\begin{aligned} &D(\mathcal{I}_2 \cup \{v\}, G_i) - D(\mathcal{I}_2, G_i) \\ &\quad = |R(A_{G_i}^{(\{v\})}, G_i)| - |R(A_{G_i}^{(\mathcal{I}_2)}, G_i) \cap R(A_{G_i}^{(\{v\})}, G_i)|. \end{aligned}$$

Since $\mathcal{I}_1 \subseteq \mathcal{I}_2$, as seen in the proof of Lemma 1, $R(A_{G_i}^{(\mathcal{I}_1)}, G_i) \subseteq R(A_{G_i}^{(\mathcal{I}_2)}, G_i)$. Therefore, $-|R(A_{G_i}^{(\mathcal{I}_1)}, G_i) \cap R(A_{G_i}^{(\{v\})}, G_i)| \geq -|R(A_{G_i}^{(\mathcal{I}_2)}, G_i) \cap R(A_{G_i}^{(\{v\})}, G_i)|$, which implies that

$$D(\mathcal{I}_1 \cup \{v\}, G_i) - D(\mathcal{I}_1, G_i) \geq D(\mathcal{I}_2 \cup \{v\}, G_i) - D(\mathcal{I}_2, G_i).$$

This together with the fact that the function $D(\mathcal{I}, G_i)$ is a monotonically increasing function of $\mathcal{I}$ (observed in the proof of Lemma 1) shows that $D(\mathcal{I}, G_i)$ is a submodular function of $\mathcal{I}$. Finally, we have $\mathcal{D}(\mathcal{I}) = \frac{1}{|\mathcal{G}|} \sum_{G_i \in \mathcal{G}} D(\mathcal{I}, G_i)$. Since a non-negative linear combination of submodular functions is also submodular, the proof is concluded. □

Our General Greedy Algorithm is presented in Algorithm 1. Define the marginal gain of variable v when the previous chosen set is $\mathcal{I}$ as $\Delta_v(\mathcal{I}) = \mathcal{D}(\mathcal{I} \cup \{v\}) - \mathcal{D}(\mathcal{I})$. The greedy algorithm iteratively adds a variable which has the largest marginal gain to the intervention target set until it runs out of budget. However, as mentioned in Section 2, another issue regarding solving the optimization problem (2) is the computational intractability of calculating $\mathcal{D}(\mathcal{I})$ for a given intervention target set $\mathcal{I}$. We propose running Monte-Carlo simulations of the intervention model for sufficiently large number of times to obtain an accurate estimation of $\mathcal{D}(\mathcal{I})$. The pseudo-code of our estimator is presented in Subroutine 1. In this subroutine, for the given essential graph $Ess(G^*)$,

[4] If f is monotonically increasing, the definition relaxes to $v \in V$.

we generate N DAGs from the Markov equivalence class of G^*. The generated DAGs are kept in a multiset $\mathcal{G}'$. Note that $\mathcal{G}'$ is a multiset in which repetition is allowed, and operator $\uplus$ in the pseudo-code indicates the multiset addition. Finally, we calculate the estimated value $\hat{\mathcal{D}}(\mathcal{I})$ on $\mathcal{G}'$ instead of $\mathcal{G}$ as $\hat{\mathcal{D}}(\mathcal{I}) = \frac{1}{|\mathcal{G}'|}\sum_{G'_i\in\mathcal{G}'} D(\mathcal{I}, G'_i)$.

We use the unbiased sampler introduced in (Ghassami et al., 2018) to generate uniform samples from the Markov equivalence class. For the sake of completeness of the presentation, we briefly describe the idea in this sampler: Let $H = (V(H), E(H))$ denote the undirected subgraph of $Ess(G^*)$. Note that in general, H can be disconnected, with the set of its components denoted by *comp*(H). Note that each of these components is an essential graph. Chickering showed that learning the direction of any edge in one component of H will not reveal any information about the direction of edges in the other components (Chickering, 2002). Therefore, we can orient the edges in each component independently. All the members of a Markov equivalence class agree on v-structures[5] (Verma & Pearl, 1991). Therefore, since the essential graphs corresponding to the components of H are undirected, all the members of their corresponding class should be v-structure-free. Therefore, in each member, there is only one source vertex[6], and once a source is determined, an edge could be oriented as long as its endpoints are not at equal distance from the source (Ghassami et al., 2018). This is achieved by function FLOWED(v, G) in the algorithm, where G is a connected undirected essential graph and v is the source vertex. Function W uses FLOWED to find W(v, G), which is the number of members of the class in which v is the source vertex. This calculation is done in a recursive manner (see (Ghassami et al., 2018) for the details). Finally, function RANDEDGE sets a vertex v^* as the source vertex with probability proportional to W(v^*, G), and saves the direction of resolved edges in set A until all the edges are directed.

The used sampler satisfies $P(G' = G) = 1/|\mathcal{G}|$ (Ghassami et al., 2018); hence for any $\mathcal{I} \subseteq V$, $\hat{\mathcal{D}}(\mathcal{I})$ obtained from Subroutine 1 is an unbiased estimate of $\mathcal{D}(\mathcal{I})$, i.e., $\mathbb{E}[\hat{\mathcal{D}}(\mathcal{I})] = \mathcal{D}(\mathcal{I})$. To show the unbiasedness, suppose G' is a random generated graph in the subroutine. The result is immediate from the fact that

$$\mathbb{E}[D(\mathcal{I}, G')] = \sum_{G\in\mathcal{G}} P(G' = G) D(\mathcal{I}, G)$$
$$= \frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}} D(\mathcal{I}, G) = \mathcal{D}(\mathcal{I}).$$

Since we use random sampling in a greedy algorithm, we term our proposed approach the *Random Greedy Intervention Design* (Ran-GrID).

Subroutine 1 Unbiased $\mathcal{D}(\mathcal{I})$ Estimator Subroutine

```
Input: $Ess(G^*)$, intervention target set $\mathcal{I}$, and $N$.
initiate: $\mathcal{G}' = \emptyset$
for $i = 1$ to $N$, do
    $A = A(Ess(G^*)) \bigcup_{\hat{H}\in comp(H)}$ RANDEDGE$(\hat{H}, \emptyset)$.
    $G'_i = (V(Ess(G^*), A)$.
    $\mathcal{G}' = \mathcal{G}' \uplus G'_i$
end for
Output: $\hat{\mathcal{D}}(\mathcal{I}) = \frac{1}{|\mathcal{G}'|}\sum_{G'_i\in\mathcal{G}'} D(\mathcal{I}, G'_i)$

function FLOWED($v$,$G$)
Initiate: $A = \emptyset$.
Set $v$ as the source variable in $G$.
for $\{u, w\} \in E(G)$ do
    if $d_G(v, u) < d_G(v, w)$ then $A = A \cup (u, w)$ end if
    if $d_G(v, u) > d_G(v, w)$ then $A = A \cup (w, u)$ end if
end for
return $A$.
end function

function W($v$,$G$)
$\overline{F}$ = Undirected version of elements of FLOWED$(v, G)$.
$G' = G\backslash\overline{F}$.
Remove isolated vertices from $G'$.
return $\prod_{\hat{G}\in comp(G')}\sum_{u\in V(\hat{G})}$ W$(u, \hat{G})$.
end function

function RANDEDGE($G$,$A$)
Set $v^* \in V(G)$ as the source variable of $G$ with
    probability W$(v^*, G)/\sum_{v\in V(G)}$ W$(v, G)$.
$A = A \cup$ FLOWED$(v^*, G)$.
$\overline{F}$ = Undirected version of elements of FLOWED$(v^*, G)$.
$G' = G\backslash\overline{F}$.
Remove isolated vertices from $G'$.
$A = A\bigcup_{\hat{G}\in comp(G')}$ RANDEDGE$(\hat{G}, A)$.
return $A$.
end function
```

Next we consider the required cardinality of the set $\mathcal{G}'$ to obtain a desired accuracy in estimating $\mathcal{D}(\mathcal{I})$. We use Chernoff bound for this purpose.

Proposition 1 (Chernoff Bound). *Let $X_1, ..., X_N$ be independent random variables such that for all i, $0 \leq X_i \leq 1$. Let $\mu = \mathbb{E}[\sum_{i=1}^N X_i]$. Then*

$$P(|\sum_{i=1}^N X_i - \mu| \geq \epsilon\mu) \leq 2\exp(-\frac{\epsilon^2}{2+\epsilon}\mu).$$

Theorem 2. *For an estimator with $\mathbb{E}[D(\mathcal{I}, G'_i)] = \mathcal{D}(\mathcal{I})$, given set $\mathcal{I}$ and $\epsilon, \delta > 0$, if $N = |\mathcal{G}'| > \frac{|E(Ess(G^*))|(2+\epsilon)}{\epsilon^2}\ln(\frac{2}{\delta})$, then*

$$\mathcal{D}(\mathcal{I})(1-\epsilon) < \hat{\mathcal{D}}(\mathcal{I}) < \mathcal{D}(\mathcal{I})(1+\epsilon),$$

with probability larger than $1 - \delta$.

[5] A v-structure is a structure containing two converging directed edges whose tails are not connected by an edge.

[6] A source vertex has incoming degree equal to zero.

Proof. Define $X_i = \frac{D(\mathcal{I},G'_i)}{|E(Ess(G^*))|}$, for $i \in \{1,...,N\}$. By the assumption of the theorem, $\mathbb{E}[X_i] = \frac{1}{|E(Ess(G^*))|}\mathcal{D}(\mathcal{I})$ (Note that as stated before, this assumption is satisfied by Subroutine 1). Using Chernoff bound we have

$$\begin{aligned} P(|\sum_{i=1}^{N} X_i - \frac{N}{|E(Ess(G^*))|}\mathcal{D}(\mathcal{I})| \geq \epsilon \frac{N}{|E(Ess(G^*))|}\mathcal{D}(\mathcal{I})) \\ \leq 2\exp(-\frac{N\epsilon^2}{|E(Ess(G^*))|(2+\epsilon)}\mathcal{D}(\mathcal{I})) \\ \leq 2\exp(-\frac{N\epsilon^2}{|E(Ess(G^*))|(2+\epsilon)}). \end{aligned}$$

Therefore, $P(|\frac{1}{N}\sum_{i=1}^{N} D(\mathcal{I},G'_i) - \mathcal{D}(S)| \geq \epsilon\mathcal{D}(\mathcal{I})) \leq 2\exp(-\frac{N\epsilon^2}{|E(Ess(G^*))|(2+\epsilon)})$. Hence, $P(|\hat{\mathcal{D}}(\mathcal{I}) - \mathcal{D}(\mathcal{I})| < \epsilon\mathcal{D}(\mathcal{I})) > 1 - 2\exp(-\frac{N\epsilon^2}{|E(Ess(G^*))|(2+\epsilon)})$.
Setting $N > \frac{|E(Ess(G^*))|(2+\epsilon)}{\epsilon^2}\ln(\frac{2}{\delta})$, upper bounds the right hand side with $1-\delta$ and concludes the desired result. □

For any $\epsilon' > 0$, General Greedy Algorithm provides us with a $(1 - \frac{1}{e} - \epsilon')$-approximation of the optimal value as formalized in the following theorem.

Theorem 3. *For any $\epsilon', \delta' > 0$, there exists $\epsilon, \delta > 0$, such that if for any set $\mathcal{I}$, $\mathcal{D}(\mathcal{I})(1-\epsilon) < \hat{\mathcal{D}}(\mathcal{I}) < \mathcal{D}(\mathcal{I})(1+\epsilon)$ with probability larger than $1-\delta$, then Algorithm 1 is a $(1-\frac{1}{e}-\epsilon')$-approximation algorithm with probability larger than $1-\delta'$.*

See the supplementary materials for the proof.

Theorem 4. *The computational complexity of General Greedy algorithm is $O(kNn^{\Delta+1})$ where n and Δ are the order and maximum degree of $Ess(G^*)$, respectively.*

See the supplementary materials for the proof.

3.1. A Fast $\mathcal{D}(\mathcal{I})$ Estimator

The computational complexity of Subroutine 1 is $O(Nn^{\Delta})$, which may be intractable when the upper bound on the degree of the input graph is large. Therefore, we propose a fast and efficient estimator for $\mathcal{D}(\mathcal{I})$, better suited for graphs with large degree. Although this estimator is not unbiased, our extensive experimental results confirm that the sampling distribution of the sampler used in this estimator is very close to uniform.

The pseudo-code of the proposed estimator is presented in Subroutine 2. In this subroutine for the given mixed graph $Ess(G^*)$, we generate N DAGs from the Markov equivalence class of G^* as follows: We consider all subsets of size 3 from V in a uniformly random order (achieved by uniformly shuffling the labels of elements of V). For each subset $\{v_i, v_j, v_k\}$, we orient the undirected edges among $\{v_i, v_j, v_k\}$ independently according to Bernoulli$(\frac{1}{2})$ distribution. If the resulting orientation on the induced subgraph on $\{v_i, v_j, v_k\}$ is a directed cycle or a new v-structure which was not in $Ess(G^*)$, we redo the orienting. We keep checking all the subsets of size 3 until the induced subgraph on all of them are directed and none of them is a new v-structure, which did not exist in $Ess(G^*)$, or a directed cycle.

Subroutine 2 Fast $\mathcal{D}(\mathcal{I})$ Estimator Subroutine

```
Input: $Ess(G^*)$, intervention target set $\mathcal{I}$, and $N$.
initiate: $\mathcal{G}' = \emptyset$
for $i = 1$ to $N$, generate $G'_i$ as follows: do
    Uniformly shuffle the order of the elements of $V$.
    while the induced subgraph on any subset of size 3 of the variables is not directed, or a directed cycle, or a v-structure which was not in $Ess(G^*)$ do
        for all $\{v_i, v_j, v_k\} \subseteq V$ do
            Orient the undirected edges among $\{v_i, v_j, v_k\}$ independently according to $Bern(\frac{1}{2})$ until it becomes a directed structure which is not a directed cycle or a v-structure which was not in $Ess(G^*)$.
        end for
    end while
    $\mathcal{G}' = \mathcal{G}' \uplus G'_i$
end for
Output: $\hat{\mathcal{D}}(\mathcal{I}) = \frac{1}{|\mathcal{G}'|}\sum_{G'_i \in \mathcal{G}'} D(\mathcal{I}, G'_i)$
```

Lemma 3. *Each generated directed graph G'_i in Fast $\mathcal{D}(\mathcal{I})$ Estimator Subroutine belongs to the Markov equivalence class of G^*.*

See the supplementary materials for the proof.

We add the generated DAG to a multiset $\mathcal{G}'$. Finally, we calculate the estimated value $\hat{\mathcal{D}}(\mathcal{I})$ on $\mathcal{G}'$ instead of $\mathcal{G}$ as $\hat{\mathcal{D}}(\mathcal{I}) = \frac{1}{|\mathcal{G}'|}\sum_{G'_i \in \mathcal{G}'} D(\mathcal{I}, G'_i)$.

4. Improved Greedy Algorithm

We exploit the submodularity of function $\mathcal{D}$ to implement an accelerated variant of the general greedy algorithm through *lazy* evaluations, originally proposed by Minoux[7] (Minoux, 1978). In each round of the general greedy algorithm, we check the marginal gain $\Delta_v(\mathcal{I})$ for all remaining vertices in $V \backslash \mathcal{I}$. Note that as a consequence of submodularity of function $\mathcal{D}$, the set function Δ_v is monotonically decreasing. The main idea of the improved greedy algorithm is to take advantage of this property to avoid checking all the variables in each round of the algorithm. More specifically, suppose for vertices v_1 and v_2, in the i-th round of the algorithm we have obtained marginal gains $\Delta_{v_1}(\mathcal{I}_i) > \Delta_{v_2}(\mathcal{I}_i)$. If in the $(i+1)$-th round, we calculate $\Delta_{v_1}(\mathcal{I}_{i+1})$ and observe that $\Delta_{v_1}(\mathcal{I}_{i+1}) > \Delta_{v_2}(\mathcal{I}_i)$, from monotonic decreasing property of function Δ_v, we can conclude that

[7] There are improved versions of this algorithm in the literature (Mirzasoleiman et al., 2015).

$\Delta_{v_1}(\mathcal{I}_{i+1}) > \Delta_{v_2}(\mathcal{I}_{i+1})$, and hence, there is no need to calculate $\Delta_{v_2}(\mathcal{I}_{i+1})$.

Improved Greedy Algorithm is presented in Algorithm 2. The idea can be formalized as follows: We define a profit parameter p_v for each variable v and initialize the value for all variables with ∞. Moreover, we define an update flag *upd*$_v$ for all variables, which will be set to false at the beginning of every round of the algorithm and will be switched to true if we update p_v with the value of the marginal gain of vertex v. In each round, the algorithm picks vertex $v \in V \backslash \mathcal{I}$ with the largest profit, updates its profit with the value of the marginal gain of v, and sets *upd*$_v$ to true. This process is repeated until the vertex with the largest profit is already updated, i.e., its update flag is true. Then we add this vertex to $\mathcal{I}$ and end the round. For example, if in a round, the vertex v has the highest profit and after updating the profit of this vertex, p_v is still larger than all the other profits, we do not need to evaluate the marginal gain of any other vertex and we add v to $\mathcal{I}$.

The correctness of the Improved Greedy Algorithm follows directly from submodularity of function $\mathcal{D}$. Theorem 3 holds for Algorithm 2 as well, that is, for any $\epsilon' > 0$, Improved Greedy Algorithm provides us with a $(1 - \frac{1}{e} - \epsilon')$-approximation of the optimal value. This algorithm can lead to orders of magnitude performance speedup, as shown in (Leskovec et al., 2007).

Algorithm 2 Improved Greedy Algorithm

```
Input: Joint distribution over $V$, and budget $k$.
Obtain $Ess(G^*)$ by performing a CCI algorithm.
initiate: $\mathcal{I}_0 = \emptyset$, and $p_v = \infty, \forall v \in V$.
for $i = 1$ to $k$ do
  $upd_v = \texttt{false}, \forall v \in V \backslash \mathcal{I}_{i-1}$
  while true do
    $v^* = \arg\max_{v \in V \backslash \mathcal{I}_{i-1}} p_v$
    if $upd_{v^*}$ then
      $\mathcal{I}_i = \mathcal{I}_{i-1} \cup \{v^*\}$
      break;
    else
      $p_{v^*} = \hat{\mathcal{D}}(\mathcal{I}_{i-1} \cup \{v^*\}) - \hat{\mathcal{D}}(\mathcal{I}_{i-1})$
      $upd_{v^*} = \texttt{true}$
    end if
  end while
end for
Output: $\hat{\mathcal{I}} = \mathcal{I}_k$
```

5. Experimental Results

5.1. Synthetic Graphs

In this subsection, we evaluate the performance of Ran-GrID approach on synthetically generated chordal graphs[8]. Subroutine 1 and Algorithm 2 are used in our experiments. We use randomly chosen perfect elimination ordering (PEO)[9] of the vertices to generate our underlying chordal graphs (Hauser & Bühlmann, 2014; Shanmugam et al., 2015). For each graph, we pick a random ordering of the vertices. Starting from the vertex v with the highest order, we connect all the vertices with lower order to v with probability inversely proportional to the order of v. Then, we connect all the parents of v with directed edges, where each directed edge is oriented from the parent with the lower order to the parent with the higher order. In order to make sure that the generated graph will be connected, if vertex v is not connected to any of the vertices with the lower order, we pick one of them uniformly at random and set it as the parent of v.

To evaluate the performance of the proposed algorithm, for any underlying graph, we consider the ratio of the number of edges whose directions are discovered merely as a result of interventions to the number of edges whose directions were not resolved from the observational data. Note that due to our specific graph generating approach, the orientation of none of the edges is learned from the observational data. We compared Ran-GrID with two naive approaches: 1. Rand: Selecting intervened variables randomly, 2. MaxDeg: Sorting the list of variables based on the number of undirected edges connected to them in descending order and picking the first k variables from the sorted list.

We generated 100 instances of chordal DAGs of order 20. Figure 1(a) depicts the discovered edge ratio with respect to the budget k. As seen in this figure, three interventions suffices to discover the direction of more than 90% of the edges whose direction was unknown prior to performing interventions. Further, to investigate the effect of the order of the graph on the performance of the proposed algorithm and two naive approaches, we evaluated the discovered edge ratio for budget $k = 3$ on graphs with order $n \in \{10, 15, 20, 25, 30\}$ in Figure 1(b). As it can be seen in the figure, the discovered edge ratio for the proposed approach is greater than 91% for all $n \leq 30$. The performance of Rand approach degrades dramatically as n increases. Moreover, MaxDeg approach has even lower performance than Rand approach. We also studied the effect of graph density on the performance of proposed algorithm. Let r be the ratio of average number of edges to $\binom{n}{2}$. The discovered edge ratio for chordal DAGs of order 20 versus budget for different densities is depicted in Figure 1(c).

Furthermore, to compare the performance of the proposed algorithm with the optimal solution, we generated 100 in-

[8] A chord of a cycle is an edge not in the cycle whose endpoints are in the cycle. A hole in a graph is a cycle of length at least 4 having no chord. A graph is chordal if it has no hole.

[9] A perfect elimination ordering $\{v_1, v_2, ..., v_n\}$ on the vertices of an undirected chordal graph is such that for all i, the induced neighborhood of v_i on the subgraph formed by $\{v_1, v_2, ..., v_{i-1}\}$ is a clique.

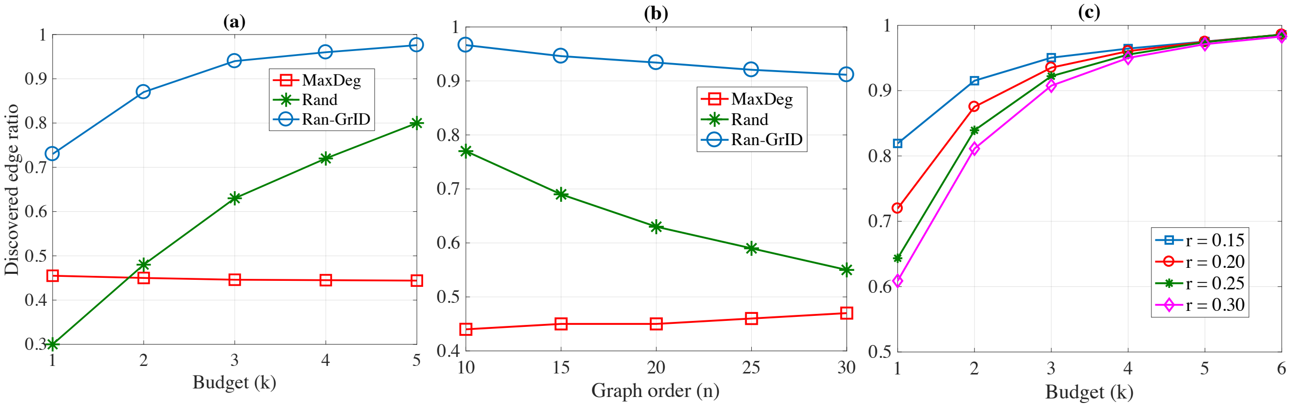

Figure 1. Discovered edge ratio versus (a) budget for $n = 20$, (b) graph orders for $k = 3$, (c) budget for $n = 20$ and different densities.

stances of chordal DAGs of order 10 and performed a brute force search to find the optimal solution of (2) for budget $k = 2$. The discovered edge ratio was 0.9 and 0.916 for our proposed algorithm and the optimal solution, respectively. For the aforementioned setting, the running time of the proposed approach on a machine with Intel Core i7 processor and 16 GB of RAM was 216 seconds while the one of the brute force approach was greater than 6000 seconds.

5.2. Real Graphs

We evaluated the performance of the proposed Improved Greedy Algorithm in gene regulatory networks (GRN). GRN is a collection of biological regulators that interact with each other. In GRN, the transcription factors are the main players to activate genes. The interactions between transcription factors and regulated genes in a species genome can be presented by a directed graph. In this graph, links are drawn whenever a transcription factor regulates a gene's expression. Moreover, some of vertices have both functions, i.e., are both transcription factor and regulated gene.

We considered GRNs in "DREAM 3 In Silico Network" challenge, conducted in 2008 (Marbach et al., 2009). The networks in this challenge were extracted from known biological interaction networks. Since we know the true causal structures in these GRNs, we can obtain $Ess(G^*)$ and give it as an input to the proposed algorithm. Figure 2 depicts the discovered edge ratio in five networks extracted from GRNs of E-coli and Yeast bacteria with budget $k = 5$. The order of each network is 100. As it can be seen, the discovered edge ratio is at least 0.65 in all GRNs.

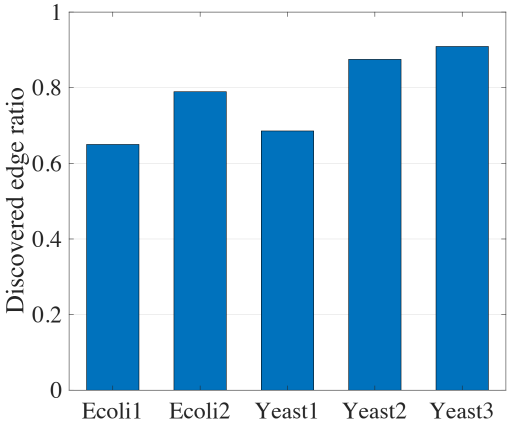

Figure 2. Discovered edge ratio in five GRNs from DREAM 3 challenge.

6. Conclusion

We studied the problem of experiment design for causal structure learning when only a limited number of experiments are available. In our model, each experiment consists of intervening on a single vertex. Also, experiments are designed merely based on the result of an initial purely observational test, which enables the experimenter to perform the interventional tests in parallel. We addressed the following question: "How much of the causal structure can be learned when only a limited number of experiments are available?" We formulated the problem of finding the best intervention target set as an optimization problem which aims to maximize the average number of edges whose directions are discovered. We introduce, for the first time, to use submoular optimization in the context of causal experimental design by showing that the objective function is monotonically increasing and submodular. Consequently, the greedy algorithm is a $(1 - \frac{1}{e})$-approximation algorithm for this problem. Moreover, we proposed estimation methods in order to compute the objective function for a given set of intervention targets. We further presented an accelerated variant of the greedy algorithm, which can achieve orders of magnitude performance speedup. We evaluated our proposed improved greedy algorithm on synthetic as well as real graphs. The results showed that a significant portion of the causal systems can be learned by only a few number of interventions.

Acknowledgements

Ghassami, Salehkaleybar, and Kiyavash's work was in part supported by Navy grant N00014-16-1-2804, and Army grant W911NF-15-1-0281. Bareinboim's work was in part supported by grants from NSF IIS-1704352 and IIS-1750807 (CAREER).

References

Andersson, Steen A, Madigan, David, Perlman, Michael D, et al. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

Bareinboim, Elias and Pearl, Judea. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

Chen, Wei, Wang, Yajun, and Yang, Siyu. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 199–208. ACM, 2009.

Chickering, David Maxwell. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Cooper, Gregory F and Yoo, Changwon. Causal discovery from a mixture of experimental and observational data. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 116–125. Morgan Kaufmann Publishers Inc., 1999.

Eberhardt, Frederick. Causation and intervention. *Unpublished doctoral dissertation, Carnegie Mellon University*, 2007.

Eberhardt, Frederick, Glymour, Clark, and Scheines, Richard. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. pp. 178–184, 2005.

Ghassami, AmirEmad, Salehkaleybar, Saber, Kiyavash, Negar, and Zhang, Kun. Learning causal structures using regression invariance. In *Advances in Neural Information Processing Systems*, pp. 3015–3025, 2017.

Ghassami, AmirEmad, Salehkaleybar, Saber, and Kiyavash, Negar. Counting and uniform sampling from markov equivalent dags. *arXiv preprint arXiv:1802.01239*, 2018.

Hauser, Alain and Bühlmann, Peter. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(Aug):2409–2464, 2012.

Hauser, Alain and Bühlmann, Peter. Two optimal strategies for active learning of causal models from interventional data. *International Journal of Approximate Reasoning*, 55(4):926–939, 2014.

He, Yang-Bo and Geng, Zhi. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov): 2523–2547, 2008.

He, Yangbo, Jia, Jinzhu, and Yu, Bin. Counting and exploring sizes of markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 16(1):2589–2609, 2015.

Hoyer, Patrik O, Janzing, Dominik, Mooij, Joris M, Peters, Jonas, and Schölkopf, Bernhard. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pp. 689–696, 2009.

Hyttinen, Antti, Eberhardt, Frederick, and Hoyer, Patrik O. Experiment selection for causal discovery. *Journal of Machine Learning Research*, 14(1):3041–3071, 2013.

Kempe, David, Kleinberg, Jon, and Tardos, Éva. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146. ACM, 2003.

Kocaoglu, Murat, Dimakis, Alexandros G, and Vishwanath, Sriram. Cost-optimal learning of causal graphs. *arXiv preprint arXiv:1703.02645*, 2017a.

Kocaoglu, Murat, Shanmugam, Karthikeyan, and Bareinboim, Elias. Experimental design for learning causal graphs with latent variables. In *Advances in Neural Information Processing Systems*, pp. 7021–7031, 2017b.

Koller, Daphne and Friedman, Nir. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Langley, P. Crafting papers on machine learning. In Langley, Pat (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Leskovec, Jure, Krause, Andreas, Guestrin, Carlos, Faloutsos, Christos, VanBriesen, Jeanne, and Glance, Natalie. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 420–429. ACM, 2007.

Marbach, Daniel, Schaffter, Thomas, Mattiussi, Claudio, and Floreano, Dario. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of computational biology*, 16(2): 229–239, 2009.

Meek, Christopher. Graphical models: Selecting causal and statistical models. 1997.

Minoux, Michel. Accelerated greedy algorithms for maximizing submodular set functions. *Optimization Techniques*, pp. 234–243, 1978.

Mirzasoleiman, Baharan, Badanidiyuru, Ashwinkumar, Karbasi, Amin, Vondrák, Jan, and Krause, Andreas. Lazier than lazy greedy. In *AAAI*, pp. 1812–1818, 2015.

Mooij, Joris M, Peters, Jonas, Janzing, Dominik, Zscheischler, Jakob, and Schölkopf, Bernhard. Distinguishing cause from effect using observational data: methods and benchmarks. *The Journal of Machine Learning Research*, 17(1):1103–1204, 2016.

Nemhauser, George L, Wolsey, Laurence A, and Fisher, Marshall L. An analysis of approximations for maximizing submodular set functions?i. *Mathematical Programming*, 14(1):265–294, 1978.

Pearl, Judea. *Causality*. Cambridge university press, 2009.

Peters, Jonas and Bühlmann, Peter. Identifiability of gaussian structural equation models with equal error variances. *arXiv preprint arXiv:1205.2536*, 2012.

Quinn, Christopher J, Kiyavash, Negar, and Coleman, Todd P. Directed information graphs. *IEEE Transactions on information theory*, 61(12):6887–6909, 2015.

Shanmugam, Karthikeyan, Kocaoglu, Murat, Dimakis, Alexandros G, and Vishwanath, Sriram. Learning causal graphs with small interventions. In *Advances in Neural Information Processing Systems*, pp. 3195–3203, 2015.

Shimizu, Shohei, Hoyer, Patrik O, Hyvärinen, Aapo, and Kerminen, Antti. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct):2003–2030, 2006.

Solus, Liam, Wang, Yuhao, Matejovicova, Lenka, and Uhler, Caroline. Consistency guarantees for permutation-based causal inference algorithms. *arXiv preprint arXiv:1702.03530*, 2017.

Spirtes, Peter, Glymour, Clark N, and Scheines, Richard. *Causation, prediction, and search*. MIT press, 2000.

Sun, Jie, Taylor, Dane, and Bollt, Erik M. Causal network inference by optimal causation entropy. *SIAM Journal on Applied Dynamical Systems*, 14(1):73–106, 2015.

Tian, Jin, He, Ru, and Ram, Lavanya. Bayesian model averaging using the k-best bayesian network structures. *arXiv preprint arXiv:1203.3520*, 2012.

Verma and Pearl, Judea. Equivalence and synthesis of causal models. In *Proceedings of UAI*, pp. 220–227, 1991.

Verma, Thomas and Pearl, Judea. An algorithm for deciding if a set of observed independencies has a causal explanation. In *Proceedings of UAI*, pp. 323–330, 1992.

Appendices

A. Example of Comparison with the Influence Maximization Problem

Suppose $k = 1$. Figure 3 depicts a graph for which the optimal solution to the influence maximization problem is different from the optimal solution to the budgeted experiment design problems. Clearly, influencing vertex v_1 leads to influencing all the vertices in the graph, and hence, this vertex is the solution to the influence maximization problem. But, intervening on v_1 leads to discovering the orientation of only 3 edges, while intervening on, say v_2, leads to discovering the orientation of 5 edges.

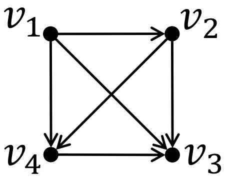

Figure 3. Example of comparison with the influence maximization problem.

B. Proof of Lemma 1

First we show that for a given directed graph $G_i \in \mathcal{G}$ the function $D(\mathcal{I}, G_i)$ is a monotonically increasing function of $\mathcal{I}$. In the proposed method, intervening on elements of $\mathcal{I}$, we first discover the orientation of the edges in $A_{G_i}^{(\mathcal{I})}$, and then applying the Meek rules, we possibly learn the orientation of some extra edges. Having $\mathcal{I}_1 \subseteq \mathcal{I}_2$ implies that $A_{G_i}^{(\mathcal{I}_1)} \subseteq A_{G_i}^{(\mathcal{I}_2)}$. Therefore using $\mathcal{I}_2$, we have more information about the direction of edges. Hence, in the step of applying Meek rules, by soundness and order-independence of Meek algorithm, we recover the direction of more extra edges, i.e., $R(A_{G_i}^{(\mathcal{I}_1)}, G_i) \subseteq R(A_{G_i}^{(\mathcal{I}_2)}, G_i)$, which in turn implies that $D(\mathcal{I}_1, G_i) \leq D(\mathcal{I}_2, G_i)$. Finally, from the relation $\mathcal{D}(\mathcal{I}) = \frac{1}{|\mathcal{G}|}\sum_{G_i \in \mathcal{G}} D(\mathcal{I}, G_i)$, the desired result is immediate.

C. Proof of Lemma 2

The direction $R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*) \subseteq R(A_{G^*}^{(\mathcal{I}_1 \cup \mathcal{I}_2)}, G^*)$ is proved in the proof of Lemma 1. Also, as observed in the proof of Lemma 1, we have $R(A_{G^*}^{(\mathcal{I}_1 \cup \mathcal{I}_2)}, G^*) \subseteq R(R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*), G^*)$. Therefore, in order to prove that $R(A_{G^*}^{(\mathcal{I}_1 \cup \mathcal{I}_2)}, G^*) \subseteq R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*)$, it suffices to show that $R(R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*), G^*) \subseteq R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*)$, for which it suffices to show that if $e \notin R(A_{G^*}^{(\mathcal{I}_1)}, G^*)$ and $e \notin R(A_{G^*}^{(\mathcal{I}_2)}, G^*)$, then $e \notin R(R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*), G^*)$.

Proof by contradiction. Let $e \notin R(A_{G^*}^{(\mathcal{I}_1)}, G^*)$ and $e \notin R(A_{G^*}^{(\mathcal{I}_2)}, G^*)$, but its orientation is learned in the first iteration of applying Meek rules to $R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*) \cup A(\mathit{Ess}(G^*))$. Then, we have learned the orientation of e due to one of Meek rules (Verma & Pearl, 1992):

- *Rule 1.* $e = \{a, b\}$ is oriented as (a, b) if $\exists c$ s.t. $e_1 = (c, a) \in R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*) \cup A(\mathit{Ess}(G^*))$, and $\{c, b\} \notin$ skeleton of G^*.
- *Rule 2.* $e = \{a, b\}$ is oriented as (a, b) if $\exists c$ s.t. $e_1 = (a, c) \in R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*) \cup A(\mathit{Ess}(G^*))$, and $e_2 = (c, b) \in R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*) \cup A(\mathit{Ess}(G^*))$.
- *Rule 3.* $e = \{a, b\}$ is oriented as (a, b) if $\exists c, d$ s.t. $e_1 = (c, b) \in R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*) \cup A(\mathit{Ess}(G^*))$, $e_2 = (d, b) \in R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*) \cup A(\mathit{Ess}(G^*))$, $\{a, c\} \in$ skeleton of G^*, $\{a, d\} \in$ skeleton of G^*, and $\{c, d\} \notin$ skeleton of G^*.
- *Rule 4.* $e = \{a, b\}$ is oriented as (a, b) and $e = \{b, c\}$ is oriented as (c, b) if $\exists d$ s.t. $e_1 = (d, c) \in R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup R(A_{G^*}^{(\mathcal{I}_2)}, G^*) \cup A(\mathit{Ess}(G^*))$, $\{a, c\} \in$ skeleton of G^*, $\{a, d\} \in$ skeleton of G^*, and $\{b, d\} \notin$ skeleton of G^*.

In what follows, we show that the orientation of e cannot be learned due to any of the Meek rules unless it belongs to $R(A_{G^*}^{(\mathcal{I}_1)}, G^*)$ or $R(A_{G^*}^{(\mathcal{I}_2)}, G^*)$.

Rule 1.

Without loss of generality, assume $e_1 \in R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup A(\mathit{Ess}(G^*))$. Therefore, we should have the condition of rule 1 satisfied when only intervening on $\mathcal{I}_1$ as well, which implies that $e \in R(A_{G^*}^{(\mathcal{I}_1)}, G^*)$, which is a contradiction.

Rule 2.

If both e_1 and e_2 belong to $R(A_{G^*}^{(\mathcal{I}_1)}, G^*) \cup A(\mathit{Ess}(G^*))$ (or $R(A_{G^*}^{(\mathcal{I}_2)}, G^*) \cup A(\mathit{Ess}(G^*))$), then we should have the condition of rule 2 satisfied when only intervening on $\mathcal{I}_1$ (or $\mathcal{I}_2$) as well, which implies that $e \in R(A_{G^*}^{(\mathcal{I}_1)}, G^*)$ (or $e \in R(A_{G^*}^{(\mathcal{I}_2)}, G^*)$), which is a contradiction. Therefore, it suffices to show that the case that e_1 belongs to exactly one of

$R(A^{(\mathcal{I}_1)}_{G^*}, G^*)\cup A(\mathit{Ess}(G^*))$ or $R(A^{(\mathcal{I}_2)}_{G^*}, G^*)\cup A(\mathit{Ess}(G^*))$ and e_2 belongs only to the other one, does not happen. To this end, it suffices to show that there does not exist intervention target set $\mathcal{I}$ such that $e_1 \in R(A^{(\mathcal{I})}_{G^*}, G^*) \cup A(\mathit{Ess}(G^*))$, and $e, e_2 \notin R(A^{(\mathcal{I})}_{G^*}, G^*) \cup A(\mathit{Ess}(G^*))$, i.e., there does not exist intervention target set $\mathcal{I}$ that has structure S_0, depicted in Figure 4, as a subgraph of $\mathit{Ess}(G^*)$ after applying the orientations learned from $R(A^{(\mathcal{I})}_{G^*}, G^*)$.

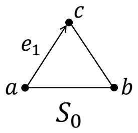

Figure 4. Structure S_0

If $e_1 \in A^{(\mathcal{I})}_{G^*}$, then $a \in \mathcal{I}$ or $c \in \mathcal{I}$, which implies $e \in A^{(\mathcal{I})}_{G^*}$ or $e_2 \in A^{(\mathcal{I})}_{G^*}$, respectively, and hence, $e \in R(A^{(\mathcal{I})}_{G^*}, G^*)$ or $e_2 \in R(A^{(\mathcal{I})}_{G^*}, G^*)$, respectively. Therefore, in either case, $e \in R(A^{(\mathcal{I})}_{G^*}, G^*)$, and S_0 will not be a subgraph. Therefore, $e_1 \notin A^{(\mathcal{I})}_{G^*}$, and hence, e_1 was learned by applying one of the Meek rules. We consider each or the rules in the following:

- If we have learned the orientation of e_1 from rule 1, then we should have had one of the structures in Figure 5 as a subgraph of $\mathit{Ess}(G^*)$ after applying the orientations learned from $R(A^{(\mathcal{I})}_{G^*}, G^*)$. In case of structure S_1, using rule 1 on subgraph induced on vertices $\{v_1, a, b\}$, we will also learn (a, b). In case of structure S_2, using rule 4, we will also learn (b, c). Therefore, we cannot learn only the direction of e_1 and hence, S_0 will not be a subgraph.

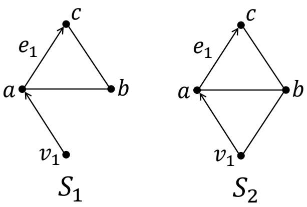

Figure 5. Rule 1

- If we have learned the orientation of e_1 from rule 3, then we have had one of the structures in Figure 6 as a subgraph of $\mathit{Ess}(G^*)$ after applying the orientations learned from $R(A^{(\mathcal{I})}_{G^*}, G^*)$. In case of structures S_3 and S_4, using rule 1 on subgraph induced on vertices $\{v_2, c, b\}$, we will also learn (c, b). In case of structure S_5, using rule 3 on subgraph induced on vertices $\{b, v_2, c, v_1\}$, we will also learn (b, c). Therefore, we cannot learn only the direction of e_1 and hence, S_0 will not be a subgraph.

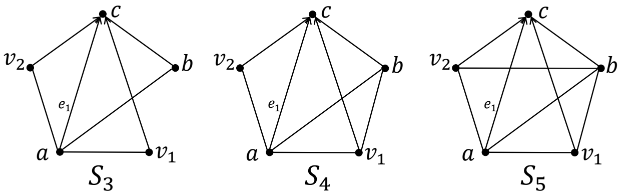

Figure 6. Rule 3

- If we have learned the orientation of e_1 from rule 4, then we have had one of the structures in Figure 7 as a subgraph of $\mathit{Ess}(G^*)$ after applying the orientations learned from $R(A^{(\mathcal{I})}_{G^*}, G^*)$. In case of structures S_6, using rule 1 on subgraph induced on vertices $\{v_1, c, b\}$, we will also learn (c, b). In case of structure S_7, using rule 1 on subgraph induced on vertices $\{v_2, v_1, b\}$, we will also learn (v_1, b), and then using rule 4 on subgraph induced on vertices $\{b, a, v_2, v_1\}$, we will also learn (a, b). In case of structures S_8, using rule 4 on subgraph induced on vertices $\{b, v_2, v_1, c\}$, we will also learn (b, c). Therefore, we cannot learn only the direction of e_1 and hence, S_0 will not be a subgraph.

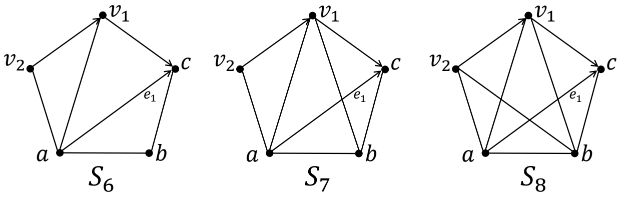

Figure 7. Rule 4

- If we have learned the orientation of e_1 from rule 2, then we should have had one of the structures in Figure 8 as a subgraph of $\mathit{Ess}(G^*)$ after applying the orientations learned from $R(A^{(\mathcal{I})}_{G^*}, G^*)$. In case of structure S_9, using rule 1 on subgraph induced on vertices $\{v_1, c, b\}$, we will also learn (c, b) and hence, S_0 will not be a subgraph. In case of structure S_{10}, if $v_1 \in \mathcal{I}$, then the direction of the edge $\{v_1, b\}$ will be also known. If the direction of this edge is (v_1, b), then using rule 2 on subgraph induced on vertices $\{a, v_1, b\}$, we will also learn (a, b); otherwise, using rule 2 on subgraph induced on vertices $\{b, v_1, c\}$, we will also learn (c, b). Therefore, $v_1 \notin \mathcal{I}$. Also, as mentioned earlier, $a \notin \mathcal{I}$. Therefore, we have learned the orientation of (a, v_1) from applying Meek rules.

  In the triangle induced on vertices $\{v_1, b, a\}$, we have learned only the orientation of one edge, which is (a, v_1). But as seen in structures S_1 to S_9, all of them lead to learning the orientation of at least 2 edges of a triangle. In the following, we will show that a structure of form S_{10}, does not lead to learning the orientation of only (a, v_1) and making S_{10} a subgraph either.

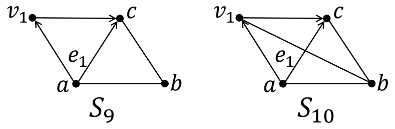

Figure 8. Rule 2

Suppose we had learned (a, v_1) via a structure of form S_{10}, as depicted in Figure 9(a). Using rule 4 on subgraph induced on vertices $\{v_2, v_1, c, b\}$, we will also learn (b, c). Therefore, we should have the edge $\{v_2, c\}$ too. Also, using rule 2 on triangle induced on vertices $\{v_2, v_1, c\}$, the orientation of this edges should be (v_2, c). Therefore, in order to have S_{10} as a subgraph, we need to have the structure depicted in Figure 9(b) as a subgraph. As seen in Figure 9(b), we again have a structure similar to S_{10}: a complete skeleton K_5, which contains (v_j, c), (a, v_j), $\{v_j, b\}$, for $j \in \{1, 2\}$ and (v_2, v_1), with a triangle on vertices $\{v_2, b, a\}$, in which we have learned only the orientation of (a, v_2).

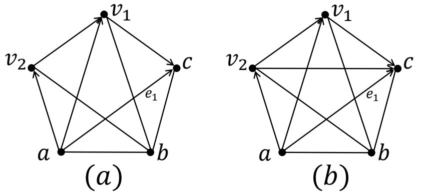

Figure 9. Step of the induction.

We claim that this procedure always repeats, i.e., at step i, we end up with skeleton K_i, which contains (v_j, c), (a, v_j), $\{v_j, b\}$, for $j \in \{1, ..., i\}$ and (v_k, v_j), for $1 \leq j < k \leq i$, with a triangle induced on vertices $\{v_i, b, a\}$, in which we have learned only the orientation of (a, v_i). We prove this claim by induction. We have already proved the base of the induction above. For the step of the induction, suppose the hypothesis is true for $i - 1$. Add vertex v_i to form a structure of form S_{10} for (a, v_{i-1}). v_i should be adjacent to v_j, for $j \in \{1, ..., i-2\}$; otherwise, using rule 4 on subgraph induced on vertices $\{v_i, v_{i-1}, v_j, b\}$, we will also learn (b, v_j). Moreover, using rule 2 on triangle induced on vertices $\{v_i, v_{i-1}, v_j\}$, the direction of $\{v_i, v_j\}$ should be (v_i, v_j). Also, using rule 4 on subgraph induced on vertices $\{v_i, v_{i-1}, c, b\}$, we will also learn (b, c). Therefore, we should have the edge $\{v_i, c\}$ too.

We showed that S_0 is a subgraph only if S_{10} is a subgraph, and S_{10} is a subgraph only if the structure in Figure 9(b) is a subgraph, and this chain of required subgraphs continue. Therefore, since the order of the graph is finite, there exist a step where since we cannot add a new vertex, it is not possible to have one of the required subgraphs, and hence we conclude that S_0 is not a subgraph.

Rule 3.

Since edges e_1 and e_2 form a v-structure, they should appear in $A(\mathit{Ess}(G^*))$ as well. Therefore, we should have the condition of rule 3 satisfied when only intervening on $\mathcal{I}_1$ as well, which implies that $e \in R(A^{(\mathcal{I}_1)}_{G^*}, G^*)$, which is a contradiction.

Rule 4.

Without loss of generality, assume $e_1 \in R(A^{(\mathcal{I}_1)}_{G^*}, G^*) \cup A(\mathit{Ess}(G^*))$. Therefore, we should have the condition of rule 4 satisfied when only intervening on $\mathcal{I}_1$ as well, which implies that $e \in R(A^{(\mathcal{I}_1)}_{G^*}, G^*)$, which is a contradiction.

The argument above proves that there is no edge e such that $e \notin R(A^{(\mathcal{I}_1)}_{G^*}, G^*)$ and $e \notin R(A^{(\mathcal{I}_2)}_{G^*}, G^*)$, but $e \in R(R(A^{(\mathcal{I}_1)}_{G^*}, G^*) \cup R(A^{(\mathcal{I}_2)}_{G^*}, G^*), G^*)$.

D. Proof of Theorem 3

Let $\mathcal{I}^* = \{v_1^*, ..., v_k^*\} \in \arg\max_{\mathcal{I}:\mathcal{I}\subseteq V, |\mathcal{I}|=k} \mathcal{D}(\mathcal{I})$. We have

$$\begin{aligned}\mathcal{D}(\mathcal{I}^*) &\overset{(a)}{\leq} \mathcal{D}(\mathcal{I}^* \cup \mathcal{I}_i) = \mathcal{D}(\mathcal{I}_i)\\ &+ \sum_{j=1}^{k}[\mathcal{D}(\mathcal{I}_i \cup \{v_1^*, ..., v_j^*\}) - \mathcal{D}(\mathcal{I}_i \cup \{v_1^*, ..., v_{j-1}^*\})]\\ &\overset{(b)}{\leq} \mathcal{D}(\mathcal{I}_i) + \sum_{j=1}^{k}[\mathcal{D}(\mathcal{I}_i \cup \{v_j^*\}) - \mathcal{D}(\mathcal{I}_i)],\end{aligned} \tag{3}$$

where (a) follows from Lemma 1, and (b) follows from Theorem 1. Define $\hat{\mathcal{D}}_{i,v,1}$ and $\hat{\mathcal{D}}_{i,v,2}$ as the first and second calls of subroutine in i-th step for variable vv, respectively. By the assumption of the theorem we have

$$\mathcal{D}(\mathcal{I}_i \cup \{v_j^*\}) - \epsilon\mathcal{D}(\mathcal{I}_i \cup \{v_j^*\}) < \hat{\mathcal{D}}_{i,v_j^*,1}(\mathcal{I}_i \cup \{v_j^*\}),$$

with probability larger than $1 - \delta$. Therefore

$$\mathcal{D}(\mathcal{I}_i \cup \{v_j^*\}) < \hat{\mathcal{D}}_{i,v_j^*,1}(\mathcal{I}_i \cup \{v_j^*\}) + \epsilon\mathcal{D}(\mathcal{I}^*),$$

with probability larger than $1 - \delta$. Similarly

$$\begin{aligned}&\hat{\mathcal{D}}_{i,v_j^*,2}(\mathcal{I}_i) < \mathcal{D}(\mathcal{I}_i) + \epsilon\mathcal{D}(\mathcal{I}_i) && w.p. > 1-\delta,\\ \Rightarrow\ &-\mathcal{D}(\mathcal{I}_i) < -\hat{\mathcal{D}}_{i,v_j^*,2}(\mathcal{I}_i) + \epsilon\mathcal{D}(\mathcal{I}^*) && w.p. > 1-\delta,\end{aligned}$$

Therefore,

$$\begin{aligned}\mathcal{D}(\mathcal{I}_i \cup \{v_j^*\}) - \mathcal{D}(\mathcal{I}_i) &< \hat{\mathcal{D}}_{i,v_j^*,1}(\mathcal{I}_i \cup \{v_j^*\}) \\ &- \hat{\mathcal{D}}_{i,v_j^*,2}(\mathcal{I}_i) + 2\epsilon\mathcal{D}(\mathcal{I}^*) \qquad w.p. > 1 - 2\delta.\end{aligned} \tag{4}$$

Also, by the definition of the greedy algorithm,

$$\begin{aligned}&\hat{\mathcal{D}}_{i,v_j^*,1}(\mathcal{I}_i \cup \{v_j^*\}) - \hat{\mathcal{D}}_{i,v_j^*,2}(\mathcal{I}_i) \\ &\quad \leq \hat{\mathcal{D}}_{i,v_{i+1},1}(\mathcal{I}_i \cup \{v_{i+1}\}) - \hat{\mathcal{D}}_{i,v_{i+1},2}(\mathcal{I}_i) \\ &\quad = \hat{\mathcal{D}}_{i,v_{i+1},1}(\mathcal{I}_{i+1}) - \hat{\mathcal{D}}_{i,v_{i+1},2}(\mathcal{I}_i),\end{aligned} \tag{5}$$

and similar to (4), we have

$$\begin{aligned}\hat{\mathcal{D}}_{i,v_{i+1},1}(\mathcal{I}_{i+1}) - \hat{\mathcal{D}}_{i,v_{i+1},2}(\mathcal{I}_i) &< \mathcal{D}(\mathcal{I}_{i+1}) \\ - \mathcal{D}(\mathcal{I}_i) + 2\epsilon\mathcal{D}(\mathcal{I}^*) &\qquad w.p. > 1 - 2\delta.\end{aligned} \tag{6}$$

Therefore, from equations (4), (5), and (6) we have

$$\mathcal{D}(\mathcal{I}_i \cup \{v_j^*\}) - \mathcal{D}(\mathcal{I}_i) < \mathcal{D}(\mathcal{I}_{i+1}) - \mathcal{D}(\mathcal{I}_i) + 4\epsilon\mathcal{D}(\mathcal{I}^*), \tag{7}$$

with probability larger than $1 - 4\delta$. Plugging (7) back in (3), we get

$$\begin{aligned}\mathcal{D}(\mathcal{I}^*) &< \mathcal{D}(\mathcal{I}_i) + \sum_{j=1}^{k} [\mathcal{D}(\mathcal{I}_{i+1}) - \mathcal{D}(\mathcal{I}_i) + 4\epsilon\mathcal{D}(\mathcal{I}^*)] \\ &= \mathcal{D}(\mathcal{I}_i) + k[\mathcal{D}(\mathcal{I}_{i+1}) - \mathcal{D}(\mathcal{I}_i)] + 4k\epsilon\mathcal{D}(\mathcal{I}^*),\end{aligned}$$

with probability larger than $1 - 4k\delta$. Therefore,

$$\begin{aligned}&\mathcal{D}(\mathcal{I}^*) - \mathcal{D}(\mathcal{I}_i) \\ &< k[\mathcal{D}(\mathcal{I}^*) - \mathcal{D}(\mathcal{I}_i)] - k[\mathcal{D}(\mathcal{I}^*) - \mathcal{D}(\mathcal{I}_{i+1})] + 4k\epsilon\mathcal{D}(\mathcal{I}^*),\end{aligned}$$

with probability larger than $1 - 4k\delta$. Defining $a_i := \mathcal{D}(\mathcal{I}^*) - \mathcal{D}(\mathcal{I}_i)$, and noting that $a_0 = \mathcal{D}(\mathcal{I}^*)$, by induction we have

$$\begin{aligned}a_k &= \mathcal{D}(\mathcal{I}^*) - \mathcal{D}(\mathcal{I}_k) \\ &< (1 - \frac{1}{k})^k \mathcal{D}(\mathcal{I}^*) + 4\epsilon\mathcal{D}(\mathcal{I}^*) \sum_{j=0}^{k-1} (1 - \frac{1}{k})^j \\ &< [\frac{1}{e} + 4\epsilon k]\mathcal{D}(\mathcal{I}^*) \qquad w.p. > 1 - 4k^2\delta.\end{aligned}$$

It concludes that

$$\mathcal{D}(\mathcal{I}_k) > (1 - \frac{1}{e} - 4\epsilon k)\mathcal{D}(\mathcal{I}^*) \qquad w.p. > 1 - 4k^2\delta.$$

Therefore, for $\epsilon = \frac{\epsilon'}{4k}$ and $\delta = \frac{\delta'}{4k^2}$, Algorithms 1 is a $(1 - \frac{1}{e} - \epsilon')$-approximation algorithm with probability larger than $1 - \delta'$.

E. Proof of Theorem 4

We run the algorithm for k iterations. In each iteration, we execute the function $\hat{\mathcal{D}}(.)$ using Subroutine 1 for at most n vertices. Furthermore, in this subroutine, we generate N random DAGs by calling the function RANDEDGE, where in (Ghassami et al., 2018) it is shown that the complexity of each call is $O(n^{\Delta})$. Hence, the computational complexity of the algorithm is $O(knN \times n^{\Delta})$.

F. Proof of Lemma 3

We require the following lemma for the proof:

Lemma 4. *A chordal graph has a directed cycle only if it has a directed cycle of size 3.*

Proof. If the directed cycle is of size 3 itself, the claim is trivial. Suppose the cycle C_n is of size $n > 3$. Relabel the vertices of C_n to have $C_n = (v_1, ..., v_n, v_1)$. Since the graph is chordal, C_n has a chord and hence we have a triangle on vertices $\{v_i, v_{i+1}, v_{i+2}\}$ for some i. If the direction of $\{v_i, v_{i+2}\}$ is (v_{i+2}, v_i), we have the directed cycle $(v_i, v_{i+1}, v_{i+2}, v_i)$ which is of size 3. Otherwise, we have the directed cycle $C_{n-1} = (v_1, ..., v_i, v_{i+2}, .., v_n, v_1)$ on $n-1$ vertices. Relabeling the vertices from 1 to $n-1$ and repeating the above reasoning concludes the lemma. □

Proof of Lemma 3. All the components in the undirected subgraph of $Ess(G^*)$ are chordal (Hauser & Bühlmann, 2012). Therefore, by Lemma 4, to insure that a generated directed graph is a DAG, it suffices to make sure that it does not have any directed cycles of length 3, which is one of the checks that we do in the proposed procedure. For checking if the generated DAG is in the same Markov equivalence class as G^*, it suffices to check if they have the same set of v-structures (Verma & Pearl, 1991), which is the other check that we do in the proposed procedure. □