Scalable Computation of Causal Bounds

Madhumitha Shridharan [1] **Garud Iyengar** [1]

Abstract

We consider the problem of computing bounds for causal inference problems with unobserved confounders, where identifiability does not hold. Existing non-parametric approaches for computing such bounds use linear programming (LP) formulations that quickly become intractable for existing solvers because the size of the LP grows exponentially in the number of edges in the underlying causal graph. We show that this LP can be significantly pruned by carefully considering the structure of the causal query, allowing us to compute bounds for significantly larger causal inference problems as compared to what is possible using existing techniques. This pruning procedure also allows us to compute the bounds in closed form for a special class of causal graphs and queries, which includes a well-studied family of problems where multiple confounded treatments influence a outcome.We also propose a very efficient greedy heuristic that produces very high quality bounds, and scales to problems that are several orders of magnitude larger than those for which the pruned LP can be solved.

1. Introduction

In most real world applications of causal inference (Imbens & Rubin, 2015; Pearl, 2009) there exist variables that are critical to the identification of causal effects, but are either unknown or unmeasured, i.e. there are *unobserved* confounders. While it is impossible to precisely identify causal effects in the presence of unobserved confounders, it is possible to obtain bounds on the causal query.

There have been multiple such attempts to bound causal effects for small special graphs. Evans (2012) bound the causal effects in the special case where any two observed variables are neither adjacent in the graph, nor share a latent parent. Richardson et al. (2014) bound the causal effect of a treatment on a parameter of interest by invoking additional (untestable) assumptions and assess how inference about the treatment effect changes as these assumptions are varied. Kilbertus et al. (2020) and Zhang & Bareinboim (2021) develop algorithms to compute causal bounds for extensions of the instrumental variable model in a continuous setting. Geiger & Meek (2013) bound causal effects in a model under specific parametric assumptions. Finkelstein & Shpitser (2020) develop a method for obtaining bounds on causal parameters using rules of probability and restrictions on counterfactuals implied by causal graphs.

While fewer in number, there have also been attempts to bound causal effects in large general graphs. Poderini et al. (2020) propose techniques to compute bounds in special large graphs with multiple instruments and observed variables. Finkelstein et al. (2021) propose a method for partial identification in a class of measurement error models, and Duarte et al. (2021) propose a polynomial programming based approach to solve general causal inference problems, but their procedure is computationally intensive for large graphs.

In this work, we extend the class of large graphs for which causal effects can be bounded. In particular, we focus on a class of causal inference problems where causal bounds can be obtained using linear programming (LP) (Balke & Pearl, 1994; Zhang & Bareinboim, 2017; Pearl, 2009; Sjölander et al., 2014). Recently, Sachs et al. (2020; 2021) identified a large problem class for which LPs can be used to compute causal bounds, and have developed an algorithm for formulating the objective function and the constraints of the corresponding LP. This problem class is a generalization of the instrumental variable setting, and is thus widely applicable. However, as we describe later, the size of the LP is exponential in the number of edges in the causal graph, and therefore, the straightforward formulation of the LP can be tractably solved only for very small causal graphs. In this work, we show how to use the structure of the causal query and the underlying graph to significantly reduce the size of the LP, and as a consequence, significantly increase the size of the graphs for which the LP method remains tractable. Our main contributions are as follows:

[1]Department of Industrial Engineering and Operations Research, Columbia University, New York, USA. Correspondence to: Madhumitha Shridharan <ms6143@columbia.edu>, Garud Iyengar <garud@ieor.columbia.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

(a) We show that an exponential number of variables in the LP can be aggregated to reduce the size of the problem without impacting the quality of the bound. See Section 3.1 for details. The reduction in size can be very substantial – compare $|R|$ with $|H|$ in Table 1 (details in Section 3.1).

(b) Although we show there exists a much smaller LP that can be solved to compute the bounds, we get a computational advantage only if this pruned LP can be constructed efficiently. Theorems 3.4 and 3.6 establish that one can construct the pruned LP without first constructing the original LP. These results critically leverage the structure of the LP corresponding to a causal inference problem. In particular, they leverage the fact that all possible functions mapping the parents $pa(V)$ to a variable V are allowed. Note that, without this second result, the savings implied by the pruned LP cannot be realized. See Section 3.1 for details.

(c) Next, we show that the structural results that help us construct the pruned LP allow us to compute the bounds in closed form for a special class of causal inference problems. This class of problems includes as a special case the problems considered by Wang & Blei (2021; 2019a), where multiple confounded treatments influence a outcome. Moreover, we are able to compute these bounds even when there are causal relationships between the treatments. See Section 3.2.1 for details.

(d) Finally, we propose a simple greedy heuristic to compute optimal solutions for the pruned LPs. We show that this heuristic allows us to compute approximate bounds for much larger scale graphs with very minimal degradation in performance. See Section 4 for details.

Although we work with causal graphs with binary variables in this paper, generalizing our results to categorical variables is straightforward.

The organization of the rest of this paper is as follows. In Section 2, we present the intuition behind our contributions using a running example. In Section 3 we introduce the formalism in Sachs et al. (2020; 2021). In Section 3.1 we introduce our main structural results for pruning the LPs. In Section 3.2 we show that the LP bounds can be computed in closed form for a large class of problems, and in Section 3.2.1 we show an example of this class of problems. In Section 4 we introduce our greedy heuristic and benchmark its performance. Section 5 discusses possible extensions.

2. Example

Consider the causal graph in Figure 1 with $X, Y, Z \in \{0, 1\}$. Suppose the data given is $p_{xy.z} = \mathbb{P}(X = x, Y = y|Z = z)$ and the goal is to compute a lower bound for the causal

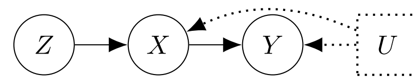


Figure 1: Causal Graph for Query $\mathcal{Q}$

query $\mathcal{Q} = \mathbb{P}(Y = 1|do(X = 1))$. Here, U is a potentially high dimensional unobserved confounder that is difficult to model directly. Response function variables circumvent this difficulty by modelling the *impact* of U on the relationships between the observed variables. For each fixed value for the unknown confounder U, the variable X is a function of Z; thus, confounder U effectively selects one function from the set $\mathcal{F} = \{f : f \text{ is a function from } Z \to X\}$. Similarly, U selects one function from the set $\mathcal{G} = \{g : g \text{ is a function from } X \to Y\}$. It is easy to see that $|\mathcal{F}| = |\mathcal{G}| = 4$, and therefore, the elements can be indexed by $r = (r_X, r_Y) \in R = \{1, \ldots, 4\}^2$, i.e. f_{r_X} denotes the r_X-th function from $\mathcal{F}$ and g_{r_Y} denotes the r_Y-th function from $\mathcal{G}$. Thus, the response function variables r_X and r_Y can be used to model the impact of U, and Z and $r \in R$ uniquely determine $X = f_{r_X}(Z)$ and $Y = g_{r_Y}(X) = g_{r_Y}(f_{r_X}(Z))$. Note that $|R|$ is exponential in the number of arcs in the causal graph.

The unknown distribution over the high dimensional U can be equivalently modeled via the joint distribution $q_{r_X r_Y} = \mathbb{P}(r_X, r_Y)$. Let

$$R_{xy.z} = \{(r_X, r_Y) : f_{r_X}(z) = x, g_{r_Y}(x) = y\}, \quad (1)$$

denote the set of r-values that map $z \mapsto (x, y)$. Hence, $\mathbb{P}(X = x, Y = y|Z = z) = \sum_{(r_X, r_Y) \in R_{xy.z}} q_{r_X r_Y}$. Furthermore, let $R_{\mathcal{Q}}$ denote the set of r values consistent with the query $\mathcal{Q} = \mathbb{P}(Y = 1|do(X = 1))$ i.e.

$$R_{\mathcal{Q}} = \{(r_X, r_Y) : g_{r_Y}(1) = 1\} \quad (2)$$

Hence, $\mathbb{P}(Y = 1|do(X = 1)) = \sum_{(r_X, r_Y) \in R_{\mathcal{Q}}} q_{r_X r_Y}$. Then the following LP gives a lower bound on the query (Balke & Pearl, 1994; Sachs et al., 2020):

$$\begin{array}{rl} \min_q & \sum_{(r_X, r_Y) \in R_{\mathcal{Q}}} q_{r_X r_Y} \\ \text{s.t.} & \sum_{(r_X, r_Y) \in R_{xy.z}} q_{r_X r_Y} = p_{xy.z}, \forall (x, y, z) \\ & q \geq 0, \end{array} \quad (3)$$

Since $r \in R$ uniquely determines the value of (X, Y) for any fixed value for Z, $R = \cup_{x,y \in \{0,1\}^2} R_{xy.z}$ is a partition for the set R for any fixed z. The constraints in LP (3) imply that $1 = \sum_{(x,y) \in \{0,1\}^2} p_{xy.z} = \sum_{(x,y) \in \{0,1\}^2} \sum_{(r_X, r_Y) \in R_{xy.z}} q_{r_X r_Y} = \sum_{(r_X, r_Y) \in R} q_{r_X r_Y}$. Therefore, we do not include the constraint $\sum_{(r_X, r_Y) \in R} q_{r_X r_Y} = 1$ in the LP.

Next, we show how to reduce the size of the LP (3) by aggregating variables. Let $h : Z \to (X, Y)$ denote any function $Z \mapsto (X, Y)$. We also refer to h as a *hyperarc* since it can be interpreted as an arc in a hypergraph. Let

$$R_h = \left\{ \begin{array}{l} (r_X, r_Y) \in R : \\ \qquad \big(f_{r_X}(0), g_{r_Y}(f_{r_X}(0))\big) = h(0) \\ \qquad \big(f_{r_X}(1), g_{r_Y}(f_{r_X}(1))\big) = h(1) \end{array} \right\} \quad (4)$$

denote the set of r values consistent with the hyperarc h. Then all $r \in R_h$ contribute to the same two constraints: $(x, y, z) = \big(h(0), 0\big)$ and $(x, y, z) = \big(h(1), 1\big)$. Therefore, since we have minimization objective, we can set $q_r = 0$ for all $r \in R_h$ with objective coefficient $\mathbf{1}\{r \in R_{\mathcal{Q}}\} > \min_{s \in R_h} \mathbf{1}\{s \in R_{\mathcal{Q}}\}$, and aggregate all variables q_r such that $r \in R_h$, i.e. we can reformulate the LP in terms of variables q_h with the objective coefficient $c_h = \min_{r \in R_h} \mathbf{1}\{r \in R_{\mathcal{Q}}\} = \mathbf{1}\{R_h \subseteq R_{\mathcal{Q}}\}$. The causal graph structure implies that $R_h \neq \emptyset$ only for a subset of hyperarcs. Let H denote the set of *valid* hyperarcs for which $R_h \neq \emptyset$. Thus, the LP (3) can be reformulated as

$$\begin{array}{rl} \min_q & \sum_{h \in H} c_h q_h \\ \text{s.t.} & \sum_{h \in H : h(z) = (x,y)} q_h = p_{xy.z}, \forall (x, y, z) \\ & q \geq 0, \end{array} \quad (5)$$

where $c_h = \mathbf{1}\{R_h \subseteq R_{\mathcal{Q}}\}$. This reformulation has *exponentially* fewer variables; however, it is useful only if the set of valid hyperarcs H and the corresponding costs c_h can be efficiently computed.

2.1. Characterizing Valid Hyperarcs

The challenge here is to avoid iterating over all values in the set R to produce a value which validates h. Suppose $h(0) = (x_0, y_0)$ and $h(1) = (x_1, y_1)$ for $x_i, y_i \in \{0, 1\}$, $i = 0, 1$. Then the "maps" (may not be functions) f_h and g_h implied by h are as follows:

$$f_h(z) = \begin{cases} x_0 & \text{if } z = 0 \\ x_1 & \text{if } z = 1 \end{cases} \qquad g_h(x) = \begin{cases} y_0 & \text{if } x = x_0 \\ y_1 & \text{if } x = x_1 \end{cases}$$

Since R indexes the set of all possible functions in $\mathcal{F}$ and $\mathcal{G}$, h is a valid hyperarc if, and only if, the "maps" f_h (resp. g_h) is consistent with some function $f \in \mathcal{F}$ (resp. $g \in \mathcal{G}$). The latter is true if, and only if, it is not the case that $x_0 = x_1$ but $y_0 \neq y_1$. Hence, to check the validity of h, it is sufficient to check if $x_0 = x_1$ but $y_0 \neq y_1$. We extend this notion of consistency to more complex causal graphs in Section 3.

2.2. Efficiently computing c_h

We start with the following definitions.

Definition 2.1 (Complete Consistency)**.** A hyperarc h is completely consistent with the query $\mathcal{Q}$ if $R_h \subseteq R_{\mathcal{Q}}$. A conditional probability $p_{xy.z}$ is completely consistent with the query $\mathcal{Q}$ if $R_{xy.z} \subseteq R_{\mathcal{Q}}$.

Thus, $c_h = 1$ if, and only if, h is completely consistent with the query $\mathcal{Q}$. Next, we describe how to check whether a hyperarc h is completely consistent. We begin characterizing complete consistency for conditional probabilities. We establish that

$$p_{xy.z} \text{ is completely consistent with } \mathcal{Q} \iff y = 1, x = 1.$$

Clearly, any $(r_X, r_Y) \in R_{11.z}$ maps $X = 1$ to $Y = 1$, and thus, $(r_X, r_Y) \in R_{\mathcal{Q}}$ defined in (2). Hence, any conditional probability of the form $p_{11.z}$ is completely consistent with $\mathcal{Q}$. To see that this is the *only* form a completely consistent probability can take, we consider two cases:

- Suppose $x \neq 1$ i.e. consider the probability $p_{01.z}$. Since (r_X, r_Y) index the set of all possible functions $\mathcal{F}$ and $\mathcal{G}$, there exists (r_X, r_Y) such that:
  - $\big(f_{r_X}(z), g_{r_Y}(f_{r_X}(z))\big) = (0, 1)$, i.e. $(r_X, r_Y) \in R_{01.z}$, but,
  - $g_{r_Y}(1) = 0$, i.e. $(r_X, r_Y) \notin R_{\mathcal{Q}}$

  Hence, there exists (r_X, r_Y) such that $(r_X, r_Y) \in R_{01.z}$, but $(r_X, r_Y) \notin R_{\mathcal{Q}}$ i.e. $p_{01.z}$ is not completely consistent.

- Suppose $y \neq 1$ i.e. consider the probability $p_{10.z}$. Then every $(r_X, r_Y) \in R_{10.z}$ maps $X = 1$ to $Y = 0$, and therefore, $(r_X, r_Y) \notin R_{\mathcal{Q}}$. Hence $p_{10.z}$ is not completely consistent.

Next, we show that the hyperarc h is completely consistent with $\mathcal{Q}$, if and only if, there exists z such that $h(z) = (x, y)$ and the conditional probability $p_{xy.z}$ is completely consistent with $\mathcal{Q}$. Thus, for the query $\mathcal{Q} = \mathbb{P}(Y = 1 | do(X = 1))$, we have that

$$\begin{aligned} & h \text{ is completely consistent with } \mathcal{Q} \qquad (6) \\ \iff & h(z) = (1, 1) \text{ for some } z \in \{0, 1\} \end{aligned}$$

First suppose $h(z) = (1, 1)$ for some $z \in \{0, 1\}$. For every $(r_X, r_Y) \in R_h$, $(f_{r_X}(z), g_{r_Y}(f_{r_X}(z))) = (1, 1)$ i.e. (r_X, r_Y) maps $Z = z$ to $(X = 1, Y = 1)$. Hence, $(r_X, r_Y) \in R_{11.z}$. Since $p_{11.z}$ is completely consistent, $(r_X, r_Y) \in R_{\mathcal{Q}}$. Therefore, $R_h \subseteq R_{\mathcal{Q}}$ i.e. h is completely consistent with $\mathcal{Q}$. In order to establish the reverse direction, suppose $h(z) \neq (1, 1)$, for all $z \in \{0, 1\}$. Since R is exhaustive, there exists $r \in R$ such that:

(i) r maps $Z = z$ to $(X, Y) = h(z) \neq (1, 1)$, for all $z \in \{0, 1\}$ i.e. $r \in R_h$

(ii) r maps $X = 1$ to $Y \neq 1$, i.e. $(r_X, r_Y) \notin R_{\mathcal{Q}}$

Hence h is not completely consistent with $\mathcal{Q}$.

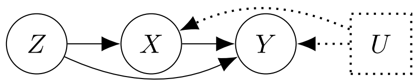


Figure 2: Causal Graph for Query $\mathcal{Q}_1$

To summarize, to compute c_h, it is sufficient to check if h is completely consistent with $\mathcal{Q}$, which can be easily characterized *without* enumerating all the elements in R. In Section 3 we show how to extend this result to more complex graphs.

2.3. Bounds in Closed Form

Now consider the causal graph in Figure 2. Given data $p_{xy.z}$, we show how to compute bounds for the modified query $\mathcal{Q}_1 = \mathbb{P}(Y = 1|do(Z = 1, X = 1))$ in closed form. More precisely, we show that the optimal value of the pruned LP for this problem is the sum of the completely consistent probabilities. Retracing the steps above we get $c_h = 1$ i.e. h is completely consistent with $\mathcal{Q}_1$ if and only if, $h(1) = (1, 1)$. The objective of the pruned LP for the problem is:

$$\begin{aligned}
&\sum_{\{h \in H: c_h = 1\}} q_h \\
&= \sum_{\{h \in H:\, h \text{ is completely consistent with } \mathcal{Q}_1\}} q_h \\
&= \sum_{\{h \in H: h(1)=(1,1)\}} q_h \\
&= p_{11.1} \qquad (7)
\end{aligned}$$

where (7) follows from the constraints of the pruned LP.

As a step towards characterizing graphs and queries for which such closed form bounds are possible, we understand why closed form bounds do not exist for the original causal graph in Figure 1 and the query $\mathcal{Q} = \mathbb{P}(Y = 1|do(X = 1))$. From (6) we have $c_h = 1$ i.e. h is completely consistent with $\mathcal{Q}$, if and only if, there exists $z \in \{0, 1\}$ such that $h(z) = (1, 1)$. The objective of the pruned LP is:

$$\begin{aligned}
\sum_{\{h \in H: c_h = 1\}} q_h &= \sum_{\{h \in H: \exists z \in \{0,1\}, h(z)=(1,1)\}} q_h \\
&\neq \sum_{h \in H: h(0)=(1,1)} q_h + \sum_{h \in H: h(1)=(1,1)} q_h,
\end{aligned}$$

since $\{h \in H : h(0) = (1, 1)\} \cap \{h \in H : h(1) = (1, 1)\} \neq \emptyset$. Hence, we cannot compute closed form bounds for the query.

Perhaps, the issue was that the "input" variable Z was not part of the intervention. So, suppose the query is $\mathcal{Q}_2 = \mathbb{P}(Y = 1|do(X = 1, Z = 1))$ instead. In this case, the steps above establish that $c_h = 1$ if and only if, there exists $z \in \{0, 1\}$ such that $h(z) = (1, 1)$. This is identical to our condition for c_h in $\mathcal{Q}$. Therefore, we will still not be able to compute closed form bounds. What went wrong? Intuitively, Z is *redundant* in the intervention $do(X = 1, Z = 1)$, since Z's effects on Y are blocked by intervening on X. The specification for c_h captures this intuition by leaving Z out when characterizing c_h. Hence, it appears that we need that the "input" variable must be non-redundant in the intervention in order to compute closed form bounds. Note that in the causal graph in Figure 2 and query $\mathcal{Q}_1$, the "input" variable Z is non-redundant in the intervention. Hence, we could compute closed form bounds. This is the premise of Assumption 3.10, which characterizes an important subclass of general causal inference problems for which the bounds can be computed in closed form.

3. General Causal Inference Problems

Let G denote the causal graph. Let $V_1, \ldots, V_n$ denote the variables in G in topologically sorted order. We let $N = \{1, \ldots, n\}$ denote the set of indices for the variables. We use lower case letters for the values for the variables, and the notation $V_i = v_i$ denotes that the variable V_i takes the value $v_i \in \{0, 1\}$. For any subset $S \subseteq N$, we define $V_S := \{V_i : i \in S\}$, and the notation $V_S = v_S$ denotes the variable $V_i = v_i$, for all $i \in S$, for some $v \in \{0, 1\}^{|N|}$. We consider the following class of "partitioned" causal graphs (Sachs et al., 2020).

Assumption 3.1 ((Sachs et al., 2020))**.** The index set N is partitioned into two sets $N = A \cup B$, where V_B topologically follow V_A, V_A can have a common unobserved confounder U_A, and V_B can have a common unobserved confounder U_B; however, no pair of variables (V_i, V_j), where $i \in A$ and $j \in B$, can share an unobserved confounder.

In the simple causal graphs discussed in Section 2, $V_A = \{Z\}$ and $V_B = \{X, Y\}$.

The conditional probability distribution $p_{v_B.v_A} = \mathbb{P}(V_B = v_B|V_A = v_A)$ is known, and our goal is to compute bounds for the causal query $\mathcal{Q} = \mathbb{P}(V_O = q_O|do(V_I = q_I))$, i.e. the probability of the output event $V_O = q_O$ given an intervention $do(V_I = q_I)$.

Assumption 3.2 (Query)**.** A query $\mathcal{Q}$ is *valid* only if $O \subseteq B$, $I \subset A \cup B$, with $O \cap I = \emptyset$, and every variable in V_A is either intervened upon, or redundant for the intervention.

More precisely, let G^I denote the *mutilated* graph after intervention $do(V_I = q_I)$, i.e. variables V_I no longer have any incoming arcs. Then, for $i \in A$, either $V_i \in V_I$, i.e. it is intervened upon, or there is no directed path from V_i to any variable in V_B in G^I. In the example discussed in Section 2, $V_I = \{X\}$ and $V_O = \{Y\}$.

The presence of unobserved confounders implies that the values for variables in V_B are not completely defined when $V_A = v_A$. However, as we note in Section 2, the unobserved confounder U_B effectively selects a particular function from all possible functions mapping V_A to V_B subject to the constraints imposed by the graph structure. For $j \in B$, let $pa(V_j)$ denote the parents of V_j in the causal graph. Analogous to sets $\mathcal{F}$ and $\mathcal{G}$ in Section 2, define $\mathcal{F}_j$ to be the set of all possible functions $pa(V_j) \mapsto V_j$. Since each variable $V_k \in pa(V_j)$ takes values in $\{0,1\}$ and $V_j \in \{0,1\}$, the cardinality of the set $|\mathcal{F}_j| = 2^{2^{|pa(V_j)|}}$. Then the elements of $\mathcal{F}_j$ can be indexed by $r_{V_j} \in R_{V_j} = \{1, \ldots, |\mathcal{F}_j|\}$. Let the set $R = \prod_{j \in B} R_{V_j}$ index all possible mappings from $pa(V_j) \mapsto V_j$ for all $j \in B$. Note that the cardinality $|R| = \prod_{j \in B} 2^{2^{|pa(V_j)|}}$.

For $r \in R$, let $F_O(V_S = v_S, r)$ denote the value of $V_O \subseteq V_B$ when $V_S = v_S$ provided it is well defined. Since V_A topologically precede V_B, setting $V_A = v_A$ and choosing $r \in R$ completely defines the values for V_B, i.e. $F_B(V_A = v_A, r)$ is well defined. Let $R_{v_B.v_A} = \{r : F_B(V_A = v_A, r) = v_B\}$. From the definition of a valid query, it follows that setting $V_I = q_I$ and selecting $r \in R$ uniquely defines the value for V_O. Let $R_\mathcal{Q} = \{r \in R : F_O(V_I = q_I, r) = q_O\}$ denote the set of r values which are consistent with the query.

Then bounds for the causal query can be obtained by solving the following pair of linear programs (Balke & Pearl, 1994):

$$\begin{array}{rl} \min_q / \max_q & \sum_{r \in R_\mathcal{Q}} q_r \\ \text{s.t.} & \sum_{r \in R_{v_B.v_A}} q_r = p_{v_B.v_A}, \ \forall v_A, v_B, \\ & q \geq 0. \end{array} \tag{8}$$

Recall for $V_A = v_A$, $r \in R$ uniquely determines the value of V_B. Hence, for fixed v_A, $\cup_{v_B} R_{v_B.v_A}$ is a partition of R. Thus, the constraint $\sum_{r \in R} q_r = \sum_{v_B} \sum_{r \in R_{v_B.v_A}} q_r = \sum_{v_B} p_{v_B.v_A} = 1$ is implied by the other constraints in the LP, and therefore, is not explicitly added to the LP.

3.1. Pruning the LP

Let $h : V_A \mapsto V_B$ denote any function from V_A to V_B. We will call these functions *hyperarcs* because they correspond to hyperarcs in appropriately defined hypergraphs, and also as a short hand for these special functions. Let $R_h = \{r : F_B(V_A = v_A, r) = h(v_A), \ \forall v_A \in \{0,1\}^{|A|}\}$ denote the set of r values which are consistent with the hyperarc h. The hyperarc h is valid only if $R_h \neq \emptyset$. Let H denote the set of valid hyperarcs and let $\{q_h : h \in H\}$ denote a probability measure over valid hyperarcs. Then the LP to compute the lower bound α_L in terms of variables $\{q_h\}$ is given by

$$\begin{array}{rl} \min_q & \sum_{h \in H} c^L_h q_h \\ \text{s.t.} & \sum_{\{h \in H : h(v_A) = v_B\}} q_h = p_{v_B.v_A}, \forall v_A, v_B \\ & q \geq 0, \end{array} \tag{9}$$

where $c^L_h = \mathbf{1}\{R_h \subseteq R_\mathcal{Q}\}$. Similarly, the LP for the upper bound α_U can be reformulated as:

$$\begin{array}{rl} \max_q & \sum_{h \in H} c^U_h q_h \\ \text{s.t.} & \sum_{\{h \in H : h(v_A) = v_B\}} q_h = p_{v_B.v_A}, \forall v_A, v_B \\ & q \geq 0 \end{array} \tag{10}$$

where $c^U_h = \max_{r \in R_h} \mathbf{1}\{r \in R_\mathcal{Q}\} = \mathbf{1}\{R_h \cap R_\mathcal{Q} \neq \emptyset\}$. Both reformulations have *exponentially* fewer variables; however, they are useful only if the set of valid hyperarcs H and the corresponding costs $c^L_h = \mathbf{1}\{R_h \subseteq R_\mathcal{Q}\}$ and $c^U_h = \mathbf{1}\{R_h \cap R_\mathcal{Q} \neq \emptyset\}$ can be efficiently computed, i.e. in particular, without iterating over R.

3.1.1. Characterizing Valid Hyperarcs

We now show how to efficiently check the validity of hyperarc h. We begin by defining functional consistency.

Definition 3.3 (Functional Consistency)**.** For $j \in B$, let $P_j \subseteq N$ denote the indices of $pa(V_j)$. A hyperarc h is *functionally consistent* if for all $a, b \in \{0,1\}^{|N|}$ such that $h(a_A) = a_B$ and $h(b_A) = b_B$, and for all $j \in B$,

$$a_{P_j} = b_{P_j} \implies a_j = b_j$$

A hyperarc h partially specifies a function mapping $pa(V_j)$ to V_j. Functional consistency ensures this partial specification is consistent with some binary function $pa(V_j) \to V_j$. The following result characterizes valid hyperarcs without producing a r-value which validates it.

Theorem 3.4 (Validity of h)**.** *A hyperarc h is valid if, and only if, h is functionally consistent.*

Proof. It is clear that if a hyperarc is valid, then it is functionally consistent. Suppose a hyperarc is functionally consistent. Then, for each $V_j \in V_B$, the partial specification of a mapping from $pa(V_j)$ to V_j implied by the hyperarc h is consistent with some binary function $pa(V_j) \to V_j$. Since, for each $V_j \in V_B$, r_{V_j} indexes the set of all possible functions $pa(V_j) \to V_j$, it follows that the set of r-values which are consistent with the mapping h i.e. R_h is non-empty. Equivalently h is valid. □

The first two columns of Table 1 compare $|R|$ with the maximum possible number of hyperarcs $2^{|B|2^{|A|}}$ for five different causal inference problems (details in Appendix). Note that the reduction in size can be several orders of magnitude, and it increases with the complexity of the causal

Graph	$\lvert R\rvert$	$2^{\lvert B\rvert^{2^{\lvert A\rvert}}}$	$\lvert H\rvert$
Ex A	1.3×10^8	1.0×10^6	2.3×10^3
Ex B	4.2×10^6	1.0×10^6	7.1×10^4
Ex C	4.2×10^6	1.0×10^6	4.4×10^4
Ex D	6.3×10^{57}	1.7×10^7	9.4×10^6
Ex E	3.2×10^{32}	1.7×10^7	9.4×10^6

Table 1: LP pruning

graph, see e.g. Examples D and E. Thus, there is a very significant reduction in size even if all hyperarcs are valid. The last column in Table 1 lists $|H|$. Considering only the valid hyperarcs further decreases the size of the LP by at least 1 order of magnitude, and sometimes more. The LPs corresponding to Examples B and C can be solved without pruning; however, the LP corresponding to Example A can only be solved after pruning the problem, and the LPs for Examples D and E are too large even after pruning. In Section 4 we propose a greedy heuristic to compute bounds for these problems.

Next, we show how to efficiently compute $c_h^L = \mathbf{1}\{R_h \subseteq R_{\mathcal{Q}}\}$ and $c_h^U = \mathbf{1}\{R_h \cap R_{\mathcal{Q}} \neq \emptyset\}$.

3.1.2. Efficiently computing c_h^L

Recall that in Section 2.2 we established that c_h^L for the simple example can be efficiently computed by checking whether h is completely consistent with $\mathcal{Q}$. Here, we extend that result to general causal graphs. Let $I_C \subseteq I$ denote the indices of variables that are *critical* in the intervention i.e. for each $V \in V_{I_C}$, there is a directed path from V to a variable in V_O in $G^{\mathcal{I}}$. Then the following result characterizes complete consistency of conditional probabilities.

Lemma 3.5 (Complete Consistency of Probability)**.** *For graphs satisfying Assumption 3.1 and queries satisfying Assumption 3.2, the conditional probability $p_{v_B.v_A}$ is completely consistent with $\mathcal{Q}$ if and only if, $v_{I_C\cap A} = q_{I_C\cap A}$, $v_{I_C\cap B} = q_{I_C\cap B}$, and $v_O = q_O$.*

This result is analogous to the one for the simple example: the conditional probability $p_{v_B.v_A}$ is completely consistent with $\mathcal{Q}$, if and only if, the variable assignments $V_A = v_A$ and $V_B = v_B$ are consistent with the query. Next, we show that a hyperarc h is completely consistent with the query $\mathcal{Q}$ if and only if, there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and the conditional probability $p_{v_B.v_A}$ is completely consistent with $\mathcal{Q}$.

Theorem 3.6 (Complete Consistency of Hyperarc)**.** *For graphs satisfying Assumption 3.1 and queries satisfying Assumption 3.2, a hyperarc h is completely consistent with $\mathcal{Q}$ if, and only if, there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, and the conditional probability $p_{v_B.v_A}$ is completely consistent with $\mathcal{Q}$.*

To summarize, to compute c_h^L , it is sufficient to check if h is completely consistent with $\mathcal{Q}$, which can be easily characterized without enumerating all the elements in R.

3.1.3. Efficiently computing c_h^U

We now show how to compute $c_h^U = \mathbf{1}\{R_h \cap R_{\mathcal{Q}} \neq \emptyset\}$ using an efficient algorithm for checking $R_h \cap R_{\mathcal{Q}} \neq \emptyset$. This condition motivates the following definition for strict inconsistency.

Definition 3.7 (Strict Inconsistency)**.** A hyperarc h is said to strictly inconsistent with the query $\mathcal{Q}$ if $R_h \cap R_{\mathcal{Q}} = \emptyset$. The conditional probability $p_{v_B.v_A}$ is strictly inconsistent with the query $\mathcal{Q}$ if $R_{v_B.v_A} \cap R_{\mathcal{Q}} = \emptyset$.

Hence, $c_h^U = 0$, if and only if, h is strictly inconsistent with $\mathcal{Q}$. The following result characterizes strict inconsistency for conditional probabilities.

Theorem 3.8 (Strict Inconsistency of Probability)**.** *For graphs satisfying Assumption 3.1 and queries satisfying Assumption 3.2, the conditional probability $p_{v_B.v_A}$ is strictly inconsistent with the query $\mathcal{Q}$, if and only if, $v_{I_C\cap A} = q_{I_C\cap A}$, $v_{I_C\cap B} = q_{I_C\cap B}$, and $v_O \neq q_O$.*

Thus, the conditional probability $p_{v_B.v_A}$ is strictly inconsistent with $\mathcal{Q}$ if and only if, the variable assignment $V_A = v_A, V_B = v_B$ is inconsistent with the query. Next, we show that a hyperarc h is strictly inconsistent with the query $\mathcal{Q}$ if, and only if, there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, and the conditional probability $p_{v_B.v_A}$ is strictly inconsistent with $\mathcal{Q}$.

Theorem 3.9 (Strict Inconsistency of Hyperarc)**.** *For graphs satisfying Assumption 3.1 and queries satisfying Assumption 3.2, the hyperarc h is strictly inconsistent with $\mathcal{Q}$ if, and only if, there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and the probability $p_{v_B.v_A}$ is strictly inconsistent with $\mathcal{Q}$.*

To summarize, to compute c_h^U, it is sufficient to check if h is strictly inconsistent with $\mathcal{Q}$, and that can be easily characterized without enumerating all the elements in R.

3.2. Bounds in Closed Form

Recall that for the class of problems identified by (Sachs et al., 2020), each variable in V_A is either in the intervention, or redundant for the intervention. Now we show that for a subclass of problems which satisfy Assumption 3.10 i.e. where the entire set A is involved in the intervention and all intervention variables are critical, the bounds can be computed in closed form.

Assumption 3.10. The query $\mathcal{Q} = \mathbb{P}(V_O = q_O \mid V_I = q_I)$ satisfies the following:

S1 $A \subseteq I$, i.e. the entire set A is involved in the intervention,

S2 there is a directed path from every $V \in V_I$ to some variable in V_O in $G^{\mathcal{I}}$, i.e. all intervention variables are critical, or equivalently $I_C = I$.

Theorem 3.11 (Lower Bound for Special Class of Problems)**.** *Suppose the query $\mathcal{Q}$ satisfies S1 and S2. Then the lower bound α_L is equal to the sum of the conditional probabilities that are completely consistent with $\mathcal{Q}$.*

Proof. Theorem 3.6 implies a hyperarc h is completely consistent with $\mathcal{Q}$ if, and only if, there exists $v = (v_A, v_B)$ such that $h(v_A) = v_B$, $v_{A\cap I_C} = q_{A\cap I_C}$, $v_{B\cap I_C} = q_{B\cap I_C}$, and $v_O = q_O$. Therefore, for any $\mathcal{Q}$ satisfying S1 and S2, it follows that a hyperarc h is completely consistent with $\mathcal{Q}$ if, and only if, there exists v_B such that $h(q_A) = v_B$, $v_{B\cap I} = q_{B\cap I}$, and $v_O = q_O$. Thus, it follows that

$$
\begin{aligned}
\alpha_L &= \sum_{\{h\in H: c_h^L=1\}} q_h \\
&= \sum_{\{h\in H:\, h \text{ is completely consistent with } \mathcal{Q}\}} q_h && (11)\\
&= \sum_{\{v_B: v_{I\cap B}=q_{I\cap B}, v_O=q_O\}} \sum_{\{h\in H: h(q_A)=v_B\}} q_h && (12)\\
&= \sum_{\{v_B: v_{I\cap B}=q_{I\cap B}, v_O=q_O\}} p_{v_B.q_A} && (13)
\end{aligned}
$$

where (11) follows from Definition 2.1, (12) from the discussion above, and (13) from the constraints of the pruned LP. By Lemma 3.5, (13) is the sum of the probabilities completely consistent with $\mathcal{Q}$. ☐

Theorem 3.12 (Upper Bound for Special Class of Problems)**.** *Suppose the query $\mathcal{Q}$ satisfies S1 and S2. Then the upper bound α_U is given by the difference of the sum of the conditional probabilities which are strictly inconsistent with $\mathcal{Q}$ and* 1.

Proof. Theorem 3.9 implies that a hyperarc h is strictly inconsistent with $\mathcal{Q}$ if, and only if, there exists $v = (v_A, v_B)$ such that $h(v_A) = v_B$, $v_{A\cap I_C} = q_{A\cap I_C}$, $v_{B\cap I_C} = q_{B\cap I_C}$, and $v_O \neq q_O$. Therefore, for any $\mathcal{Q}$ satisfying S1 and S2, it follows that a hyperarc h is strictly inconsistent with $\mathcal{Q}$ if, and only if, there exists v_B such that $h(q_A) = v_B$, $v_{B\cap I} = q_{B\cap I}$, and $v_O \neq q_O$. Thus, it follows that

$$
\begin{aligned}
&\sum_{h\in H: c_h^U=0} q_h \\
&= \sum_{\{h\in H:\, h \text{ is strictly inconsistent with } \mathcal{Q}\}} q_h && (14)\\
&= \sum_{\{v_B: v_{B\cap I}=q_{B\cap I}, v_O\neq q_O\}} \sum_{\{h\in H: h(q_A)=v_B\}} q_h && (15)\\
&= \sum_{\{v_B: v_{B\cap I}=q_{B\cap I}, v_O\neq q_O\}} p_{v_B.q_A} && (16)
\end{aligned}
$$

where (14) follows from Definition 3.7, (15) from the discussion above, and (16) from the constraints of the pruned LP. The result follows by the fact that $\alpha_U = \sum_{h\in H: c_h^U=1} q_h = 1 - \sum_{h\in H: c_h^U=0} q_h$. ☐

3.2.1. Example

An important example of the class of problems that satisfy S1 and S2 is a causal inference problem where multiple confounded treatments influence an outcome (Ranganath & Perotte, 2019; Janzing & Schölkopf, 2018; D'Amour, 2019; Tran & Blei, 2017). For example, the setting where the treatments are medications and procedures, and the outcome is the progression of the disease in the patient. In this case, there are many confounders which influence both the prescribed treatments and outcome. Some of these confounders can be measured, e.g. the pre-existing conditions of the patient, and others are unobserved, e.g. the treatment preferences of the attending doctor (Wang & Blei, 2019a). See Figure 3 for the causal graph of the patient response where

- C_i indicates the presence of pre-existing condition i in the patient
- U_A is an unobserved confounder (e.g. a patient characteristic) which influences the presence of pre-existing conditions
- T_i, $i = 1, \ldots, 5$, indicates whether the patient was prescribed treatment i
- U_B is an unobserved confounder which influences both the prescribed treatments and outcome (e.g. doctor biases, treatment preferences)
- Y indicates the progression of the disease in the patient. $Y = 0$ indicates that the disease is mild, and $Y = 1$ that the disease is lethal.

Wang & Blei (2019a; 2021) introduced the *deconfounder* as a method to predict the expected value of the outcome

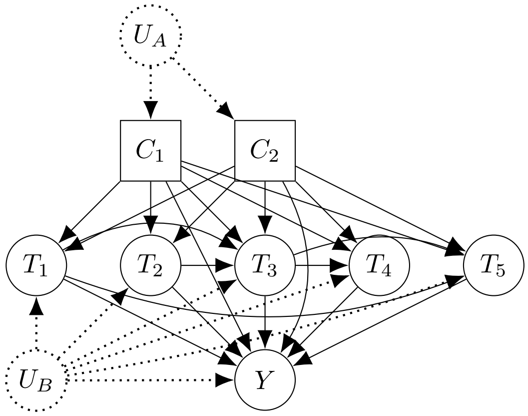


Figure 3: Example of G where $V_A = \{C_1, C_2\}$ and $V_B = \{T_1, T_2, T_3, T_4, T_5, Y\}$

variable under treatment interventions when multiple confounded treatments influence an outcome. One of the limitations of the deconfounder approach is that it cannot be applied in a setting where the treatment variables have causal relationships between them (Ogburn et al., 2019; Wang & Blei, 2019b; Imai & Jiang, 2019). For example, in our context, the side effects of one treatment can influence the prescription of another treatment (Ogburn et al., 2019), as implied by arrows $T_1 \to T_3, T_2 \to T_3, T_3 \to T_4$ and $T_3 \to T_5$ in Figure 3. The deconfounder cannot be used for inference in this setting. However, since the entire set $V_A = \{C_1, C_2\}$ is involved in the intervention, and every node intervened upon is a parent of Y, Theorems (3.11) and (3.12) can be used to compute bounds for the query $\mathbb{E}[Y|do(\mathbf{T} = t, C_1 = c_1, C_2 = c_2)]$ in closed form. In particular,

$$\begin{aligned}\alpha_L &= \mathbb{P}(\mathbf{T} = t, Y = 1 | C_1 = c_1, C_2 = c_2) \\ \alpha_U &= 1 - \mathbb{P}(\mathbf{T} = t, Y = 0 | C_1 = c_1, C_2 = c_2)\end{aligned}$$

4. Greedy Heuristic

For problems where even the pruned LPs are intractable, and the conditions in Section 3.2 are not satisfied, we propose Algorithm 1 as a greedy heuristic to compute the bounds α_L and α_U. This heuristic is motivated by the duals of LPs (9) and (10), which are given by:

$$\begin{aligned}\alpha_L = \quad \max_\lambda \quad & \textstyle\sum_{(v_A, v_B) \in \{0,1\}^{|A|} \times \{0,1\}^{|B|}} p_{v_B . v_A} \lambda_{v_B . v_A} \\ \text{s.t.} \quad & \textstyle\sum_{v_A \in \{0,1\}^{|A|}} \lambda_{h(v_A) . v_A} \leq c_h^L\end{aligned} \tag{17}$$

$$\begin{aligned}\alpha_U = \quad \min_\lambda \quad & \textstyle\sum_{(v_A, v_B) \in \{0,1\}^{|A|} \times \{0,1\}^{|B|}} p_{v_B . v_A} \lambda_{v_B . v_A} \\ \text{s.t.} \quad & \textstyle\sum_{v_A \in \{0,1\}^{|A|}} \lambda_{h(v_A) . v_A} \geq c_h^U\end{aligned} \tag{18}$$

We utilize the fact that in our numerical experiments, we observed that for both dual LPs, there was always an optimal solution that only took values in the set $\{-1, 0, 1\}$, and the fact that in the symbolic bounds introduced by Balke & Pearl (1994) (see, also (Zhang & Bareinboim, 2017; Pearl, 2009; Sjölander et al., 2014; Sachs et al., 2020)) the constraints were combined using coefficients taking values in $\{-1, 0, 1\}$. In fact, we believe that the following conjecture is true.

Conjecture 4.1 (Dual Integrality)**.** *The dual LPs in (17) and (18) have optimal solutions which only take values in* $\{-1, 0, 1\}$.

Graph	% with $\alpha_L^G = \alpha_L$	% with $\alpha_U^G = \alpha_U$	% with $\epsilon_L \leq 0.1$	% with $\epsilon_U \leq 0.1$
Ex A	100	100	100	100
Ex B	99	86	99	94
Ex C	100	84	100	86

Table 2: Greedy Solution vs Optimal Solution

Algorithm 1 Greedy Heuristic

Let the permutation which sorts the conditional probabilities in descending order be $i_1, \ldots, i_{2^{|N|}}$.

Function `GreedyLowerBound():`
  Initialize $\lambda = -\mathbb{1}$.
  for $j = 1, .., 2^{|N|}$ **do**
    while λ *is feasible* **do**
      $\lambda_{i_j} = \lambda_{i_j} + 1$

Function `GreedyUpperBound():`
  Initialize $\lambda = \mathbb{1}$.
  for $j = 1, .., 2^{|N|}$ **do**
    while λ *is feasible* **do**
      $\lambda_{i_j} = \lambda_{i_j} - 1$

We benchmark the greedy heuristic by computing bounds for 100 instances of each of the examples in Appendix A which do not satisfy the conditions in Section 3.2. Table 2 reports the results for Examples A, B and C for which the LP can be solved. We see that bounds from the greedy heuristic matches the LP bounds in most instances for these problems. Recall that one can compute the optimal bounds for Example A only after pruning the LP. In Table 2, $\epsilon_L = 1 - \frac{\alpha_L^G}{\alpha_L}$ and $\epsilon_U = \frac{\alpha_U^G}{\alpha_U} - 1$ denote the relative errors of the lower and upper bounds, respectively. We see that the lower bound is always within 10% of the true value, whereas the upper bound is within 10% for at least 86% of the cases. See Appendix for the empirical distribution function of errors.

Furthermore, the greedy heuristic yields non-trivial bounds for Examples D and E, where the corresponding pruned LP is too large to be solved to optimality (see Table 3).

Graph	α_L^G	α_U^G
Ex D	0.92	0.93
Ex E	0.87	0.98

Table 3: Greedy Solution for Large Problems

5. Conclusion

In this work, we show how to leverage structural properties of the LPs corresponding to causal inference problems to significantly reduce their size. We also show how to construct these LPs efficiently. As a direct consequence of our results, bounds for causal queries can be computed for graphs of much larger size. We show that there are examples of causal inference problems for which bounds could be computed only after the pruning we introduce. Our structural results also allow us to characterize a set of causal inference problems for which the bounds can be computed in closed form. This class includes as a special case extensions of problems considered in the multiple causes literature (Wang & Blei, 2021; Ranganath & Perotte, 2019; Janzing & Schölkopf, 2018; D'Amour, 2019; Tran & Blei, 2017).

We are currently considering two extensions. The constraints in the dual LPs are all packing constraints (for α_L) or covering constraints (for α_U); however, the variables are free. However, if Conjecture 4.1 is true, the feasible set of the LP can be assumed to be bounded. This could potentially be used to compute fast approximation algorithms (Bienstock & Iyengar, 2006).

Note that here we do not allow the query to contain observations about the unit under consideration. We can show that bounds for queries containing observations can be computed by solving *fractional* LPs (Bitran & Novaes, 1973). These fractional LPs are special because the denominator is restricted to be non-negative. This allows us to homogenize the problem into a LP with one additional constraint. Therefore, we expect that the structural results presented here will extend to the setting where the query contains an observation.

References

Balke, A. and Pearl, J. Counterfactual probabilities: Computational methods, bounds and applications. In *Uncertainty Proceedings 1994*, pp. 46–54. Elsevier, 1994.

Bienstock, D. and Iyengar, G. Approximating fractional packings and coverings in o (1/epsilon) iterations. *SIAM Journal on Computing*, 35(4):825–854, 2006.

Bitran, G. R. and Novaes, A. G. Linear programming with a fractional objective function. *Operations Research*, 21 (1):22–29, 1973.

D'Amour, A. On multi-cause approaches to causal inference with unobserved counfounding: Two cautionary failure cases and a promising alternative. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 3478–3486. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/d-amour19a.html.

Duarte, G., Finkelstein, N., Knox, D., Mummolo, J., and Shpitser, I. An automated approach to causal inference in discrete settings. *arXiv preprint arXiv:2109.13471*, 2021.

Evans, R. J. Graphical methods for inequality constraints in marginalized DAGs. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, 2012. doi: 10.1109/MLSP.2012.6349796.

Finkelstein, N. and Shpitser, I. Deriving bounds and inequality constraints using logical relations among counterfactuals. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1348–1357. PMLR, 2020.

Finkelstein, N., Adams, R., Saria, S., and Shpitser, I. Partial identifiability in discrete data with measurement error. In de Campos, C. and Maathuis, M. H. (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1798–1808. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/finkelstein21b.html.

Geiger, D. and Meek, C. Quantifier elimination for statistical problems, 2013. URL https://arxiv.org/abs/1301.6698.

Imai, K. and Jiang, Z. Discussion of "the blessings of multiple causes" by wang and blei, 2019.

Imbens, G. W. and Rubin, D. B. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Janzing, D. and Schölkopf, B. Detecting confounding in multivariate linear models via spectral analysis. *Journal of Causal Inference*, 6(1):20170013, 2018. doi: doi:10.1515/jci-2017-0013. URL https://doi.org/10.1515/jci-2017-0013.

Kilbertus, N., Kusner, M. J., and Silva, R. A class of algorithms for general instrumental variable models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20108–20119. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/

e8b1cbd05f6e6a358a81dee52493dd06-Paper.pdf.

Ogburn, E. L., Shpitser, I., and Tchetgen, E. J. T. Comment on "blessings of multiple causes". *Journal of the American Statistical Association*, 114(528):1611–1615, 2019. doi: 10.1080/01621459.2019.1689139. URL https://doi.org/10.1080/01621459.2019.1689139.

Pearl, J. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Poderini, D., Chaves, R., Agresti, I., Carvacho, G., and Sciarrino, F. Exclusivity graph approach to instrumental inequalities. In Adams, R. P. and Gogate, V. (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 1274–1283. PMLR, 22–25 Jul 2020. URL https://proceedings.mlr.press/v115/poderini20a.html.

Ranganath, R. and Perotte, A. Multiple causal inference with latent confounding, 2019.

Richardson, A., Hudgens, M. G., Gilbert, P., and Fine, J. Nonparametric bounds and sensitivity analysis of treatment effects. *Stat Sci*, 29(4):596–618, 2014. doi: 10.1214/14-STS499.

Sachs, M. C., Gabriel, E. E., and Sjolander, A. Symbolic computation of tight causal bounds. *Biometrika*, 103(1): 1–19, 2020.

Sachs, M. C., Jonzon, G., Sjölander, A., and Gabriel, E. E. A general method for deriving tight symbolic bounds on causal effects, 2021.

Sjölander, A., Lee, W., Källberg, H., and Pawitan, Y. Bounds on causal interactions for binary outcomes. *Biometrics*, 70(3):500–505, 2014. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/24538083.

Tran, D. and Blei, D. M. Implicit causal models for genome-wide association studies, 2017.

Wang, Y. and Blei, D. A proxy variable view of shared confounding. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10697–10707. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wang21c.html.

Wang, Y. and Blei, D. M. The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528): 1574–1596, 2019a.

Wang, Y. and Blei, D. M. The blessings of multiple causes: Rejoinder. *Journal of the American Statistical Association*, 114(528):1616–1619, 2019b. doi: 10.1080/01621459.2019.1690841. URL https://doi.org/10.1080/01621459.2019.1690841.

Zhang, J. and Bareinboim, E. Transfer learning in multi-armed bandits: A causal approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 1340–1346, 2017. doi: 10.24963/ijcai.2017/186. URL https://doi.org/10.24963/ijcai.2017/186.

Zhang, J. and Bareinboim, E. Bounding causal effects on continuous outcome. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):12207–12215, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17449.

A. Examples of Causal Inference Problems

In this section, we report the causal graph structure and the data generation process for the 5 examples in Table 1.

Example A

The causal graph for this example is display in Figure 4a. The query is: $P(Y=1|do(X_2=1, Z_1=1, Z_2=1))$, and data generating process used to generate the input data is given by

$$
\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
Z_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
Z_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
S_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B)) \\
X_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + S_1)) \\
S_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(S_1 + U_B + X_1 + Z_2)) \\
X_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(S_2 + U_B + Z_1 + Z_2)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + S_2 + X_2 + Z_2))
\end{aligned}
$$

After sampling U_A, U_B, Z_1, Z_2 we compute

$$\mathbb{P}(Y, X_2, S_2, X_1, S_1|Z_2, Z_1) = \mathbb{P}(Y|U_B, S_2, X_2, Z_2)\mathbb{P}(X_2|S_2, Z_1, Z_2, U_B)\mathbb{P}(S_2|Z_2, U_B, S_1, X_1)\mathbb{P}(X_1|S_1, U_B)\mathbb{P}(S_1|U_B)$$

that gives input distribution.

Example B

The causal graph for this example is in Figure 4b and the query is: $P(Y=1|do(A=1, B=1, C=1, F=1))$. The data generating process used to generate the input information is as follows:

$$
\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
C &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
F &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
A &\sim \text{Bernoulli}(\text{logit}^{-1}(C + F + U_B)) \\
B &\sim \text{Bernoulli}(\text{logit}^{-1}(C + F + U_B)) \\
D &\sim \text{Bernoulli}(\text{logit}^{-1}(A + U_B)) \\
E &\sim \text{Bernoulli}(\text{logit}^{-1}(A + B + U_B)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + D + C + E))
\end{aligned}
$$

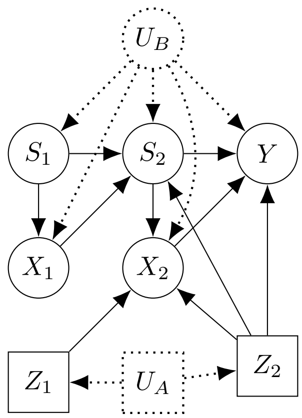
After sampling U_A, U_B, C, F we compute

$$\mathbb{P}(A, B, D, E, Y|C, F) = \mathbb{P}(Y|U_B, D, C, E)\mathbb{P}(E|A, B, U_B)\mathbb{P}(D|A, U_B)\mathbb{P}(B|C, F, U_B)\mathbb{P}(A|C, F, U_B).$$

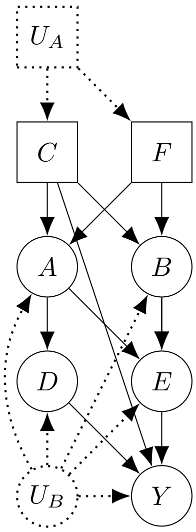
that gives the input distribution.

Example C

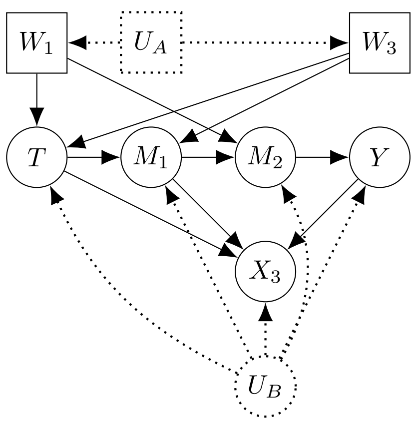
The causal graph for this example is in Figure 4c and the query is: $P(Y=1|do(M_1=1, W_1=1, W_3=1))$. The data generating process used to generate the input information is as follows:



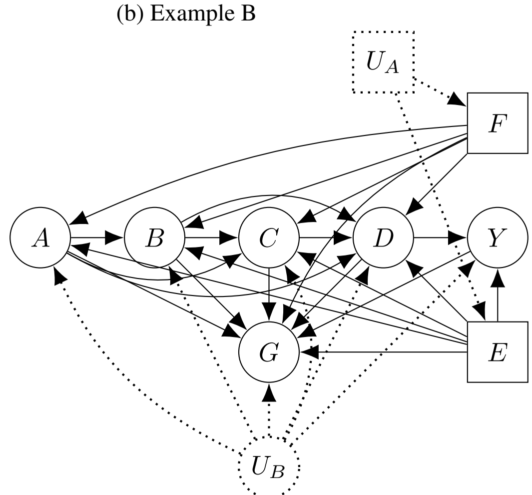
(a) Example A



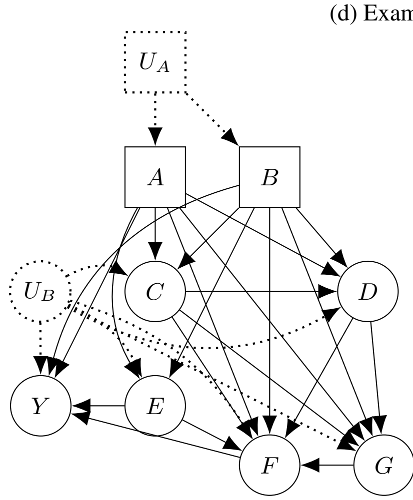
(b) Example B



(c) Example C



(d) Example D



(e) Example E

Figure 4: Examples of Causal Inference Problems

$$
\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
W_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
W_3 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
T &\sim \text{Bernoulli}(\text{logit}^{-1}(W_1 + W_3 + U_B)) \\
M_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(T + W_3 + U_B)) \\
M_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(M_1 + W_1 + U_B)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(M_2 + U_B)) \\
X_3 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + Y + M_1 + T))
\end{aligned}
$$

After sampling U_A, U_B, W_1, W_3 we compute

$\mathbb{P}(T, M_1, M_2, X_3, Y|W_1, W_3) = \mathbb{P}(X_3|U_B, Y, M_1, T)\mathbb{P}(Y|M_2, U_B)\mathbb{P}(M_2|M_1, W_1, U_B)$
$\mathbb{P}(M_1|T, W_3, U_B)\mathbb{P}(T|W_1, W_3, U_B)$

that gives the input distribution.

Example D

The causal graph for this example is in Figure 4d and the query is: $P(Y = 1|do(D = 1, E = 1, F = 1))$. The data generating process used to generate the input information is as follows:

$$
\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
F &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
E &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
A &\sim \text{Bernoulli}(\text{logit}^{-1}(F + E + U_B)) \\
B &\sim \text{Bernoulli}(\text{logit}^{-1}(E + F + A + U_B)) \\
C &\sim \text{Bernoulli}(\text{logit}^{-1}(B + E + F + A + U_B)) \\
D &\sim \text{Bernoulli}(\text{logit}^{-1}(B + E + F + A + C + U_B)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(E + D + U_B)) \\
G &\sim \text{Bernoulli}(\text{logit}^{-1}(A + B + C + D + Y + E + F + U_B))
\end{aligned}
$$

After sampling U_A, U_B, E, F we compute

$\mathbb{P}(G, Y, D, C, B, A|E, F) = \mathbb{P}(G|U_B, A, B, C, D, Y, E, F)\mathbb{P}(Y|E, D, U_B)\mathbb{P}(D|B, E, F, A, C, U_B)$
$\mathbb{P}(C|B, E, F, A, U_B)\mathbb{P}(B|E, F, A, U_B)\mathbb{P}(A|E, F, U_B)$

that gives the input distribution.

Example E

The causal graph for this example is in Figure 4e and the query is: $P(Y = 1|do(A = 1, B = 1, C = 1, F = 1))$. The data generating process used to generate the input information is as follows:

$$\begin{aligned}
U_A &\sim N(0,1)\\
U_B &\sim N(0,1)\\
A &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A))\\
B &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A))\\
C &\sim \text{Bernoulli}(\text{logit}^{-1}(A+B+U_B))\\
D &\sim \text{Bernoulli}(\text{logit}^{-1}(A+C+B+U_B))\\
E &\sim \text{Bernoulli}(\text{logit}^{-1}(A+B+U_B))\\
F &\sim \text{Bernoulli}(\text{logit}^{-1}(A+C+B+D+E+G+U_B))\\
G &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B+A+B+C+D))\\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B+A+E+B+F))
\end{aligned}$$

After sampling U_A, U_B, A, B we compute

$$\mathbb{P}(C, D, E, F, G, Y|A, B) = \mathbb{P}(C|A, B, U_B)\mathbb{P}(D|A, C, B, U_B)\mathbb{P}(E|A, B, U_B)\mathbb{P}(F|A, C, B, D, E, G, U_B)$$

$$\mathbb{P}(G|A, B, C, D, U_B)\mathbb{P}(Y|U_B, A, E, B, F)$$

that gives the input distribution.

B. Proof of results

Lemma B.1 (Complete Consistency of Probability)**.** *For graphs satisfying Assumption 3.1 and queries satisfying Assumption 3.2, the conditional probability $p_{v_B.v_A}$ is completely consistent with $\mathcal{Q}$ if and only if, $v_{I_C \cap A} = q_{I_C \cap A}$, $v_{I_C \cap B} = q_{I_C \cap B}$, and $v_O = q_O$.*

Proof. Suppose $p_{v_B.v_A}$ satisfies $v_{I_C \cap A} = q_{I_C \cap A}$, $v_{I_C \cap B} = q_{I_C \cap B}$ and $v_O = q_O$. Then, every $r \in R_{v_B.v_A}$ maps $V_{I_C} = q_{I_C}$ to $V_O = q_O$, and thus, $r \in R_\mathcal{Q}$ i.e. $R_{v_B.v_A} \subseteq R_\mathcal{Q}$. Hence, $p_{v_B.v_A}$ is completely consistent with $\mathcal{Q}$. To see that this is the *only* form a completely consistent probability can take, consider two cases:

- Suppose either $v_{I_C \cap A} \neq q_{A \cap I_C}$ or $v_{I_C \cap B} \neq q_{B \cap I_C}$ i.e. the value of V_{I_C} in $p_{v_B.v_A}$ is not q_{I_C}. Since R is exhaustive, there exists r such that:
  - r maps $V_A = v_A$ to $V_B = v_B$ i.e. $r \in R_{v_B.v_A}$. In particular, r maps $V_{I_C} \neq q_{I_C}$ to $V_O = q_O$.
  - r maps $V_{I_C} = q_{I_C}$ to $V_O \neq q_O$ i.e. $r \notin R_Q$

  Hence, there exists r such that $r \in R_{v_B.v_A}$, but $r \notin R_\mathcal{Q}$ i.e. $p_{v_B.v_A}$ is not completely consistent.
- Suppose $v_O \neq q_O$. Then every $r \in R_{v_B.v_A}$ maps $V_{I_C} = q_{I_C}$ to $V_O \neq q_O$, and therefore, $r \notin R_\mathcal{Q}$. Hence $p_{v_B.v_A}$ is not completely consistent.

□

Theorem 3.6 (Complete Consistency of Hyperarc)**.** *For graphs satisfying Assumption 3.1 and queries satisfying Assumption 3.2, a hyperarc h is completely consistent with $\mathcal{Q}$ if, and only if, there exists $v \in \{0, 1\}^{|N|}$ such that $h(v_A) = v_B$, and the conditional probability $p_{v_B.v_A}$ is completely consistent with $\mathcal{Q}$.*

Proof. Suppose there exists $v \in \{0, 1\}^{|N|}$ such that $h(v_A) = v_B$, and $p_{v_B.v_A}$ is completely consistent with $\mathcal{Q}$. Since $R_h \subseteq R_{v_B.v_A}$, and $p_{v_B.v_A}$ is completely consistent with $\mathcal{Q}$, $R_h \subseteq R_\mathcal{Q}$ i.e. h is completely consistent with $\mathcal{Q}$.

Suppose h is completely consistent with $\mathcal{Q}$, but there does not exist $v \in \{0,1\}^{|N|}$ which satisfies Theorem 3.6. That is, for all $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and $v_{I_C \cap A} = q_{I_C \cap A}$, we have either $v_{I_C \cap B} \neq q_{I_C \cap B}$ or $v_O \neq q_O$. Since R is exhaustive, there exists $r \in R$ such that:

(i) r maps $V_A = v_A$ to $V_B = v_B$ for all (v_A, v_B) such that $h(v_A) = v_B$ i.e. $r \in R_h$

(ii) r maps $V_{I_C \cap A} = q_{I_C \cap A}, V_{I_C \cap B} = q_{I_C \cap B}$ to $V_O \neq q_O$.

Hence h is not completely consistent with $\mathcal{Q}$, a contradiction. □

Theorem 3.8 (Strict Inconsistency of Probability). *For graphs satisfying Assumption 3.1 and queries satisfying Assumption 3.2, the conditional probability $p_{v_B.v_A}$ is strictly inconsistent with the query $\mathcal{Q}$, if and only if, $v_{I_C \cap A} = q_{I_C \cap A}$, $v_{I_C \cap B} = q_{I_C \cap B}$, and $v_O \neq q_O$.*

Proof. Suppose $p_{v_B.v_A}$ satisfies $v_{I_C \cap A} = q_{I_C \cap A}, v_{I_C \cap B} = q_{I_C \cap B}$ and $v_O \neq q_O$. Then, any $r \in R_{v_B.v_A}$ maps $V_{I_C} = q_{I_C}$ to $V_O \neq q_O$, and thus, $r \notin R_{\mathcal{Q}}$ i.e. $R_{v_B.v_A} \cap R_{\mathcal{Q}} = \emptyset$. Hence, $p_{v_B.v_A}$ is strictly inconsistent with $\mathcal{Q}$. To see that this is the *only* form a strictly inconsistent probability can take, consider two cases:

- Suppose either $v_{I_C \cap A} \neq q_{A \cap I_C}$ or $v_{I_C \cap B} \neq q_{B \cap I_C}$ i.e. the value of V_{I_C} in $p_{v_B.v_A}$ is not q_{I_C}. Since R is exhaustive, there exists r such that:
  - r maps $V_A = v_A$ to $V_B = v_B$ i.e. $r \in R_{v_B.v_A}$. In particular, r maps $V_{I_C} \neq q_{I_C}$ to $V_O = q_O$.
  - r maps $V_{I_C} = q_{I_C}$ to $V_O = q_O$ i.e. $r \in R_{\mathcal{Q}}$

  Hence, there exists r such that $r \in R_{v_B.v_A}$, but $r \in R_{\mathcal{Q}}$ i.e. $p_{v_B.v_A}$ is not strictly inconsistent.
- Suppose $v_O = q_O$. Then every $r \in R_{v_B.v_A}$ maps $V_{I_C} = q_{I_C}$ to $V_O = q_O$, and therefore, $r \in R_{\mathcal{Q}}$. Hence $p_{v_B.v_A}$ is not strictly inconsistent.

□

Theorem 3.9 (Strict Inconsistency of Hyperarc). *For graphs satisfying Assumption 3.1 and queries satisfying Assumption 3.2, the hyperarc h is strictly inconsistent with $\mathcal{Q}$ if, and only if, there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and the probability $p_{v_B.v_A}$ is strictly inconsistent with $\mathcal{Q}$.*

Proof. Suppose there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, and $p_{v_B.v_A}$ is strictly inconsistent with $\mathcal{Q}$. Since $R_h \subseteq R_{v_B.v_A}$, and $p_{v_B.v_A}$ is strictly inconsistent with $\mathcal{Q}$, $R_h \cap R_{\mathcal{Q}} = \emptyset$ i.e. h is strictly inconsistent with $\mathcal{Q}$.

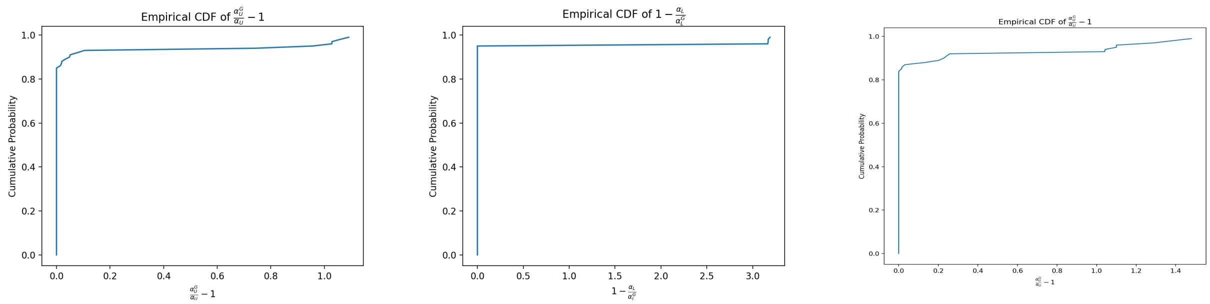
Suppose h is strictly inconsistent with $\mathcal{Q}$, but there does not exist $v \in \{0,1\}^{|N|}$ which satisfies Theorem 3.9. That is, for all $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and $v_{I_C \cap A} = q_{I_C \cap A}$, we have either $v_{I_C \cap B} \neq q_{I_C \cap B}$ or $v_O = q_O$. Since R is exhaustive, there exists $r \in R$ such that:

(i) r maps $V_A = v_A$ to $V_B = v_B$ for all (v_A, v_B) such that $h(v_A) = v_B$ i.e. $r \in R_h$

(ii) r maps $V_{I_C \cap A} = q_{I_C \cap A}, V_{I_C \cap B} = q_{I_C \cap B}$ to $V_O = q_O$.

Hence h is not strictly inconsistent with $\mathcal{Q}$, a contradiction. □

C. Empirical CDF for Error of Greedy Heuristic



(a) Empirical CDF of the Relative Error of α_U for Example B

(b) Empirical CDF for Relative Error of α_L for Example B

(c) Empirical CDF of the Relative Error of α_U for Example C

Figure 5: Empirical Distribution Functions of Errors for Examples