Efficient Intervention Design for Causal Discovery with Latents

Raghavendra Addanki [1] **Shiva Prasad Kasiviswanathan** [2] **Andrew McGregor** [1] **Cameron Musco** [1]

Abstract

We consider recovering a causal graph in presence of latent variables, where we seek to minimize the cost of interventions used in the recovery process. We consider two intervention cost models: (1) a linear cost model where the cost of an intervention on a subset of variables has a linear form, and (2) an identity cost model where the cost of an intervention is the same, regardless of what variables it is on, i.e., the goal is just to minimize the number of interventions. Under the linear cost model, we give an algorithm to identify the ancestral relations of the underlying causal graph, achieving within a 2-factor of the optimal intervention cost. This approximation factor can be improved to $1+\epsilon$ for any $\epsilon > 0$ under some mild restrictions. Under the identity cost model, we bound the number of interventions needed to recover the entire causal graph, including the latent variables, using a parameterization of the causal graph through a special type of colliders. In particular, we introduce the notion of p-colliders, that are colliders between pair of nodes arising from a specific type of conditioning in the causal graph, and provide an upper bound on the number of interventions as a function of the maximum number of p-colliders between any two nodes in the causal graph.

1. Introduction

Causality has long been a key tool in studying and analyzing various behaviors in fields such as genetics, psychology, and economics [Pearl, 2009]. Causality also plays a pivotal role in helping us build systems that can understand the world around us, and in turn, in helping us understand the behavior of machine learning systems deployed in the real world. Although the theory of causality has been around for more than three decades, for these reasons it has received increasing attention in recent years. In this paper, we study one of the fundamental problems of causality: *causal discovery*. In causal discovery, we want to learn all the causal relations existing between variables (nodes of the causal graph) of our system. It has been shown that, under certain assumptions, observational data alone only lets us recover *the existence of a causal relationship* between two variables, but not the *direction* of all relationships. To recover the directions of causal edges, we use the notion of an *intervention* described in the Structural Causal Models (SCM) framework introduced by Pearl [2009].

An intervention requires us to fix a subset of variables to a set of values, inducing a new distribution on the free variables. Such a system manipulation is generally expensive and thus there has been significant interest in trying to minimize the number of interventions and their cost in causal discovery. In a general cost model, intervening on any subset of variables has a cost associated with it, and the goal is to identify all causal relationships and their directions while minimizing the total cost of interventions applied. This captures the fact that some interventions are more expensive than others. For example, in a medical study, intervening on certain variables might be impractical or unethical. In this work, we study two simplifications of this general cost model. In the *linear cost model*, each variable has an intervention cost, and the cost of an intervention on a subset of variables is the sum of costs for each variable in the set [Kocaoglu et al., 2017a; Lindgren et al., 2018]. In the *identity cost model*, every intervention has the same cost, regardless of what variables it contains and therefore minimizing the intervention cost is the same as minimizing the number of interventions [Kocaoglu et al., 2017b].

As is standard in the causality literature, we assume that our causal relationship graph satisfies the *causal Markov condition* and *faithfulness* [Spirtes et al., 2000b]. We assume that faithfulness holds both in the observational and interventional distributions following [Hauser & Bühlmann, 2014]. As is common, we also assume that we are given access to an oracle that can check if two variables are independent, conditioned on a subset of variables. We discuss

[1]College of Information and Computer Sciences, University of Massachusetts, Amherst, USA. [2]Amazon. Correspondence to: Raghavendra Addanki <raddanki@cs.umass.edu>, Shiva Prasad Kasiviswanathan <kasivisw@gmail.com>, Andrew McGregor <mcgregor@cs.umass.edu>, Cameron Musco <cmusco@cs.umass.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

this assumption in more detail in Section 2. Unlike much prior work, we *do not make* the causal sufficiency assumption: that there are no unobserved (or latent) variables in the system. Our algorithms apply to the causal discovery problem with the existence of latent variables.

Results. Our contributions are as follows. Let $\mathcal{G}$ be a causal graph on both observable variables V and latent variables L. A directed edge (u, v) in $\mathcal{G}$ indicates a causal relationship from u to v. Let G be the induced subgraph of $\mathcal{G}$ on the n observable variables (referred to as observable graph). See Section 2 for a more formal description.

Linear Cost Model: In the linear cost model, we give an algorithm that given $m = \Omega(\log n)$, outputs a set of m interventions that can be used to recover all ancestral relations of the observable graph G.[1] We show that cost of interventions generated by the algorithm is at most twice the cost of the optimum set of interventions for this task. Our result is based on a characterization that shows that generating a set of interventions sufficient to recover ancestral relations is equivalent to designing a *strongly separating set system* (Def. 2.2). We show how to design such a set system with at most twice the optimum cost based on a greedy algorithm that constructs intervention sets which includes a variable with high cost in the least number of sets possible.

In the special case when each variable has unit intervention cost [Hyttinen et al., 2013a] gives an exact algorithm to recover ancestral relations in G with minimal total cost. Their algorithm is based on the Kruskal-Katona theorem in combinatorics [Kruskal, 1963; Katona, 1966]. We show that a modification of this approach yields a $(1+\epsilon)$-approximation algorithm in the general linear cost model for any $0 < \epsilon \leq 1$, under mild assumptions on m and the maximum intervention cost on any one variable.

The linear cost model was first considered in [Kocaoglu et al., 2017a] and studied under the causal sufficiency (no latents) assumption. Lindgren et al. [2018] showed that under this assumption, which translates to the undirected component of the *Essential graph* of G being chordal, the problem of recovering G with optimal cost under the linear cost model is NP-hard. To the best of our knowledge, our result is the first to minimize intervention cost under the popular linear cost model in the presence of latents, and without the assumption of unit intervention cost on each variable.

We note that, while we give a 2-approximation for recovering ancestral relations in G, under the linear cost model, there seems to be no known characterization of the optimal intervention sets needed to recover the entire causal graph $\mathcal{G}$, making it hard to design a good approximation here. Tackling this problem in the linear cost model is an interesting direction for future work.

Identity Cost Model: In the identity cost model, where we seek to just minimize the number of interventions, recovering ancestral relations in G with minimum cost becomes trivial (see Section 4). Thus, in this case, we focus on algorithms that recover the causal graph $\mathcal{G}$ completely. We start with the notion of *colliders* in causal graphs [Pearl, 2009]. Our idea is to parameterize the causal graph in terms of a specific type of colliders that we refer to as p-colliders (Def. 4.2). Intuitively, a node v_k is p-collider for a pair of nodes (v_i, v_j) if a) it is a collider on a path between v_i and V_j and b) at least one of the parents v_i, v_j is a descendant of v_k. If the graph $\mathcal{G}$ has at most τ p-colliders between every pair of nodes, then our algorithm uses at most $O(n\tau \log n + n \log n)$ interventions. We also present causal graph instances where any non-adaptive algorithm requires $\Omega(n)$ interventions.

The only previous bound on recovering $\mathcal{G}$ in this setting utilized $O(\min\{d \log^2 n, \ell\} + d^2 \log n)$ interventions where d is the maximum (undirected) node degree and ℓ is the length of the longest directed path of the causal graph [Kocaoglu et al., 2017b]. Since we use a different parameterization of the causal graph, a direct comparison with this result is not always possible. We argue that a parameterization in terms of p-colliders is inherently more "natural" as it takes the directions of edges in $\mathcal{G}$ into account whereas the maximum degree does not. The presence of a single high-degree node can make the number of interventions required by existing work extremely high, even if the overall causal graph is sparse. In this case, the notion of p-colliders is a more global characterization of a causal graph. See Section 5 for a more detailed discussion of different parameter regimes under which our scheme provides a better bound. We also experimentally show that our scheme achieves a better bound over [Kocaoglu et al., 2017b] in some popular random graph models.

1.1. Other Related Work

Broadly, the problem of causal discovery has been studied under two different settings. In the first, one assumes *causal sufficiency*, i.e., that there are no unmeasured (latent) variables. Most work in this setting focuses on recovering causal relationships based on just observational data. Examples include algorithms like *IC* [Pearl, 2009] and *PC* [Spirtes et al., 2000a]. Much work has focused on understanding the limitations and assumptions underlying these algorithms [Hauser & Bühlmann, 2014; Hoyer et al., 2009; Heinze-Deml et al., 2018; Loh & Bühlmann, 2014; Hoyer et al., 2009; Shimizu et al., 2006]. It is well-known, that to disambiguate a causal graph from its equivalence class, interventional, rather than just observational data is

[1] As noted in Section 3, $m \geq \log n$ is a lower bound for any solution.

required [Hauser & Bühlmann, 2012; Eberhardt & Scheines, 2007; Eberhardt, 2007]. In particular, letting $\chi(\mathcal{G})$ be the chromatic number of G, $\Theta(\log \chi(\mathcal{G}))$ interventions are necessary and sufficient for recovery under the causal sufficiency assumption [Hauser & Bühlmann, 2014]. Surprising connections have been found [Hyttinen et al., 2013a; Katona, 1966; Mao-Cheng, 1984] between combinatorial structures and causality. Using these connections, much recent work has been devoted to minimizing the intervention cost while imposing constraints such as sparsity or different costs for different sets of nodes [Shanmugam et al., 2015; Kocaoglu et al., 2017a; Lindgren et al., 2018].

In many cases, causal sufficiency is too strong an assumption: it is often contested if the behavior of systems we observe can truly be attributed to measured variables [Pearl, 2000; Bareinboim & Pearl, 2016]. In light of this, many algorithms avoiding the causal sufficiency assumption, such as IC* [Verma & Pearl, 1992] and FCI [Spirtes et al., 2000b], have been developed. The above algorithms only use observational data. However, there is a growing interest in optimal intervention design in this setting [Silva et al., 2006; Hyttinen et al., 2013b; Parviainen & Koivisto, 2011]. We contribute to this line of work, focusing on minimizing the intervention cost required to recover the full intervention graph, or its ancestral graph, without causal sufficiency, i.e., in the presence of latents.

2. Preliminaries

Notation. Following the SCM framework introduced by Pearl [2009], we represent the set of random variables of interest by $V \cup L$ where V represents the set of endogenous (observed) variables that can be measured and L represents the set of exogenous (latent) variables that cannot be measured. We define a directed graph on these variables where an edge corresponds to a causal relation between the corresponding variables. The edges are directed with an edge (v_i, v_j) meaning that $v_i \to v_j$. As is common, we assume that all causal relations that exist between random variables in $V \cup L$ belong to one of the two categories : (i) $E \subseteq V \times V$ containing causal relations between the observed variables and (ii) $E_L \subseteq L \times V$ containing relations of the form $l \to v$ where $l \in L, v \in V$. Thus, the full edge set of our causal graph is denoted by $\mathcal{E} = E \cup E_L$. We also assume that every latent $l \in L$ influences exactly two observed variables i.e., $(l, u), (l, v) \in E_L$ and no other edges are incident on l. This assumption, also known as the *semi-Markovian* causal model, is well studied in many previous works [Shpitser & Pearl, 2006; Tian & Pearl, 2002; Kocaoglu et al., 2017b]. We let $\mathcal{G} = \mathcal{G}(V \cup L, \mathcal{E})$ denote the entire causal graph and refer to $G = G(V, E)$ as the *observable graph*. Let $V = \{v_1, \ldots, v_n\}$ and $|V| = n$.

Unless otherwise specified a path between two nodes is a undirected path. For every observable $v \in V$, let the parents of v be defined as $\mathrm{Pa}(v) = \{w \mid w \in V \text{ and } (w, v) \in E\}$. For a set of nodes $S \subseteq V$, $\mathrm{Pa}(S) = \cup_{v \in S} \mathrm{Pa}(v)$. If $v_i, v_j \in V$, we say v_j is a descendant of v_i (and v_i is an ancestor of v_j) if there is a directed path from v_i to v_j. $\mathrm{Anc}(v) = \{w \mid w \in V \text{ and } v \text{ is a descendant of } w\}$. We let $\mathrm{Anc}(G)$ denote the *ancestral graph*[2] of G where an edge $(v_i, v_j) \in \mathrm{Anc}(G)$ if and only if there is a directed path from v_i to v_j in G. One of our primary interests is in recovering $\mathrm{Anc}(G)$ using a minimal cost set of interventions.

Using Pearl's do-notation, we represent an intervention on a set of variables $S \subseteq V$ as $\mathrm{do}(S = s)$ for a value s in the domain of S and the joint probability distribution on $V \cup L$ conditioned on this intervention by $\Pr[\cdot \mid do(S)]$. We assume that there exists an oracle that answers queries such as *"Is v_i independent of v_j given Z in the interventional distribution* $\Pr[\cdot \mid \mathrm{do}(S = s)]$*?"*

Assumption 2.1 (Conditional Independence (CI)-Oracle)**.** *Given any $v_i, v_j \in V$ and $Z, S \subseteq V$ we have an oracle that tests whether $v_i \perp\!\!\!\perp v_j \mid Z, \mathrm{do}(S = s)$.*

Such conditional independence tests have been widely investigated with sublinear (in domain size) bounds on the sample size needed for implementing this oracle [Canonne et al., 2018; Zhang et al., 2011].

Intervention Cost Models. We study the causal discovery problem under two cost models:

Linear Cost Model. In this model, each node $v \in V$ has a different cost $c(v) \in \mathbb{R}^+$ and the cost of intervention on a set $S \subset V$ is defined as $\sum_{v \in S} c(v)$ (akin to [Lindgren et al., 2018]). That is, interventions that involve a larger number of, or more costly nodes, are more expensive. Our goal is to find an intervention set $\mathcal{S}$ minimizing $\sum_{S \in \mathcal{S}} \sum_{v \in S} c(v)$. We constrain the number of interventions to be upper bounded by some budget m. Without such a bound, we can observe that for ancestral graph recovery, the optimal intervention set is $\mathcal{S} = \{\{v_1\}, \{v_2\}, \ldots, \{v_n\}\}$ with cost $\sum_{v \in V} c(v)$ as intervention on every variable is necessary, as we need to account for the possibility of latent variables (See Lemma 3.1 for more details). The optimality of $\mathcal{S}$ here follows from a characterization of any feasible set system we establish in Lemma 3.1.

Identity Cost Model. As an intervention on a set of variables requires controlling the variables, and generating a new distribution, we want to use as few interventions as possible. In this cost model, an intervention on any set of observed variables has *unit cost* (no matter how many variables are in the set). We assume that for any intervention, querying the CI-oracle comes free of cost. This model is akin to the

[2] We note that the term *ancestral graph* has also been previously used in a different context, see e.g., [Richardson et al., 2002].

model studied in [Kocaoglu et al., 2017b].

Causal Discovery Goals. We will study two variations of the causal discovery problem. In the first, we aim to recover the ancestral graph $\mathrm{Anc}(G)$, which contains all the causal ancestral relationships between the observable variables V. In the second, our goal is to recover all the causal relations in $\mathcal{E}$, i.e., learn the entire causal graph $\mathcal{G}(V \cup L, \mathcal{E})$. We aim to perform both tasks using a set of intervention sets $\mathcal{S} = \{S_1, \ldots, S_m\}$ (each $S_i \subseteq V$) with minimal cost, with our cost models defined above.

For ancestral graph recovery, we will leverage a simple characterization of when a set of interventions $\mathcal{S} = \{S_1, \ldots, S_m\}$ is sufficient to recover $\mathrm{Anc}(G)$. In particular, $\mathcal{S}$ is sufficient if it is a *strongly separating set system* [Kocaoglu et al., 2017b].

Definition 2.2 (Strongly Separating Set System)**.** *A collection of subsets $\mathcal{S} = \{S_1, \cdots, S_m\}$ of the ground set V is a strongly separating set system if for every distinct $u, v \in V$ there exists S_i and S_j such that $u \in S_i \setminus S_j$ and $v \in S_j \setminus S_i$.*

Ancestral graph recovery using a strongly separating set system is simple: we intervene on each of the sets $S_1, \ldots, S_m$. Using CI-tests we can identify for every pair of v_i and v_j, if there is a path from v_i to v_j or not in G using the intervention corresponding to $S \in \mathcal{S}$ with $v_i \in S$ and $v_j \notin S$. We add an edge to $\mathrm{Anc}(G)$ if the test returns dependence. Finally, we take the transitive closure and output the resulting graph as $\mathrm{Anc}(G)$. In Lemma 3.1, we show that in fact being strongly separating is *necessary* for any set of interventions to be used to identify $\mathrm{Anc}(G)$.

3. Linear Cost Model

We begin with our results on recovering the ancestral graph $\mathrm{Anc}(G)$ in the linear cost model. Recall that, given a budget of m interventions, our objective is to find a set of interventions $\mathcal{S} = \{S_1, S_2, \cdots S_m\}$ that can be used to identify $\mathrm{Anc}(G)$ while minimizing $\sum_{S \in \mathcal{S}} \sum_{v \in S} c(v)$.

As detailed in Section 2, a strongly separating set system is sufficient to recover the ancestral graph. We show that it is also necessary: a set of interventions to discover $\mathrm{Anc}(G)$ must be a strongly separating set system (Definition 2.2). See proof in Appendix A.

Lemma 3.1. *Suppose $\mathcal{S} = \{S_1, S_2, \cdots, S_m\}$ is a collection of subsets of V. If $\mathcal{S}$ is not a strongly separating system, then there exists a causal graph G, for which $\mathrm{Anc}(G)$ is not identifiable using CI-tests under the interventions on $\mathcal{S}$.*

Given this characterization, the problem of constructing the ancestral graph $\mathrm{Anc}(G)$ with minimum linear cost *reduces* to that of constructing a strongly separating set system with minimum cost. In developing our algorithm for finding such a set system, it will be useful to represent a set system by a binary matrix, with rows corresponding to observable variables V and columns corresponding to interventions (sets $S_1, \ldots, S_m$).

Definition 3.2 (Strongly Separating Matrix)**.** *Matrix $U \in \{0,1\}^{n \times m}$ is a* strongly separating matrix *if $\forall i, j \in [n]$ there exists $k, k' \in [m]$ such that $U(i,k) = 1, U(j,k) = 0$ and $U(i,k') = 0, U(j,k') = 1$.*

Note that given a strongly separating set system $\mathcal{S}$, if we let U be the matrix where $U(i,k) = 1$ if $v_i \in S_k$ and 0 otherwise, U will be a strongly separating matrix. The other direction is also true. Let $U(j)$ denote the jth row of U. Using Definition 3.2 and above connection between recovering $\mathrm{Anc}(G)$ and strongly separating set system, we can reformulate the problem at hand as:

$$\min_U \sum_{j=1}^{n} c(v_j) \cdot \|U(j)\|_1 \quad (1)$$
$$\text{s.t. } U \in \{0,1\}^{n \times m} \text{ is a strongly separating matrix.}$$

We can thus view our problem as finding an assignment of vectors in $\{0,1\}^m$ (i.e., rows of U) to nodes in V that minimizes (1). Throughout, we will call $\|U(j)\|_1$ the *weight* of row $U(j)$, i.e., the number of 1s in that row. It is easy to see that $m \geq \log n$ is necessary for a feasible solution to exist as each row must be distinct.

We start by giving a 2-approximation algorithm for (1). In Section 3.2, we show how to obtain an improved approximation under certain assumptions.

3.1. 2-approximation Algorithm

In this section, we present an algorithm (Algorithm SSMatrix) that constructs a strongly separating matrix (and a corresponding intervention set) which minimizes (1) to within a 2-factor of the optimum. Missing details from section are collected in Appendix A.1.

Outline. Let U_{OPT} denote a strongly separating matrix minimizing (1). Let $c_{\mathrm{OPT}} = \sum_{j=1}^{n} c(v_j) \|U_{\mathrm{OPT}}(j)\|_1$ denote the objective value achieved by this optimum U_{OPT}. We start by relaxing the constraint on U so that it does *not need to be strongly separating*, but just must have unique rows, where none of the rows is all zero. In this case, we can optimize (1) very easily. We simply take the rows of U to be the n unique binary vectors in $\{0,1\}^m \setminus \{0^m\}$ with lowest weights. That is, m rows will have weight 1, $\binom{m}{2}$ will have weight 2, etc. We then assign the rows to the nodes in V in descending order of their costs. So the m nodes with the highest costs will be assigned the weight 1 rows, the next $\binom{m}{2}$ assigned weight 2 rows, etc. The cost of this assignment is only lower than c_{OPT}, as we have only relaxed the constraint in (1).

We next convert this relaxed solution into a valid strongly

separating matrix. Given $m + \log n$ columns, we can do this easily. Since there are n nodes, in the above assignment, all rows will have weight of at most $\log n$. Let $\bar{U} \in \{0,1\}^{m+\log n}$ have its first m columns equal to those of U. Additionally, use the last $\log n$ columns as 'row weight indicators': if $\|U(j)\|_1 = k$ then set $\bar{U}(j, m+k) = 1$. We can see that $\bar{U}$ is a strongly separating matrix. If two rows have different weights k, k' in $\bar{U}$, then the last $\log n$ columns ensure that they satisfy the strongly separating condition. If they have the same weight in $\bar{U}$, then they already satisfy the condition, as to be unique in U they must have at least 2 entries on which they differ.

To turn the above idea into a valid approximation algorithm that outputs $\bar{U}$ with just m (not $m + \log n$) columns, we argue that we can 'reserve' the last $\log n$ columns of $\bar{U}$ to serve as weight indicator columns. We are then left with just $m - \log n$ columns to work with. Thus we can only assign $m - \log n$ weight 1 rows, $\binom{m-\log n}{2}$ weight 2 rows, etc. Nevertheless, if $m \geq \gamma \log n$ (for a constant $\gamma > 1$), this does not affect the assignment much: for any i we can still 'cover' the $\binom{m}{i}$ weight i rows in U with rows of weight $\leq 2i$. Thus, after accounting for the weight indicator columns, each weight k row in U has weight $\leq 2k+1$ in $\bar{U}$. Overall, this gives us a 3-approximation algorithm: when k is 1 the weight of a row may become as large as 3.

To improve the approximation to a 2-approximation we *guess* the number of weight 1 vectors a_1 in the optimum solution U_{OPT} and assign the a_1 highest cost variables to weight 1 vectors, achieving optimal cost for these variables. There are $O(m)$ possible values for a_1 and so trying all guesses is still efficient. We then apply our approximation algorithm to the remaining $m - a_1$ available columns of U and $n - a_1$ variables. Since no variables are assigned weight 1 in this set, we achieve a tighter 2-approximation using our approach. The resulting matrix has the form:

$$U = \left(\begin{array}{c|c|c} \mathbb{I}_{a_1} & 0 & 0 \\ \hline 0 & C_1 & M_1 \\ \hline 0 & C_2 & M_2 \\ \hline \vdots & \vdots & \vdots \end{array}\right)$$

where $\mathbb{I}_{a_1}$ is the $a_1 \times a_1$ identity matrix, the rows of C_w are all weight w binary vectors of length $m - \log n - a_1$, and the rows of M_w are length $\log n$ binary vectors with 1's in the wth column. The entire approach is presented in Algorithm SSMATRIX and a proof of the approximation bound in Theorem 3.3 is present in Appendix A.1.

Theorem 3.3. *Let $m \geq \gamma \log n$ for constant $\gamma > 1$ and U be the strongly separating matrix returned by* SSMATRIX.[3] *Let $c_U = \sum_{j=1}^{n} c(v_j) \|U(j)\|_1$. Then, $c_U \leq 2 \cdot c_{\text{OPT}}$, where c_{OPT} is the objective value associated with optimum set of interventions corresponding to U_{OPT}.*

[3] In our proof, $\gamma = 66$ but this can likely be decreased.

Algorithm 1 SSMATRIX (V, m)

```
c_{U_min} ← ∞
for a_1 ∈ {0, 1, ··· , 2m/3} do
    U ∈ {0,1}^{n×m} be initialized with all zeros
    Assign the highest cost a_1 nodes with unit weight vectors such that U(i,i) = 1 for i ≤ a_1
    Set m' ← m − a_1
    Mark all vectors of weight at least 1 in {0,1}^{m'−log n} as available
    for unassigned v_i ∈ V (in decreasing order of cost) do
        Set U(i, (a_1 + 1) : m − log n) to smallest available weight vector in {0,1}^{m'−log n} and make this vector unavailable. Let the weight of the assigned vector be k
        Set 'row weight indicator' U(i, m' − log n + k) = 1
    end for
    Compute cost of objective for U be c_U
    if c_U < c_{U_min} then
        c_{U_min} ← c_U, U_min ← U
    end if
end for
Return U_min
```

Using the interventions from the matrix U returned by Algorithm SSMATRIX, we obtain a cost within twice the optimum for recovering $\text{Anc}(G)$.

3.2. $(1+\epsilon)$-approximation Algorithm

In [Hyttinen et al., 2013a], the authors show how to construct a collection $\mathcal{A}$ of m strongly separating intervention sets with minimum average set size, i.e., $\sum_{A \in \mathcal{A}} |A|/m$. This is equivalent to minimizing the objective (1) in the linear cost model when the cost of intervening on any node equals 1. In this section, we analyze an adaptation of their algorithm to the general linear cost model, and obtain a $(1+\epsilon)$-approximation for any given $0 < \epsilon \leq 1$, an improvement over the 2-approximation of Section 3.1. Our analysis requires mild restrictions on the number of interventions and an upper bound on the maximum cost. The algorithm will not depend on ϵ but these bounds will. Missing details from this section are collected in Appendix A.2.

Algorithm ϵ-SSMATRIX Outline. The famous Kruskal-Katona theorem in combinatorics forms the basis of the scheme presented in [Hyttinen et al., 2013a] for minimizing the average size of the intervention sets. To deal with with varying costs of node interventions, we augment this approach with a greedy strategy. Let $\mathcal{A}$ denote a set of m intervention sets over the nodes $\{v_1, v_2 \cdots, v_n\}$ obtained using the scheme from [Hyttinen et al., 2013a]. Construct a strongly separating matrix $\widetilde{U}$ from $\mathcal{A}$ with $\widetilde{U}(i,j) = 1$ iff $v_i \in A_j$ for $A_j \in \mathcal{A}$. Let ζ denote the ordering of rows of $\widetilde{U}$ in the increasing order of weight. Our Algorithm ϵ-SSMATRIX outputs the strongly separating matrix U where, for every $i \in [n]$, $U(i) = \widetilde{U}(\zeta(i))$ and the ith row of U corresponds to the node with ith largest cost.

Let $c_{max} = \max_{v_i \in V} c(v_i)/\min_{v_i \in V} c(v_i)$ be the ratio of maximum cost to minimum cost of nodes in V. For ease of analysis, we assume that the cost of any node is least 1.

Theorem 3.4. *Let U be the strongly separating matrix returned by ϵ-SSMATRIX. If $c_{max} \leq \frac{\epsilon n}{3\binom{m}{t}}$ for $0 < \epsilon \leq 1$ where $\binom{m}{k-1} < n \leq \binom{m}{k}$ and $t = \lfloor k - \epsilon k/3 \rfloor$, then,*

$$c_U := \sum_{j=1}^{n} c(v_j) \left\| U(j) \right\|_1 \leq (1+\epsilon) \cdot c_{\mathrm{OPT}} \ ,$$

where c_{OPT} is the objective value associated with optimum set of interventions corresponding to U_{OPT}.

Proof. Suppose the optimal solution U_{OPT} includes a_q^* vectors of weight q. Let S be the $a_1^* + a_2^* + \ldots + a_t^*$ nodes with highest cost in U_{OPT}. Since $a_q^* \leq \binom{m}{q}$, it immediately follows that $|S| \leq \sum_{i=q}^{t} \binom{m}{q}$. However, a slightly tighter analysis (see Lemma A.10) implies $|S| \leq \binom{m}{t}$. Let $c_{\mathrm{OPT}}(S)$ be the total contribution of the nodes in S to c_{OPT}. Let $c_U(S)$ denote the sum of contribution of the nodes in S to c_U for the matrix U returned by ϵ-SSMATRIX. Let $\bar{k}_{|S|}$ and $\bar{k}_n$ be the average of the smallest $|S|$ and n respectively of the vector weights assigned by the algorithm. It is easy to observe that $\bar{k}_{|S|} \leq \bar{k}_n$.

$$\begin{aligned} c_U(S) &= \sum_{v_i \in S} c(v_i) \left\| U(i) \right\|_1 \leq c_{max} \sum_{v_i \in S} \left\| U(i) \right\|_1 \\ &= c_{max} \bar{k}_{|S|} |S| \leq c_{max} \bar{k}_{|S|} \binom{m}{t} \leq \epsilon \bar{k}_{|S|} n/3. \end{aligned}$$

As every node in $V \setminus S$ receives weight at least $t = k - \epsilon k/3$ in U_{OPT} and at most k in U returned by ϵ-SSMATRIX, we have $c_U(V \setminus S) \leq \frac{c_{\mathrm{OPT}}(V \setminus S)}{1 - \epsilon/3}$. Now, we give a lower bound on the cost of the optimum solution $c_{\mathrm{OPT}}(V)$. We know that when costs of all the nodes are 1, then ϵ-SSMATRIX achieves optimum cost denoted by $c'_{\mathrm{OPT}}(V)$ (see Appendix A.2 for more details). As all the nodes of V have costs more than 1, we have:

$$c_{\mathrm{OPT}}(V) \geq c'_{\mathrm{OPT}}(V) = \bar{k}_n \cdot n \geq \bar{k}_{|S|} \cdot n.$$

Hence,

$$\frac{c_U(V)}{c_{\mathrm{OPT}}(V)} \leq \frac{c_U(S)}{\bar{k}_{|S|} n} + \frac{c_U(V \setminus S)}{c_{\mathrm{OPT}}(V \setminus S)} \leq \frac{\epsilon}{3} + \frac{1}{1 - \epsilon/3} \leq 1 + \epsilon.$$

This completes the proof. □

By bounding the binomial coefficients in Thm. 3.4, we obtain the following somewhat easier to interpret corollary:

Corollary 3.5. *If $c_{max} \leq (\epsilon/6) n^{\Omega(\epsilon)}$ and either a) $n^{\epsilon/6} \geq m \geq (2 \log_2 n)^{c_1}$ for some constant $c_1 > 1$ or b) $4 \log_2 n \leq m \leq c_2 \log_2 n$ for some constant c_2 then the Algorithm ϵ-SSMATRIX returns an $(1+\epsilon)$-approximation.*

4. Identity Cost Model

In this section, we consider the identity cost model, where the cost of intervention for any subset of variables is the same. Our goal is to construct the entire causal graph $\mathcal{G}$, while minimizing the number of interventions. Our algorithm is based on parameterizing the causal graph based on a specific type of collider structure. Before describing our algorithms, we recall the notion of d-separation and introduce this specific type of colliders that we rely on. Missing details from section are collected in Appendix B.

Colliders. Given a causal graph $\mathcal{G}(V \cup L, \mathcal{E})$, let $v_i, v_j \in V$ and a set of nodes $Z \subseteq V$. We say v_i and v_j are *d-separated* by Z if and only if every undirected path π between v_i and v_j is *blocked* by Z. A path π between v_i and v_j is blocked by Z if at least one of the following holds.

Rule 1: π contains a node $v_k \in Z$ such that the path $\pi = v_i \cdots \rightarrow v_k \rightarrow \cdots v_j$ or $v_i \cdots \leftarrow v_k \leftarrow \cdots v_j$.

Rule 2: $\pi = v_i \cdots \rightarrow v_k \leftarrow \cdots v_j$ contains a node v_k and both $v_k \notin Z$ and no descendant of v_k is in Z.

Lemma 4.1. *[Pearl, 2009] If v_i and v_j are d-separated by Z, then $v_i \perp\!\!\!\perp v_j \mid Z$.*

For the path $\pi = v_i \cdots \rightarrow v_k \leftarrow \cdots v_j$ between v_i and v_j, v_k is called a *collider* as there are two arrows pointing towards it. We say that v_k is a collider for the pair v_i and v_j, if there exists a path between v_i and v_j for which v_k is a collider. As shown by Rule 2, colliders play an important role in d-separation. We give a more restrictive definition for colliders that we will rely on henceforth.

Definition 4.2 (p-**colliders**). *Given a causal graph $\mathcal{G}(V \cup L, E \cup E_L)$. Consider $v_i, v_j \in V$ and $v_k \in V$. We say v_k is a p-collider for the pair v_i and v_j, if there exists a path $v_i \cdots \rightarrow v_k \leftarrow \cdots v_j$ in $\mathcal{G}$ and either $v_k \in \mathrm{Pa}(v_i) \cup \mathrm{Pa}(v_j)$ or has at least one descendant in $\mathrm{Pa}(v_i) \cup \mathrm{Pa}(v_j)$. Let $P_{ij} \subset V$ denote all the p-colliders between v_i and v_j.*

Intervening on p-colliders essentially breaks down all the *primitive inducing paths*. Primitive inducing paths are those whose endpoints cannot be separated by any conditioning [Richardson et al., 2002]. Now, between every pair of observable variables, we can define a set of p-colliders as above. Computing P_{ij} for the pair of variables v_i and v_j explicitly requires the knowledge of $\mathcal{G}$, however as we show below we can use randomization to overcome this issue.

The following parameterization of a causal graph will be useful in our discussions.

Definition 4.3 (τ-causal graph). *A causal graph $\mathcal{G}(V \cup L, \mathcal{E})$ is a τ-causal graph if for every pair of nodes in V the number of p-colliders is at most τ, i.e., $v_i, v_j \in V$ $(i \neq j)$, we have $|P_{ij}| \leq \tau$.*

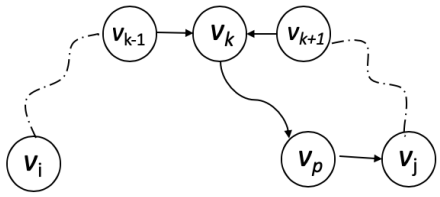

Figure 1. v_k is a p-collider for v_i, v_j as it has a path to v_p, a parent of v_j.

Note that every causal graph is at most $n-2$-causal. In practice, we expect τ to be significantly smaller. Given a causal graph $\mathcal{G}$, it is easy to determine the minimum values of τ for which it is τ-causal, as checking for p-colliders is easy. Our algorithm recovers $\mathcal{G}$ with number of interventions that grow as a function of τ and n.

Outline of our Approach. Let $\mathcal{G}$ be a τ-causal graph. As in [Kocaoglu et al., 2017b], we break our approach into multiple steps. Firstly, we construct the ancestral graph $\mathrm{Anc}(G)$ using the strongly separating set system (Definition 2.2) idea detailed in Section 2. For example, a strongly separating set system can be constructed with $m = 2\log n$ interventions by using the binary encoding of the numbers $1, \cdots, n$ [Kocaoglu et al., 2017b]. After that the algorithm has two steps. In the first step, we recover the observable graph G from $\mathrm{Anc}(G)$. In the next step, after obtaining the observable graph, we identify all the latents L between the variables in V to construct $\mathcal{G}$. In both these steps, an underlying idea is to construct intervention sets with the aim of making sure that all the p-colliders between every pair of nodes is included in at least one of the intervention sets. As we do not know the graph $\mathcal{G}$, we devise randomized strategies to hit all the p-colliders, whilst ensuring that we do not create a lot of interventions.

A point to note is that, we design the algorithms to achieve an overall success probability of $1 - O(1/n^2)$, however, the success probability can be boosted to any $1 - O(1/n^c)$ for any constant c, by just adjusting the constant factors (see for example the proof of Lemma B.2). Also for simplicity of discussion, we assume that we know τ. However as we discuss in Appendix B this assumption can be easily removed with an additional $O(\log \tau)$ factor.

4.1. Recovering the Observable Graph

$\mathrm{Anc}(G)$ encodes all the ancestral relations on observable variables V of the causal graph G. To recover G from $\mathrm{Anc}(G)$, we want to differentiate whether $v_i \to v_j$ represents an edge in G or a directed path going through other nodes in G. We use the following observation, if v_i is a parent of v_j, the path $v_i \to v_j$ is never blocked by any conditioning set $Z \subseteq V \setminus \{v_i\}$. If $v_i \notin \mathrm{Pa}(v_j)$, then we show that we can provide a conditioning set Z in some interventional distribution S such that $v_i \perp\!\!\!\perp v_j \mid Z, \mathrm{do}(S)$. For every pair of variables that have an edge in $\mathrm{Anc}(G)$, we design conditioning sets in Algorithm 2 that blocks all the paths between them.

Let $v_i \in \mathrm{Anc}(v_j) \setminus \mathrm{Pa}(v_j)$. We argue that conditioning on $\mathrm{Anc}(v_j) \setminus \{v_i\}$ in $\mathrm{do}(v_i \cup P_{ij})$ blocks all the paths from v_i to v_j. The first simple observation, from d-separation is that if we take a path that has no p-colliders between v_i to v_j (a p-collider free path) then it is blocked by conditioning on $\mathrm{Anc}(v_j) \setminus \{v_i\}$ i.e., $v_i \perp\!\!\!\perp v_j \mid \mathrm{Anc}(v_j) \setminus \{v_i\}$.

The idea then will be to intervene on colliders P_{ij} to remove these dependencies between v_i and v_j as shown by the following lemma.

Lemma 4.4. *Let* $v_i \in Anc(v_j)$. $v_i \perp\!\!\!\perp v_j \mid \mathrm{do}(v_i \cup P_{ij}), \mathrm{Anc}(v_j) \setminus \{v_i\}$ *iff* $v_i \notin \mathrm{Pa}(v_j)$.

From Lemma 4.4, we can recover the edges of the observable graph G provided we know the p-colliders between every pair of nodes. However, since the set of p-colliders is unknown without the knowledge of $\mathcal{G}$, we construct multiple intervention sets by independently sampling every variable with some probability. This ensures that there exists an intervention set S such that $\{v_i\} \cup P_{ij} \subseteq S$ and $v_j \notin S$ with high probability.

Formally, let $A_t \subseteq V$ for $t \in \{1, 2, \cdots, 72\tau' \log n\}$ be constructed by including every variable $v_i \in V$ with probability $1 - 1/\tau'$ where $\tau' = \max\{\tau, 2\}$. Let $\mathcal{A}_\tau = \{A_1, \cdots, A_{72\tau' \log n}\}$ be the collection of the set A_t's. Algorithm 2 uses the interventions in $\mathcal{A}_\tau$.

Algorithm 2 RECOVERG $(\mathrm{Anc}(G), \mathcal{A}_\tau)$

```
E = φ
for v_i → v_j in Anc(G) do
    Let A_ij = {A | A ∈ A_τ such that v_i ∈ A, v_j ∉ A}
    if ∀ A ∈ A_ij, v_i ⊥̸⊥ v_j | Anc(v_j)\{v_i}, do(A) then
        E = E ∪ {(v_i, v_j)}
    end if
end for
return E
```

Proposition 4.5. *Let* $\mathcal{G}(V \cup L, E \cup E_L)$ *be a* τ*-causal graph with observable graph* $G(V, E)$*. There exists a procedure to recover the observable graph using* $O(\tau \log n + \log n)$ *many interventions with probability at least* $1 - 1/n^2$.

Lower Bound. Complementing the above result, the following proposition gives a lower bound on the number of interventions by providing an instance of a $O(n)$-causal graph such that any non-adaptive algorithm requires $\Omega(n)$ interventions for recovering it. The lower bound comes because of the fact that the algorithm cannot rule out the possibility of latent.

Proposition 4.6. *There exists a causal graph* $\mathcal{G}(V \cup L, E \cup E_L)$ *such that every non-adaptive algorithm requires* $\Omega(n)$

Algorithm 3 LatentsWEdges($\mathcal{G}(V \cup L, E \cup E_L), \mathcal{B}_\tau$)

Consider the edge $v_i \to v_j \in E$.
Let $\mathcal{B}_{ij} = \{B \setminus \{v_i\} \mid B \in \mathcal{B}_\tau \text{ s.t. } v_i \in B, v_j \notin B\}$
if $\forall B \in \mathcal{B}_{ij}, \Pr[v_j \mid v_i, \text{Pa}(v_j), \text{do}(\text{Pa}(v_i) \cup B)] \neq \Pr[v_j \mid \text{Pa}(v_j), \text{do}(\{v_i\} \cup \text{Pa}(v_i) \cup B)]$ **then**
  $L \leftarrow L \cup l_{ij}, E_L \leftarrow E_L \cup \{(l_{ij}, v_i), (l_{ij}, v_j)\}$
end if
return $\mathcal{G}(V \cup L, E \cup E_L)$

many interventions to recover even the observable graph $G(V, E)$ of $\mathcal{G}$.

4.2. Detecting the Latents

We now describe algorithms to identify latents that affect the observable variables V to learn the entire causal graph $\mathcal{G}(V \cup L, E \cup E_L)$. We start from the observable graph $G(V, E)$ constructed in the previous section. Our goal will be to use the fact that $\mathcal{G}$ is a τ-causal graph, which means that $|P_{ij}| \leq \tau$ for every pair v_i, v_j. Since we assumed that each latent variable (in L) effects at most two observable variables (in V), we can split the analysis into two cases: a) pairs of nodes in G without an edge (non-adjacent nodes) and b) pairs of nodes in G with a direct edge (adjacent). In Algorithm LATENTSNEDGES (Appendix B), we describe the algorithm for identifying the latents effecting pairs of non-adjacent nodes. The idea is to block the paths by conditioning on parents and intervening on p-colliders. We use the observation that for any non-adjacent pair v_i, v_j an intervention on the set P_{ij} and conditioning on the parents of v_i and v_j will make v_i and v_j independent, unless there is a latent between them.

Proposition 4.7. *Let $\mathcal{G}(V \cup L, E \cup E_L)$ be a τ-causal graph with observable graph $G(V, E)$. Algorithm* LATENTSNEDGES *with $O(\tau^2 \log n + \log n)$ interventions recovers all latents effecting pairs of non-adjacent nodes in the observable graph G with probability at least $1 - 1/n^2$.*

Latents Affecting Adjacent Nodes in G. Suppose we have an edge $v_i \to v_j$ in $G(V, E)$ and we want to detect whether there exists a latent l_{ij} that effects both of them. Here, we cannot block the edge path $v_i \to v_j$ by conditioning on any $Z \subseteq V$ in any given interventional distribution $\text{do}(S)$ where S does not contain v_j. However, intervening on v_j also disconnects v_j from its latent parent. Therefore, CI-tests are not helpful. Hence, we make use of another test called *do-see* test [Kocaoglu et al., 2017b], that compares two probability distributions. We assume there exists an oracle that answers whether two distributions are the same or not. This is a well-studied problem with sublinear (in domain size) bound on the sample size needed for implementing this oracle [Chan et al., 2014].

Assumption 4.8 (Distribution Testing (DT)-Oracle)**.** *Given any $v_i, v_j \in V$ and $Z, S \subseteq V$ tests whether two distributions $\Pr[v_j \mid v_i, Z, \text{do}(S)]$ and $\Pr[v_j \mid Z, \text{do}(S \cup \{v_i\})]$ are identical or not.*

The intuition of the do-see test is as follows: if v_i and v_j are the only two nodes in the graph G with $v_i \to v_j$, then, $\Pr[v_j \mid v_i] = \Pr[v_j \mid \text{do}(v_i)]$ iff there exists no latent that effects both of them. This follows from the *conditional invariance* principle [Bareinboim et al., 2012] (or page 24, property 2 in [Pearl, 2009]). Therefore, the presence or absence of latents can be established by invoking a DT-oracle.

As we seek to minimize the number of interventions, our goal is to create intervention sets that contain p-colliders between every pair of variables that share an edge in G. However, in Lemmas 4.9, 4.10 we argue that it is not sufficient to consider interventions with only p-colliders. We must also intervene on $Pa(v_i)$ to detect a latent between $v_i \to v_j$. The main idea behind LATENTSWEDGES is captured by the following two lemmas.

Lemma 4.9 (No Latent Case)**.** *Suppose $v_i \to v_j \in G$ and $v_i, v_j \notin B$, and $P_{ij} \subseteq B$ then, $\Pr[v_j \mid v_i, \text{Pa}(v_j), \text{do}(\text{Pa}(v_i) \cup B)] = \Pr[v_j \mid \text{Pa}(v_j), \text{do}(\{v_i\} \cup \text{Pa}(v_i) \cup B)]$ if there is no latent l_{ij} with $v_i \leftarrow l_{ij} \to v_j$.*

Lemma 4.10 (Latent Case)**.** *Suppose $v_i \to v_j \in G$ and $v_i, v_j \notin B$, and $P_{ij} \subseteq B$, then, $\Pr[v_j \mid v_i, \text{Pa}(v_j), \text{do}(\text{Pa}(v_i) \cup B)] \neq \Pr[v_j \mid \text{Pa}(v_j), \text{do}(\{v_i\} \cup \text{Pa}(v_i) \cup B)]$ if there is a latent l_{ij} with $v_i \leftarrow l_{ij} \to v_j$.*

From Lemmas 4.9, 4.10, we know that to identify a latent l_{ij} between $v_i \to v_j$, we must intervene on all the p-colliders between them with $Pa(v_i) \cup \{v_i\}$. To do this, we again construct random intervention sets. Let $B_t \subseteq V$ for $t \in \{1, 2, \cdots, 72\tau' \log n\}$ be constructed by including every variable $v_i \in V$ with probability $1 - 1/\tau'$ where $\tau' = \max\{\tau, 2\}$. Let $\mathcal{B}_\tau = \{B_1, \cdots, B_{72\tau' \log n}\}$ be the collection of the sets. Consider a pair $v_i \to v_j$. To obtain the interventions given by the above lemmas, we iterate over all sets in $\mathcal{B}_\tau$ and identify all the sets containing v_i, but not v_j. From these sets, we remove v_i to obtain $\mathcal{B}_{ij}$. These new interventions are then used in LATENTSWEDGES to perform the required distribution tests using a DT-oracle on the interventions $B \cup Pa(v_i)$ and $B \cup Pa(v_i) \cup \{v_i\}$ for every $B \in \mathcal{B}_{ij}$. We can show:

Proposition 4.11. *Let $\mathcal{G}(V \cup L, E \cup E_L)$ be a τ-causal graph with observable graph $G(V, E)$.* LATENTSWEDGES *with $O(n\tau \log n + n \log n)$ interventions recovers all latents effecting pairs of adjacent nodes in the observable graph G with probability at least $1 - 1/n^2$.*

Putting it all Together. Using Propositions 4.5, 4.7, and 4.11, we get the following result. Note that $\tau \leq n - 2$.

Theorem 4.12. *Given access to a τ-causal graph $\mathcal{G} = \mathcal{G}(V \cup L, E \cup E_L)$ through Conditional Independence (CI) and Distribution Testing (DT) oracles, Algorithms* RECOVERG, LATENTSNEDGES, *and* LATENTSWEDGES *put together recovers $\mathcal{G}$ with $O(n\tau \log n + n \log n)$ interventions, with probability at least $1 - O(1/n^2)$ (where $|V| = n$).*

5. Experiments

In this section, we compare the total number of interventions required to recover causal graph $\mathcal{G}$ parameterized by p-colliders (See Section 4) vs. maximum degree utilized by [Kocaoglu et al., 2017b].

Since the parameterization of these two results are different, a direct comparison between them is not always possible. If $\tau = o(d^2/n)$, we use fewer interventions than Kocaoglu et al. [2017b] for recovering the causal graph. Roughly, for any $0 \leq \epsilon \leq 1$, (a) when $\tau < n^{\epsilon}, d > n^{(1+\epsilon)/2}$, our bound is better, (b) when $\tau > n^{\epsilon}, \tau < d < n^{(1+\epsilon)/2}$, then we can identify latents using the algorithms of Kocaoglu et al. [2017b] after using our algorithm for observable graph recovery, and (c) when $\tau > d > n^{\epsilon}, d < n^{(1+\epsilon)/2}$, the bound in Kocaoglu et al. [2017b] is better.

In this section, our main motivation is to show that p-colliders can be a useful measure of complexity of a graph. As discussed in Section 1, even few nodes of high degree could make d^2 quite large.

Setup. We demonstrate our results by considering sparse random graphs generated from the families of: (i) Erdös-Rényi random graphs $G(n, c/n)$ for constant c, (ii) Random bipartite graphs generated using $G(n_1, n_2, c/n)$ model, with partitions L, R and edges directed from L to R, (iii) Random directed trees with degrees of nodes generated from power law distribution. In each of the graphs that we consider, we include latent variables by sampling 5% of $\binom{n}{2}$ pairs and adding a latent between them.

Finding p-colliders. Let $\mathcal{G}$ contain observable variables and the latents. To find p-colliders between every pair of observable nodes of $\mathcal{G}$, we enumerate all paths between them and check if any of the observable nodes on a path can be a possible p-collider. As this became practically infeasible for larger values of n, we devised an algorithm that runs in polynomial time (in the size of the graph) by constructing an appropriate flow network and finding maximum flow in this network. Please refer to Appendix C for more details.

Results. In our plots (Figure 2), we compare the maximum undirected degree (d) with the maximum number of p-colliders between any pair of nodes (which defines τ). We ran each experiment 10 times and plot the mean value along with one standard deviation error bars.

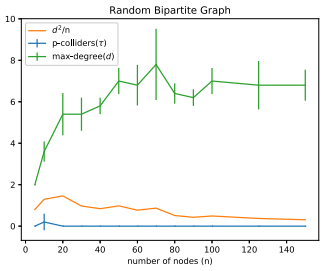

Figure 2. Comparison of τ vs. maximum degree in sparse random bi-partite graphs.

For random bipartite graphs, that can be used to model causal relations over time, we use equal partition sizes $n_1 = n_2 = n/2$ and plot the results for $\mathcal{G}(n/2, n/2, c/n)$ for constant $c = 5$. We observe that the behaviour is uniform for small constant values of c. In Figure 2, we observe that the maximum number of p-colliders(τ) is close to zero for all values of n while the values of d^2/n using the mean value of d, is significantly higher. So, in the range considered our algorithms use fewer interventions. We show similar results for other random graphs in Appendix C.

Therefore, we believe that minimizing the number of interventions based on the notion of p-colliders is a reasonable direction to consider.

6. Concluding Remarks

We have studied how to recover a causal graph in presence of latents while minimizing the intervention cost. In the linear cost setting, we give a 2-approximation algorithm for ancestral graph recovery. This approximation factor can be improved to $(1 + \epsilon)$ under some additional assumptions. Removing these assumptions would be an interesting direction for future work. In the identity cost setting, we give a randomized algorithm to recover the full causal graph, through a novel characterization based on p-colliders. In this setting, understanding the optimal intervention cost is open, and an important direction for research.

While we focus on non-adaptive settings, where all the interventions are constructed at once in the beginning, an adaptive (sequential) setting has received recent attention [He & Geng, 2008; Shanmugam et al., 2015], and is an interesting direction in both our cost models.

Acknowledgements

The first two named authors would like to thank Nina Mishra, Yonatan Naamad, MohammadTaghi Hajiaghayi and Dominik Janzing for many helpful discussions during the initial stages of this project. We would also like to thank anonymous reviewers for many helpful suggestions. This work was partially supported by NSF grants CCF-1934846, CCF-1908849, and CCF-1637536.

References

Adamic, L. A. and Huberman, B. A. Power-law distribution of the world wide web. *Science*, 287(5461):2115–2115, 2000.

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

Bareinboim, E., Brito, C., and Pearl, J. Local characterizations of causal bayesian networks. In *Graph Structures for Knowledge Representation and Reasoning*, pp. 1–17. Springer, 2012.

Canonne, C. L., Diakonikolas, I., Kane, D. M., and Stewart, A. Testing conditional independence of discrete distributions. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–57. IEEE, 2018.

Chan, S.-O., Diakonikolas, I., Valiant, P., and Valiant, G. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1193–1203. SIAM, 2014.

Eberhardt, F. Causation and intervention. *PhD Thesis, Carnegie Mellon University*, 2007.

Eberhardt, F. and Scheines, R. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.

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

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

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

Heinze-Deml, C., Maathuis, M. H., and Meinshausen, N. Causal structure learning. *Annual Review of Statistics and Its Application*, 5:371–391, 2018.

Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., and Schölkopf, B. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pp. 689–696, 2009.

Hyttinen, A., Eberhardt, F., and Hoyer, P. O. Experiment selection for causal discovery. *The Journal of Machine Learning Research*, 14(1):3041–3071, 2013a.

Hyttinen, A., Hoyer, P. O., Eberhardt, F., and Jarvisalo, M. Discovering cyclic causal models with latent variables: A general sat-based procedure. *arXiv preprint arXiv:1309.6836*, 2013b.

Jukna, S. *Extremal combinatorics: with applications in computer science*. Springer Science & Business Media, 2011.

Katona, G. On separating systems of a finite set. *Journal of Combinatorial Theory*, 1(2):174–194, 1966.

Kisvölcsey, Á. Flattening antichains. *Combinatorica*, 1(26): 65–82, 2006.

Kocaoglu, M., Dimakis, A., and Vishwanath, S. Cost-optimal learning of causal graphs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1875–1884. JMLR. org, 2017a.

Kocaoglu, M., Shanmugam, K., and Bareinboim, E. Experimental design for learning causal graphs with latent variables. In *Advances in Neural Information Processing Systems*, pp. 7018–7028, 2017b.

Kruskal, J. B. The number of simplices in a complex. *Mathematical Optimization Techniques*, 10:251–278, 1963.

Lindgren, E., Kocaoglu, M., Dimakis, A. G., and Vishwanath, S. Experimental design for cost-aware learning of causal graphs. In *Advances in Neural Information Processing Systems*, pp. 5279–5289, 2018.

Loh, P.-L. and Bühlmann, P. High-dimensional learning of linear causal networks via inverse covariance estimation. *The Journal of Machine Learning Research*, 15(1):3065–3105, 2014.

MacWilliams, F. J. and Sloane, N. J. A. *The theory of error-correcting codes*, volume 16. Elsevier, 1977.

Mao-Cheng, C. On separating systems of graphs. *Discrete Mathematics*, 49(1):15–20, 1984.

Parviainen, P. and Koivisto, M. Ancestor relations in the presence of unobserved variables. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 581–596. Springer, 2011.

Pearl, J. *Causality: models, reasoning and inference*, volume 29. Springer, 2000.

Pearl, J. *Causality: Models, Reasoning, and Inference*. Cambridge university press, 2009.

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

Shanmugam, K., Kocaoglu, M., Dimakis, A. G., and Vishwanath, S. Learning causal graphs with small interventions. In *Advances in Neural Information Processing Systems*, pp. 3195–3203, 2015.

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

Shpitser, I. and Pearl, J. Identification of joint interventional distributions in recursive semi-markovian causal models. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pp. 1219–1226, 2006.

Silva, R., Scheine, R., Glymour, C., and Spirtes, P. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7(Feb):191–246, 2006.

Spirtes, P., Glymour, C. N., Scheines, R., Heckerman, D., Meek, C., Cooper, G., and Richardson, T. *Causation, prediction, and search*. MIT press, 2000a.

Spirtes, P., Glymour, C. N., Scheines, R., Heckerman, D., Meek, C., Cooper, G., and Richardson, T. *Causation, prediction, and search*. MIT press, 2000b.

Tian, J. and Pearl, J. A general identification condition for causal effects. In *AAAI*, 2002.

Verma, T. and Pearl, J. An algorithm for deciding if a set of observed independencies has a causal explanation. In *Uncertainty in artificial intelligence*, pp. 323–330. Elsevier, 1992.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 804–813. AUAI Press, 2011.