Journal of Machine Learning Research 18 (2018) 1-23 Submitted 9/17; Revised 5/18; Published 6/18

Enhancing Identification of Causal Effects by Pruning

Santtu Tikka SANTTU.TIKKA@JYU.FI
Juha Karvanen JUHA.T.KARVANEN@JYU.FI
Department of Mathematics and Statistics
P.O.Box 35 (MaD) FI-40014 University of Jyvaskyla, Finland

Editor: Peter Spirtes

Abstract

Causal models communicate our assumptions about causes and effects in real-world phenomena. Often the interest lies in the identification of the effect of an action which means deriving an expression from the observed probability distribution for the interventional distribution resulting from the action. In many cases an identifiability algorithm may return a complicated expression that contains variables that are in fact unnecessary. In practice this can lead to additional computational burden and increased bias or inefficiency of estimates when dealing with measurement error or missing data. We present graphical criteria to detect variables which are redundant in identifying causal effects. We also provide an improved version of a well-known identifiability algorithm that implements these criteria.

Keywords: causal inference, identifiability, causal model, pruning, algorithm

1. Introduction

A formal framework for causal inference is provided by the probabilistic causal model (Pearl, 2009) that encodes our knowledge of the variables of interest and their mutual relationships. In observational studies experimentation is not available, but through the causal model framework we can still symbolically intervene on variables, forcing them to take certain values as if an experiment had taken place. The question is whether we can make inferences about the effect of the intervention in the post-intervention model using only the observed probability distribution of the variables in the model before the intervention took place. This question is formally defined as identifiability of causal effects, and it has received considerable attention in literature, including a number of algorithmic solutions (Huang and Valtorta, 2006; Shpitser and Pearl, 2006; Tian and Pearl, 2002).

A causal model can be associated with a directed acyclic graph (DAG) that represents the functional relationships of the variables included in the model. The graphical representation provides us with the concept of d-separation (Geiger et al., 1990), that can be used to infer conditional independences between variables from the graph. If the distribution of the variables implies no conditional independence statements other than those already encoded in the graph, we say that the distribution is faithful (Spirtes et al., 2000).

The use of d-separation in the post-intervention model is the basis of do-calculus (Pearl, 1995), which consists of a set of inference rules for manipulating interventional distributions. The purpose of do-calculus is to derive formulas for causal effects and other causal queries, and it has been shown to be complete with respect to the identifiability of causal effects

(Huang and Valtorta, 2006; Shpitser and Pearl, 2006). The derived formulas provide recipes for estimating the causal effects from observational data.

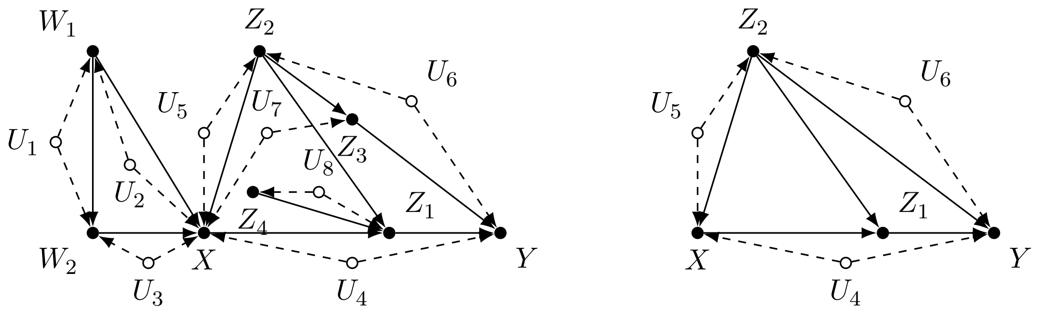
When computing causal effect formulas, we often apply an identifiability algorithm, such as the ID algorithm by Shpitser and Pearl (2006). Criteria for identifiability such as the back-door criterion and front-door criterion are available for manual derivations (Pearl, 2009) but the ID algorithm is more general and thus more suitable for automated processing. The ID algorithm splits the original problem into smaller subproblems which are then solved and aggregated as the final expression for the causal effect.

Complicated expressions are likely to arise in situations where we have included variables in our model that do not provide further benefit for the identification of the causal effect of interest. It is often the case that these variables nonetheless appear in the resulting formula, and deriving a simpler expression with the variable eliminated can be non-trivial. It is hard to specify what makes one expression simpler than another, but we can consider a number of criteria to evaluate simplicity. For example, we can compare the number of sums and fractions and the number of variables present in the expression.

In this paper we propose a number of graphical criteria to infer which variables in our causal model are in fact not necessary for identification. These criteria allow us to prune the graph, which in practice means removing specific vertices and considering identification in a latent projection. A significantly simpler expression can be obtained by pruning alone, but we may also combine pruning with simplification procedures that operate symbolically on the interventional distribution as presented in (Tikka and Karvanen, 2017b). Applying these methods in conjunction often provides additional benefits.

We present an identifiability algorithm that is able to recognize and eliminate unnecessary variables from the graph based on our criteria resulting in a simpler expression. When a large number of graphs and identifiability queries are processed, evaluating simpler expressions has apparent computational benefits. First, it is more efficient to evaluate a simpler expression repeatedly especially when some variables have been completely removed which further reduces the complexity of the task. Second, in practical applications that involve real-world data, variables often contain missing data or are affected by bias. Obtaining expression that do not involve such variables can be of great benefit in estimation. Third, a simpler expression is easier to communicate.

An introductory example motivates the use of the improved algorithm. We are interested in the causal effect of X on Y in graph G of Figure 1(a). Here, open circles denote unobserved variables. A more in-depth overview of graph theoretic concepts used in this paper is provided in Section 2. The causal effect is identifiable and the output of the ID algorithm is



(a) Graph G.

(b) Graph G after pruning.

Figure 1: Graph G before and after pruning for the introductory example.

$$\begin{aligned}
\sum_{z_2,z_4,z_3,z_1} &\left(\sum_{w_1,w_2,x'} P(y|w_1,z_2,z_4,w_2,z_3,x',z_1)P(x'|w_1,z_2,z_4,w_2,z_3) \times \right. \\
&\left. P(z_3|w_1,z_2,z_4,w_2)P(w_2|w_1,z_2,z_4)P(z_2|w_1)P(w_1) \right) \Big/ \\
&\left(\sum_{w_1,w_2,x',y'} P(y'|w_1,z_2,z_4,w_2,z_3,x',z_1)P(x'|w_1,z_2,z_4,w_2,z_3) \times \right. \\
&\left. P(z_3|w_1,z_2,z_4,w_2)P(w_2|w_1,z_2,z_4)P(z_2|w_1)P(w_1) \right) \times \\
&\left(\sum_{w_1,w_2,z_3',x',y'} P(y'|w_1,z_2,z_4,w_2,z_3',x',z_1)P(x'|w_1,z_2,z_4,w_2,z_3') \times \right. \\
&\left. P(z_3'|w_1,z_2,z_4,w_2)P(w_2|w_1,z_2,z_4)P(z_2|w_1)P(w_1) \right) \times \\
&P(z_1|w_1,z_2,z_4,w_2,x)P(z_3|z_2)P(z_4).
\end{aligned}$$

This expression is very cumbersome and complicated. However, it turns out that a simpler expression exists for the causal effect. By exploiting the structure of the graph and using standard probability calculus the following expression can be obtained

$$\sum_{z_2,z_1} \left(\sum_{x'} P(y|z_2,z_1,x')P(x'|z_2)P(z_2) \right) P(z_1|z_2,x).$$

This expression is simpler in every regard compared to the original output. It contains fewer terms and no fractions. Also, we have completely removed the variables w_1, w_2 and z_4 from the expression. It can be shown that identifying the causal effect in the original graph is equivalent to identifying it in the graph depicted in Figure 1(b). By running our improved algorithm we are able to prune the original graph and obtain this simpler expression directly. The algorithm works recursively and the pruning is carried out at each stage of the recursion. The recursive pruning provides significant benefits over pruning as a pre-processing step as demonstrated later.

The paper is structured as follows. In Section 2 we review crucial definitions and concepts related to graph theory and causal models. In Section 3 we focus on semi-Markovian causal models and present the original formulation of the ID algorithm. Our main results are presented in Section 4 and they are implemented into an improved identifiability algorithm in Section 5. Examples on the benefits of recursive pruning are provided in Section 6. Section 7 concludes with a discussion.

2. Definitions

We assume the reader to be familiar with a number of graph theoretic concepts and refer them to works such as (Koller and Friedman, 2009). We use capital letters to denote vertices and the respective variables, and small letters to denote their values. Bold letters are used to denote sets. A directed graph with a vertex set $\mathbf{V}$ and an edge set $\mathbf{E}$ is denoted by $\langle \mathbf{V}, \mathbf{E} \rangle$. For a graph $G = \langle \mathbf{V}, \mathbf{E} \rangle$ and a set of vertices $\mathbf{W} \subseteq \mathbf{V}$ the sets $\mathrm{Pa}(\mathbf{W})_G, \mathrm{Ch}(\mathbf{W})_G, \mathrm{An}(\mathbf{W})_G$ and $\mathrm{De}(\mathbf{W})_G$ denote a set that contains $\mathbf{W}$ in addition to its parents, children, ancestors and descendants in G, respectively. We also define the set $\mathrm{Co}(\mathbf{W})_G$ to denote the set of vertices that are connected to $\mathbf{W}$ in G via paths where the directionality of the edges is ignored, including $\mathbf{W}$. The root set of a graph G is the set of vertices without any descendants $\{X \in V \mid \mathrm{De}(X)_G \setminus \{X\} = \emptyset\}$, where $\setminus$ denotes the set difference. A subgraph of a graph $G = \langle \mathbf{V}, \mathbf{E} \rangle$ induced by a set of vertices $\mathbf{W} \subset \mathbf{V}$ is denoted by $G[\mathbf{W}]$. This subgraph retains all edges $V \rightarrow W$ of G such that $V, W \in \mathbf{W}$. The graph obtained from G by removing all incoming edges of $\mathbf{X}$ and all outgoing edges of $\mathbf{Z}$ is written as $G_{\bar{\mathbf{X}},\underline{\mathbf{Z}}}$. To facilitate analysis of causal effects we must first define the probabilistic causal model (Pearl, 2009).

Definition 1 (Probabilistic Causal Model) *A* probabilistic causal model *is a quadruple*

$$M = \langle \mathbf{U}, \mathbf{V}, \mathbf{F}, P(\mathbf{u}) \rangle,$$

where

1. $\mathbf{U}$ *is a set of unobserved (exogenous) variables that are determined by factors outside the model.*

2. $\mathbf{V}$ *is a set* $\{V_1, V_2, \ldots, V_n\}$ *of observed (endogenous) variables that are determined by variables in* $\mathbf{U} \cup \mathbf{V}$*.*

3. $\mathbf{F}$ *is a set of functions* $\{f_{V_1}, f_{V_2}, \ldots, f_{V_n}\}$ *such that each* f_{V_i} *is a mapping from (the respective domains of)* $\mathbf{U} \cup (\mathbf{V} \setminus \{V_i\})$ *to* V_i*, and such that the entire set* $\mathbf{F}$ *forms a mapping from* $\mathbf{U}$ *to* $\mathbf{V}$*.*

4. $P(\mathbf{u})$ *is a joint probability distribution of the variables in the set* $\mathbf{U}$*.*

Each causal model induces a causal diagram which is a directed graph that provides a graphical means to convey our assumptions of the causal mechanisms involved. The induced graph is constructed by adding a vertex for each variable in $\mathbf{U} \cup \mathbf{V}$ and a directed edge from $V_i \in \mathbf{U} \cup \mathbf{V}$ into $V_j \in \mathbf{V}$ whenever f_{V_j} is defined in terms of V_i.

Causal inference often focuses on a sub-class of models that satisfy additional assumptions: each $U \in \mathbf{U}$ appears in at most two functions of $\mathbf{F}$, the variables in $\mathbf{U}$ are mutually independent and the induced graph of the model is acyclic. Models that satisfy these additional assumptions are called *semi-Markovian causal models.* A graph associated with a semi-Markovian model is called a *semi-Markovian graph* (SMG). In SMGs every $U \in \mathbf{U}$ has at most two children. When semi-Markovian models are considered it is common not to depict background variables in the induced graph explicitly. Unobserved variables with exactly two children are not denoted as $V_i \leftarrow U \rightarrow V_j$ but as a bidirected edge $V_i \leftrightarrow V_j$ instead. Furthermore, unobserved variables with only one or no children are omitted entirely. We also adopt these abbreviations. For SMGs the sets $\mathrm{Pa}(\cdot)_G, \mathrm{Ch}(\cdot)_G, \mathrm{An}(\cdot)_G, \mathrm{De}(\cdot)_G$ and $\mathrm{Co}(\cdot)_G$ contain only observed vertices. Additionally, a subgraph $G[\mathbf{W}]$ of an SMG G will also retain any bidirected edges between vertices in $\mathbf{W}$.

Any DAG can be associated with an SMG by constructing its *latent projection* (Verma, 1993).

Definition 2 (latent projection) *Let $G = \langle \mathbf{V} \cup \mathbf{L}, \mathbf{E} \rangle$ be a DAG such that the vertices in $\mathbf{V}$ are observed and the vertices in $\mathbf{L}$ are latent. The* latent projection $L(G, \mathbf{V})$ *is a DAG $\langle \mathbf{V}, \mathbf{E}_L \rangle$, where for every pair of distinct vertices $Z, W \in \mathbf{V}$ it holds that:*

1. *$L(G, \mathbf{V})$ contains an edge $Z \rightarrow W$ if there exists a directed path $Z \rightarrow \cdots \rightarrow W$ in G on which every vertex except Z and W is in $\mathbf{L}$.*
2. *$L(G, \mathbf{V})$ contains an edge $Z \leftrightarrow W$ if there exists a path from Z to W in G that does not contain the pattern $Z \rightarrow M \leftarrow W$ (a collider) and on which every vertex except Z and W is in $\mathbf{L}$ and the first edge has an arrowhead pointing into W and the last edge has an arrowhead pointing into Z.*

From the construction it is easy to see that a latent projection is in fact an SMG. The induced graph of a probabilistic causal model can also be used to derive conditional independences among the variables in the model using a concept known as d-separation. We provide a definition for d-separation (Shpitser and Pearl, 2008) which takes into account the presence of bidirected edges and is thus suitable for SMGs.

Definition 3 (d-separation) *A path P in an SMG G is said to be d-separated by a set $\mathbf{Z}$ if and only if either*

1. *P contains one of the following three patterns of edges: $I \rightarrow M \rightarrow J$, $I \leftrightarrow M \rightarrow J$ or $I \leftarrow M \rightarrow J$, such that $M \in \mathbf{Z}$, or*
2. *P contains one of the following three patterns of edges: $I \rightarrow M \leftarrow J$, $I \leftrightarrow M \leftarrow J$, $I \leftrightarrow M \leftrightarrow J$, such that $De(M)_G \cap \mathbf{Z} = \emptyset$.*

Disjoint sets $\mathbf{X}$ and $\mathbf{Y}$ are said to be d-separated by $\mathbf{Z}$ in G if every path from $\mathbf{X}$ to $\mathbf{Y}$ is d-separated by $\mathbf{Z}$ in G.

Whenever we can decompose the joint distribution of the observed variables $\mathbf{V}$ and the unobserved variables $\mathbf{U}$ as $P(\mathbf{v}, \mathbf{u}) = \prod_{W \in \mathbf{V} \cup \mathbf{U}} P(w|\mathrm{Pa}^*(w)_G)$, where $\mathrm{Pa}^*(\cdot)$ also contains the unobserved parents but not the argument itself, we say that G is an I-map of $P(\mathbf{v}, \mathbf{u})$ (Pearl, 2009). If sets $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G, then $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$ in every P for which G is an I-map (Pearl, 1988). We use the notation of (Dawid, 1979) to denote this d-separation and conditional independence statement as $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_G$. It is clear that the graph induced by any semi-Markovian causal model is an I-map for the joint distribution $P(\mathbf{v}, \mathbf{u})$ induced by the model.

Our interest lies in the effects of actions imposing changes to the model. An action that forces $\mathbf{X}$ to take a specific value $\mathbf{x}$ is called an *intervention* and it is denoted by $\mathrm{do}(\mathbf{x})$ (Pearl, 2009). An intervention $\mathrm{do}(\mathbf{x})$ on a model M creates a new sub-model, denoted by $M_{\mathbf{x}}$, where the functions in $\mathbf{F}$ that determine the value of $\mathbf{X}$ have been replaced with constant functions. The *interventional distribution* of a set of variables $\mathbf{Y}$ in the model $M_{\mathbf{x}}$ is denoted by $P_{\mathbf{x}}(\mathbf{y})$. This distribution is also known as the *causal effect* of $\mathbf{X}$ on $\mathbf{Y}$.

Multiple causal models can share the same graph, and thus the same sub-model resulting from an intervention. The question is, are our assumptions encoded in the causal model sufficient to uniquely specify an interventional distribution of interest. This notion is captured by the following definition (Shpitser and Pearl, 2006).

Definition 4 (identifiability) *Let $G = \langle \mathbf{V}, \mathbf{E} \rangle$ be an SMG and let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint sets of variables such that $\mathbf{X}, \mathbf{Y} \subset \mathbf{V}$. The causal effect of $\mathbf{X}$ on $\mathbf{Y}$ is said to be* identifiable *from P in G if $P_{\mathbf{x}}(\mathbf{y})$ is uniquely computable from $P(\mathbf{V})$ in any causal model that induces G.*

In order to show the identifiability of a given effect we have to express the interventional distribution in terms of observed probabilities only. The link between observed probabilities and interventional distributions is provided by three inference rules known as *do-calculus* (Pearl, 1995):

1. Insertion and deletion of observations:

$$P_{\mathbf{x}}(\mathbf{y}|\mathbf{z}, \mathbf{w}) = P_{\mathbf{x}}(\mathbf{y}|\mathbf{w}), \text{ if } (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X}, \mathbf{W})_{G_{\overline{\mathbf{X}}}}.$$

2. Exchanging actions and observations:

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{x}}(\mathbf{y}|\mathbf{z}, \mathbf{w}), \text{ if } (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X}, \mathbf{W})_{G_{\overline{\mathbf{X}},\underline{\mathbf{Z}}}}.$$

3. Insertion and deletion of actions:

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{x}}(\mathbf{y}|\mathbf{w}), \text{ if } (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X}, \mathbf{W})_{G_{\overline{\mathbf{X}}, Z\overline{(\mathbf{W}})}},$$

where $Z(\mathbf{W}) = \mathbf{Z} \setminus \mathrm{An}(\mathbf{W})_{G_{\overline{\mathbf{X}}}}$.

Completeness of do-calculus was established independently by Huang and Valtorta (2006) and Shpitser and Pearl (2006). In this paper we focus on the solution provided by Shpitser and Pearl (2006). They constructed an identifiability algorithm called ID, which in essence applies the rules of do-calculus and breaks the problem into smaller sub-problems repeatedly.

3. ID Algorithm

In order to present the ID algorithm, we first need some additional definitions that are used to construct the graphical criterion for non-identifiability (Shpitser and Pearl, 2006).

Definition 5 (C-component) *Let G be an SMG and let $C \subseteq G$. If every pair of vertices in C is connected by a bidirected path, that is a path consisting entirely of bidirected edges, then C is a* C-component *(confounded component). Furthermore, C is a* maximal C-component *if C contains every vertex connected to C via bidirected paths in G and C is an induced subgraph of G.*

No restrictions are imposed on the directed edges of a C-component. The same is not true for the maximal C-components (also known as districts) of an SMG G, which are assumed to be induced subgraphs of G. This requirement guarantees the uniqueness of the maximal C-components.

Maximal C-components are an important tool for identifying causal effects. The set of maximal C-components of a semi-Markovian graph G is denoted by $C(G)$. A result in (Tian, 2002) states that if $C = \langle \mathbf{C}, \mathbf{E} \rangle$ is a maximal C-component and $C \subset G$ then the causal effect $P_{\mathbf{v} \setminus \mathbf{c}}(\mathbf{c})$ is identifiable from P in G. A distribution P of a semi-Markovian model also factorizes with respect to the maximal C-components of the induced graph G such that $P(\mathbf{v}) = \prod_{\langle \mathbf{C}, \mathbf{E} \rangle \in C(G)} P_{\mathbf{v} \setminus \mathbf{c}}(\mathbf{c})$ (Shpitser and Pearl, 2006). It is precisely this factorization that the ID algorithm takes advantage of. A specific type of C-component is used to characterize problematic structures for identifiability.

Definition 6 (C-forest) *Let G be an SMG and let $\mathbf{Y}$ be the root set of G. If G is a C-component and all observed vertices have at most one child, then G is a $\mathbf{Y}-$*rooted C-forest*.*

The complete criterion for non-identifiability uses a structure formed by two C-forests:

Definition 7 (hedge) *Let $\mathbf{X}, \mathbf{Y} \subset \mathbf{V}$ be disjoint sets of variables and let G be an SMG. Let $F = \langle \mathbf{V}_F, \mathbf{E}_F \rangle$ and $F' = \langle \mathbf{V}_{F'}, \mathbf{E}_{F'} \rangle$ be $\mathbf{R}$-rooted C-forests in G such that $\mathbf{V}_F \cap \mathbf{X} \neq \emptyset$, $\mathbf{V}_{F'} \cap \mathbf{X} = \emptyset$, $F' \subseteq F$, and $\mathbf{R} \subseteq An(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$. Then F and F' form a hedge for $P_{\mathbf{x}}(\mathbf{y})$ in G.*

Intuitively hedges are a difficult concept. Whenever a hedge is present, there exists two causal models with the same probability distribution over $\mathbf{V}$ but their interventional distributions do not agree. Observational data can not be used to estimate causal effects in this scenario. We are now ready to present the ID algorithm.

Shpitser and Pearl (2006) showed that whenever Algorithm 1 returns an expression for a causal effect, it is correct. Additionally whenever line 5 is triggered there exists a hedge for the causal effect currently being identified. This result establishes the completeness of the algorithm and also the completeness of do-calculus, since the soundness of each line of the algorithm can be shown with do-calculus and standard probability calculus alone.

Algorithm 1 The causal effect of intervention $do(\mathbf{X} = \mathbf{x})$ on $\mathbf{Y}$ (ID).

INPUT: Value assignments $\mathbf{x}$ and $\mathbf{y}$, joint distribution $P(\mathbf{v})$ and an SMG $G = \langle \mathbf{V}, \mathbf{E} \rangle$. G is an I-map of P.

OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y})$ in terms of $P(\mathbf{v})$ or $\mathbf{FAIL}(F, F')$.

```
   function ID($\mathbf{y}, \mathbf{x}, P, G$)
1: if $\mathbf{x} = \emptyset$,
      return $\sum_{v \in \mathbf{v} \setminus \mathbf{y}} P(\mathbf{v})$.
2: if $\mathbf{V} \neq \mathrm{An}(\mathbf{Y})_G$,
      return ID($\mathbf{y}, \mathbf{x} \cap \mathrm{An}(\mathbf{y})_G, P(\mathrm{An}(\mathbf{Y})_G), G[\mathrm{An}(\mathbf{Y})_G]$).
3: let $\mathbf{W} = (\mathbf{V} \setminus \mathbf{X}) \setminus \mathrm{An}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$.
   if $\mathbf{W} \neq \emptyset$,
      return ID($\mathbf{y}, \mathbf{x} \cup \mathbf{w}, P, G$).
4: if $C(G[\mathbf{V} \setminus \mathbf{X}]) = \{G[\mathbf{S}_1], \ldots, G[\mathbf{S}_k]\}$,
      return $\sum_{v \in \mathbf{v} \setminus (\mathbf{y} \cup \mathbf{x})} \prod_{i=1}^{k}$ ID($\mathbf{s}_i, \mathbf{v} \setminus \mathbf{s}_i, P, G$).
   if $C(G[\mathbf{V} \setminus \mathbf{X}]) = \{G[\mathbf{S}]\}$,
5:    if $C(G) = \{G\}$,
         throw FAIL($G, G[\mathbf{S}]$).
6:    if $G[\mathbf{S}] \in C(G)$,
         return $\sum_{v \in \mathbf{s} \setminus \mathbf{y}} \prod_{V_i \in \mathbf{S}} P(v_i | v_\pi^{(i-1)})$.
7:    if $(\exists \mathbf{S}')\mathbf{S} \subset \mathbf{S}'$ such that $G[\mathbf{S}'] \in C(G)$,
         return ID($\mathbf{y}, \mathbf{x} \cap \mathbf{s}', \prod_{V_i \in \mathbf{S}'} P(V_i | V_\pi^{(i-1)} \cap \mathbf{S}', v_\pi^{(i-1)} \setminus \mathbf{s}'), G[\mathbf{S}']$).
```

4. Pruning of Variables

In this section we present a number of results that deal with variables that are not necessary for identification either by removing them from the graph or by considering them latent. When the causal effect $P_{\mathbf{x}}(\mathbf{y})$ is considered in an SMG we can present an outline of the pruning process:

1. Removal of non-ancestors of $\mathbf{Y}$.
2. Removal of ancestors of $\mathbf{X}$ that are connected to $\mathbf{Y}$ only via $\mathbf{X}$ under certain conditions.
3. Removal of vertices connected to other vertices only through a single vertex.
4. Identification in a latent projection under certain conditions.

Steps 2–4 are new and they are based on the results of this section. Step 1 is derived from a useful result by Shpitser and Pearl (2006) which states that for a causal effect $P_{\mathbf{x}}(\mathbf{y})$ we can always ignore non-ancestors of $\mathbf{Y}$.

Lemma 8 *Let $\mathbf{X}' = \mathbf{X} \cap An(\mathbf{Y})_G$. Then $P_{\mathbf{x}}(\mathbf{y})$ obtained from P in G is equal to $P'_{\mathbf{x}'}(\mathbf{y})$ obtained from $P' = P(An(\mathbf{Y})_G)$ in $G[An(\mathbf{Y})_G]$.*

Lemma 8 is implemented on line 2 of Algorithm 1. Not all ancestors of $\mathbf{Y}$ are always necessary for identification. The next result states that we may sometimes remove ancestors of $\mathbf{X}$ that are connected to $\mathbf{Y}$ only through $\mathbf{X}$.

Theorem 9 *Let G be an SMG and let $\mathbf{Z} \subset \mathbf{V}$ be the set of all vertices such that $\mathbf{X}$ intercepts all paths from $\mathbf{Z}$ to $\mathbf{Y}$. Then the causal effect $P_{\mathbf{x}}(\mathbf{y})$ obtained from P in G is equal to $P'_{\mathbf{x}}(\mathbf{y})$ obtained from $P' = P(\mathbf{V} \setminus \mathbf{Z})$ in $G[\mathbf{V} \setminus \mathbf{Z}]$ if $\mathbf{Z}$ contains no members of $\mathbf{X}$ and if $G[\mathbf{V} \setminus \mathbf{Z}] = L(G, \mathbf{V} \setminus \mathbf{Z})$.*

Proof Let $G' = G[\mathbf{V} \setminus \mathbf{Z}]$ and assume that $G' = L(G, \mathbf{V} \setminus \mathbf{Z})$. Let $\mathbf{U}_{\mathbf{Z}}$, $\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}$ and $\mathbf{U}_{\mathbf{X}}$ be sets of unobserved variables such that for all $U \in \mathbf{U}_{\mathbf{Z}}$ it holds that $\mathrm{Ch}(U)_{G_{\overline{\mathbf{X}}}} \subseteq \mathbf{Z}$, for all $U \in \mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}$ it holds that $\mathrm{Ch}(U)_{G_{\overline{\mathbf{X}}}} \subseteq \mathbf{V} \setminus \mathbf{Z}$ and for all $U \in \mathbf{U}_{\mathbf{X}}$ it holds that $\mathrm{Ch}(U)_G \in \mathbf{X}$. The sets $\mathbf{U}_{\mathbf{Z}}$, $\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}$ and $\mathbf{U}_{\mathbf{X}}$ partition $\mathbf{U}$ because $\mathbf{X}$ intercepts all paths from $\mathbf{Z}$ to $\mathbf{Y}$. According to the third rule of do-calculus $P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x},\mathbf{z}}(\mathbf{y})$ because the condition $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z}|\mathbf{X})_{G_{\overline{\mathbf{X}}}}$ holds as removing the edges incoming to $\mathbf{X}$ separates $\mathbf{X}$ from its ancestors. Applying the truncated factorization formula (Pearl, 2009) we have that

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}) = \sum_{\mathbf{U}} \sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X}\cup\mathbf{Z})} \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\mathbf{Z})} P(v_i|\mathrm{Pa}(v_i)_G \setminus \{v_i\}) \prod_{\mathbf{U}} P(u_i).$$

Since variables in $\mathbf{U}_{\mathbf{Z}}$ can only be parents of variables in $\mathbf{Z}$ or $\mathbf{X}$ in G, we can sum them out from the previous expression and obtain

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}) = \sum_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}\cup\mathbf{U}_{\mathbf{X}}} \sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X}\cup\mathbf{Z})} \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\mathbf{Z})} P(v_i|\mathrm{Pa}(v_i)_G \setminus \{v_i\}) \prod_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}\cup\mathbf{U}_{\mathbf{X}}} P(u_i).$$

Similarly, variables in $\mathbf{U}_{\mathbf{X}}$ can only be parents of variables in $\mathbf{X}$ in G, so we can also sum them out of the expression to obtain

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}) = \sum_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} \sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X}\cup\mathbf{Z})} \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\mathbf{Z})} P(v_i|\mathrm{Pa}(v_i)_G \setminus \{v_i\}) \prod_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} P(u_i).$$

We let $\mathbf{V}' = \mathbf{V} \setminus \mathbf{Z}$. Verma (1993) showed that a graph and its latent projection have the same set of conditional independence relations among the observed variables. Because we have assumed that $G' = L(G, \mathbf{V} \setminus \mathbf{Z})$ every conditional independence between variables in $\mathbf{V}'$ and $\mathbf{U}_{\mathbf{V}'}$ applies in both G and G'. We have that for all $V_i \in \mathbf{V}' \setminus \mathbf{X}$ it holds that $P(v_i|\mathrm{Pa}(v_i)_G \setminus \{v_i\}) = P'(v_i|\mathrm{Pa}(v_i)_{G'} \setminus \{v_i\})$ and for all $U_i \in \mathbf{U}_{\mathbf{V}'}$ it holds that $P(u_i) = P'(u_i)$. Finally we obtain

$$P_{\mathbf{x},\mathbf{z}}(\mathbf{y}) = \sum_{\mathbf{U}_{\mathbf{V}'}} \sum_{\mathbf{V}'\setminus(\mathbf{Y}\cup\mathbf{X})} \prod_{\mathbf{V}'\setminus\mathbf{X}} P'(v_i|\mathrm{Pa}(v_i)_{G'} \setminus \{v_i\}) \prod_{\mathbf{U}_{\mathbf{V}'}} P'(u_i) = P'_{\mathbf{x}}(\mathbf{y}).$$

■

Theorem 9 can also be applied in a more general setting where a subset of $\mathbf{X}$ intercepts all paths from a set $\mathbf{Z}$ to $\mathbf{Y}$

Corollary 10 *Let G be an SMG and let $\mathbf{Z} \subset \mathbf{V}$ be the set of all vertices such that a set $\mathbf{W} \subseteq \mathbf{X}$ intercepts all paths from $\mathbf{Z}$ to $\mathbf{Y}$ and no member of $\mathbf{X} \setminus \mathbf{W}$ is a descendant of $\mathbf{W}$. Then the causal effect $P_{\mathbf{x}}(\mathbf{y})$ obtained from P in G is equal to $P'_{\mathbf{x}\setminus\mathbf{z}}(\mathbf{y})$ obtained from $P' = P(\mathbf{V} \setminus \mathbf{Z})$ in $G[\mathbf{V} \setminus \mathbf{Z}]$ if $\mathbf{Z}$ contains no members of $\mathbf{W}$ and if $G[\mathbf{V} \setminus \mathbf{Z}] = L(G, \mathbf{V} \setminus \mathbf{Z})$.*

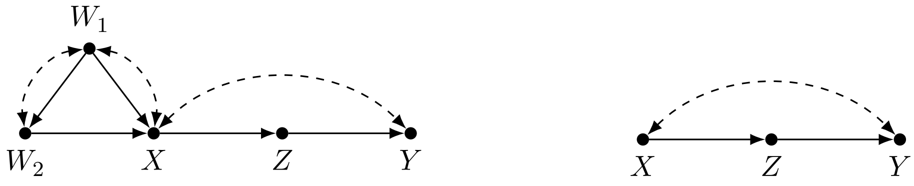
Proof Since $\mathbf{W}$ intercepts all paths from $\mathbf{Z}$ to $\mathbf{Y}$ and no member of $\mathbf{X} \setminus \mathbf{W}$ is a descendant of $\mathbf{W}$, it follows that no member of $\mathbf{W}$ is in $\mathbf{Z}$. According to the third rule of do-calculus we have that $(\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} | \mathbf{X} \setminus \mathbf{Z})_{G_{\overline{\mathbf{X} \setminus \mathbf{Z}}}}$ and $P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x} \setminus \mathbf{z}}(\mathbf{y})$. The claim now follows by applying Theorem 9 to $P_{\mathbf{x} \setminus \mathbf{z}}(\mathbf{y})$. ■

Corollary 11 *When the causal effect $P_{\mathbf{x}}(\mathbf{y})$ is considered in graph G, a set of vertices $\mathbf{Z} = An(\mathbf{Y})_G \setminus Co(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$ can be removed from G if $G[\mathbf{V} \setminus \mathbf{Z}] = L(G, \mathbf{V} \setminus \mathbf{Z})$.*

Proof The set $\mathrm{An}(\mathbf{Y})_G$ contains $\mathbf{Y}$ in addition to the ancestors of $\mathbf{Y}$, and the set $\mathrm{Co}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$ contains $\mathbf{Y}$ and all vertices that are connected to $\mathbf{Y}$ via a path that does not contain edges incoming to $\mathbf{X}$. Therefore, $\mathbf{Z}$ contains such ancestors of $\mathbf{Y}$ that all paths from $\mathbf{Z}$ to $\mathbf{Y}$ contain $\mathbf{X}$. The removal of $\mathbf{Z}$ from G is now licensed by Corollary 10. ■

Corollary 11 provides a constructive criterion for the set $\mathbf{Z}$ described in Corollary 10 when G consists only of $\mathbf{Y}$ and its ancestors. If a vertex Z_i is a member of $\mathrm{An}(\mathbf{Y})_G \setminus \mathrm{Co}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$ then it must be connected to $\mathbf{Y}$ only through paths containing some $\mathbf{W}_{Z_i} \subseteq \mathbf{X}$. We can always choose the sets $\mathbf{W}_{Z_i}$ in such a way that the union $\mathbf{W} = \cup \mathbf{W}_{Z_i}$ over the members Z_i of $\mathrm{An}(\mathbf{Y})_G \setminus \mathrm{Co}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$ has no descendants in $\mathbf{X} \setminus \mathbf{W}$. The set $\mathbf{W}$ intercepts all paths from $\mathbf{Z} = \cup \{Z_i\}$ to $\mathbf{Y}$. Conversely, if Z_i is a vertex such that a set $\mathbf{W} \subseteq \mathbf{X}$ intercepts all paths from Z_i to $\mathbf{Y}$, then Z_i cannot be connected to $\mathbf{Y}$ in $G_{\overline{\mathbf{X}}}$. If we assume that $G = G[\mathrm{An}(\mathbf{Y})_G]$ it follows that Z_i is a member of $\mathrm{An}(\mathbf{Y})_G \setminus \mathrm{Co}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$.

We present a simple example to motivate the usefulness of Corollary 11. We apply the ID algorithm to identify the causal effect of X on Y in graph G of Figure 2(a). Applying the ID algorithm results in the following expression for the causal effect



(a) Graph G.

(b) Subgraph $G[\{X, Y, Z\}]$.

Figure 2: A graph for an example where Corollary 11 allows us to remove vertices W_1 and W_2 when the causal effect of X on Y is considered.

$$\sum_z P(z|w_1, w_2, x) \left(\sum_{w_1, w_2, x'} P(y|w_1, w_2, x', z) P(x'|w_1, w_2) P(w_2|w_1) P(w_1) \right).$$

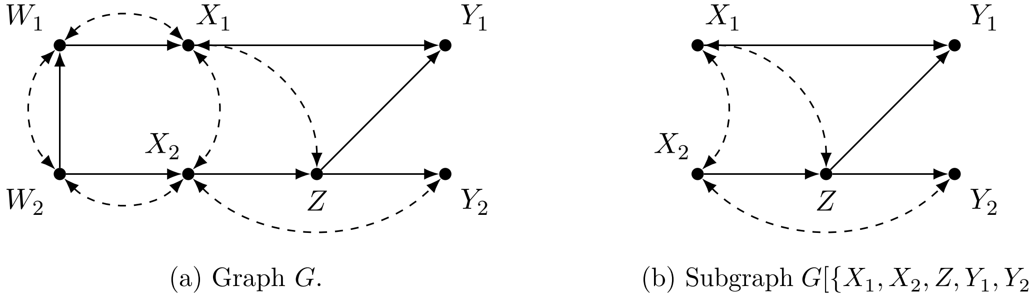
Applying Corollary 11 in this case would result in the removal of the vertices W_1 and W_2 from the graph, since they are ancestors of Y in G but not connected to Y in $G_{\overline{X}}$ and the corresponding latent projection is the subgraph $G[\{X, Y, Z\}]$ of Figure 2(b). Running the

ID algorithm in this subgraph provides us the following expression

$$\sum_{z} P(z|x) \left(\sum_{x'} P(y|x', z)P(x') \right).$$

We may consider this expression simpler compared to the previous output by noting that W_1 and W_2 do not appear in the expression and it has fewer unique terms. The same expression can also be obtained manually by applying the front-door criterion (Pearl, 2009).

Often the question of identifiability can not be answered directly by neither the back-door nor the front-door criterion which leads us to more general methods, such as the ID algorithm. We are interested in the causal effect of W_1, X_1 and X_2 on Y_1 and Y_2 in the graph of Figure 3(a).



(a) Graph G.

(b) Subgraph $G[\{X_1, X_2, Z, Y_1, Y_2\}]$.

Figure 3: A graph for an example where the back-door and front-door criteria are unavailable, but Corollary 11 allows us to remove vertices W_1 and W_2 even when W_1 is part of the intervention.

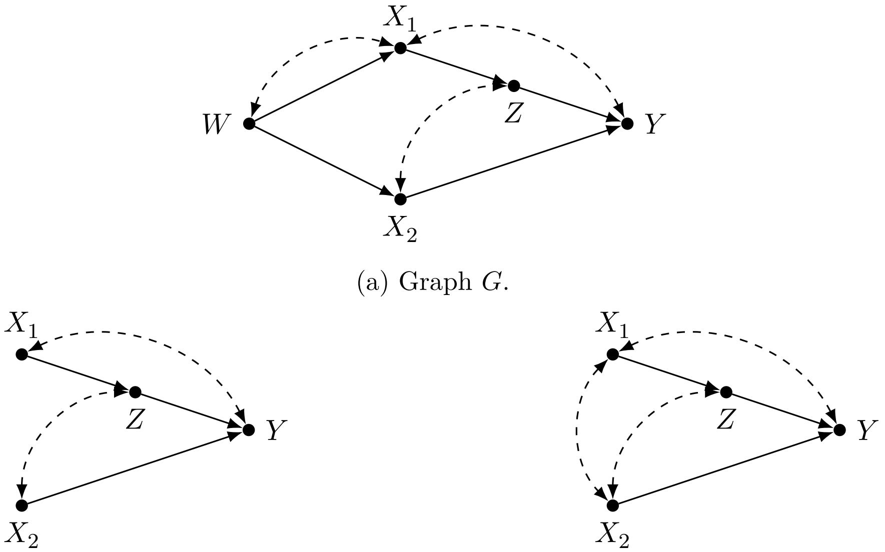
Direct application of the ID algorithm provides us with the following expression

$$\sum_{z} P(y_1|w_2, w_1, x_2, x_1, z)P(z|w_2, x_2) \left(\sum_{w_2, x_2'} P(y_2|w_2, x_2', z)P(x_2'|w_2)P(w_2) \right).$$

Corollary 11 licenses the removal of W_1 and W_2 from the graph. By running ID again in the resulting subgraph $G[\{X_1, X_2, Z, Y_1, Y_2\}]$ as shown in Figure 3(b) we obtain a simpler expression for the causal effect

$$\sum_{z} P(y_1|x_1, x_2, z)P(z|x_2) \left(\sum_{x_2'} P(y_2|x_2', z)P(x_2') \right).$$

The next example illustrates the necessity of the assumption $G[\mathbf{V} \setminus \mathbf{Z}] = L(G, \mathbf{V} \setminus \mathbf{Z})$ of Theorem 9 and Corollary 10. We are interested in the causal effect of X_1 and X_2 on Y in graph G of Figure 4(a). In this graph W is connected to Y only through X_1 and X_2, but the corresponding latent projection does not match the subgraph with W removed as seen in Figures 4(b) and 4(c). In G the causal effect is identifiable, but it is not identifiable in the latent projection $G' = L(G, \{X_1, X_2, Z, Y\})$. In this latent projection a bidirected edge exists between X_1 and X_2 and a hedge is formed by the C-forests G' and $G[\{Y\}]$.



(a) Graph G.

(b) Subgraph $G[\{X_1, X_2, Z, Y\}]$.

(c) Latent projection $L(G, \{X_1, X_2, Z, Y\})$.

Figure 4: An example where the assumptions of Corollary 10 are not met because the subgraph in panel (b) and the latent projection in panel (c) differ from each other.

We may also remove sets of vertices that are connected to the rest of the graph only through a single vertex even when no intervention on the corresponding variable has taken place.

Theorem 12 *Let G be an SMG such that $G = G[An(\mathbf{Y})_G]$ for a set of vertices $\mathbf{Y}$ and let W be a vertex of G. If there exists a set $\mathbf{Z}$ such that $\mathbf{Z} \cap (\mathbf{Y} \cup \mathbf{X}) = \emptyset$ and $\mathbf{Z}$ is connected to $\mathbf{V} \setminus \mathbf{Z}$ only through W. Then the causal effect $P_{\mathbf{x}}(\mathbf{y})$ obtained from P in G is equal to $P'_{\mathbf{x}}(\mathbf{y})$ obtained from $P' = P(\mathbf{V} \setminus \mathbf{Z})$ in $G[\mathbf{V} \setminus \mathbf{Z}]$.*

Proof Let $G' = G[\mathbf{V} \setminus \mathbf{Z}]$ and let $\mathbf{U}_{\mathbf{Z}}$ and $\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}$ be sets of unobserved variables such that for all $U \in \mathbf{U}_{\mathbf{Z}}$ it holds that $\text{Ch}(U)_G \in \mathbf{Z} \cup \{W\}$ and for all $U \in \mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}$ it holds that $\text{Ch}(U)_G \in (\mathbf{V} \setminus \mathbf{Z}) \cup \{W\}$. Sets $\mathbf{U}_{\mathbf{Z}}$ and $\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}$ partition $\mathbf{U}$ because $\mathbf{Z}$ is connected to $\mathbf{V} \setminus \mathbf{Z}$ only through W. Applying the truncated factorization formula yields

$$P_{\mathbf{x}}(\mathbf{y}) = \sum_{\mathbf{U}} \sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X})} P(w|\text{Pa}(w)_G \setminus \{w\}) \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\{W\})} P(v_i|\text{Pa}(v_i)_G \setminus \{v_i\}) \prod_{\mathbf{U}} P(u_i).$$

Since variables in $\mathbf{U}_{\mathbf{Z}}$ and $\mathbf{Z}$ can be connected to the other vertices of G only through W we can complete the marginalization over $\mathbf{Z}$ and $\mathbf{U}_{\mathbf{Z}}$

$$P_{\mathbf{x}}(\mathbf{y}) = \sum_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} \sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X}\cup\mathbf{Z})} P(w|\text{Pa}(w)_G \setminus (\mathbf{z} \cup \{w\})) \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\mathbf{Z}\cup\{W\})} P(v_i|\text{Pa}(v_i)_G \setminus \{v_i\}) \prod_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} P(u_i).$$

Because we have assumed that $\mathbf{Z}$ is disconnected from $\mathbf{V} \setminus \mathbf{Z}$ in $G_{\overline{W}}$ we have that $G[\mathbf{V} \setminus \mathbf{Z}] = L(G, \mathbf{V} \setminus \mathbf{Z})$. Therefore, just as in the proof of Theorem 9, we have that $P(v_i|\text{Pa}(v_i)_G \setminus \{v_i\}) =$

$P'(v_i|\mathrm{Pa}(v_i)_{G'} \setminus \{v_i\})$ for all $V_i \in \mathbf{V} \setminus (\mathbf{X} \cup \mathbf{Z} \cup \{W\})$ and $P(u_i) = P'(u_i)$ for all $U_i \in \mathbf{U}_{\mathbf{V} \setminus \mathbf{Z}}$. Additionally, we have $P(w|\mathrm{Pa}(w)_G \setminus (\mathbf{z} \cup \{w\})) = P(w|Pa(w)_{G'} \setminus \{w\})$. Finally we obtain

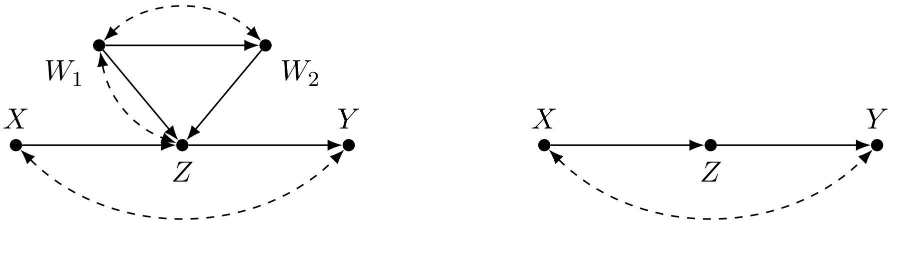
$$\begin{aligned}
P_{\mathbf{x}}(\mathbf{y}) &= \sum_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} \sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X}\cup\mathbf{Z})} P(w|\mathrm{Pa}(w)_{G'} \setminus \{w\}) \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\mathbf{Z}\cup\{W\})} P(v_i|\mathrm{Pa}(v_i)_{G'} \setminus \{v_i\}) \prod_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} P(u_i) \\
&= \sum_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} \sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X}\cup\mathbf{Z})} \prod_{\mathbf{V}\setminus(\mathbf{X}\cup\mathbf{Z})} P(v_i|\mathrm{Pa}(v_i)_{G'} \setminus \{v_i\}) \prod_{\mathbf{U}_{\mathbf{V}\setminus\mathbf{Z}}} P(u_i) \\
&= P'_{\mathbf{x}}(\mathbf{y}).
\end{aligned}$$

■

Corollary 13 *Let W be a vertex of an SMG G and let $\mathbf{R} = An(W)_{G_{\overline{\mathbf{X}}}} \setminus De(\mathbf{X})_G$. When the causal effect $P_{\mathbf{x}}(\mathbf{y})$ is considered in graph G, the set of vertices $\mathbf{T} = \mathbf{R} \setminus Co(\mathbf{V} \setminus \mathbf{R})_{G_{\overline{W}}}$ can be removed from the graph if $G = G[An(\mathbf{Y})_G]$.*

Proof No descendant of $\mathbf{X}$ can be removed via theorem 12 since they are connected to $\mathbf{X}$ and the set to be removed cannot itself contain $\mathbf{X}$. By removing descendants of $\mathbf{X}$ and $\mathbf{X}$ itself, and assuming that $G = G[\mathrm{An}(\mathbf{Y})_G]$, we have that $\mathbf{T} \cap (\mathbf{X} \cup \mathbf{Y}) = \emptyset$. Thus it remains to remove those vertices from $\mathbf{R}$ that are connected to $\mathbf{V} \setminus \mathbf{R}$ through a path that does not contain W. Removal of the resulting set $\mathbf{T}$ from the graph is now licensed by theorem 12. ■

In the following example we consider the causal effect of X on Y in graph G of Figure 5(a) and show how Corollary 13 can be applied. The ID algorithm succeeds in identifying the causal effect and returns following expression for it



(a) Graph G.

(b) Subgraph $G[\{X, Y, Z\}]$.

Figure 5: First example of Corollary 13.

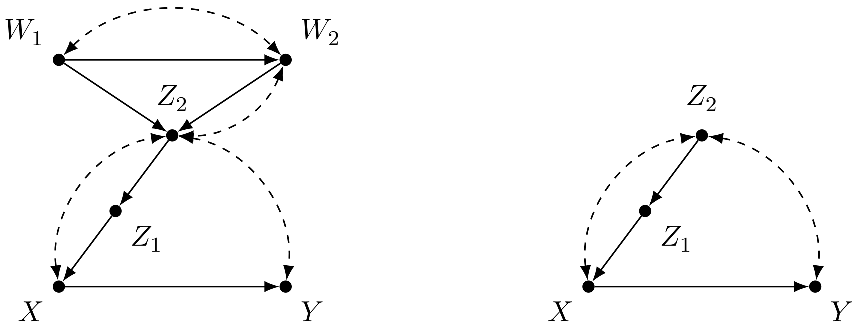
$$\sum_{w_1,w_2,z} P(z|x, w_1, w_2)P(w_2|x, w_1)P(w_1|x) \left(\sum_{x'} P(y|x', w_1, w_2, z)P(x') \right).$$

We apply Corollary 13 which allows us to remove W_1 and W_2 from the graph, since they are connected to other vertices of the graph through a single vertex Z. Applying the ID algorithm in the resulting subgraph $G[\{X, Y, Z\}]$ of Figure 5(b) provides us the following expression

$$\sum_{z} P(z|x) \left(\sum_{x'} P(y|x', z) P(x') \right).$$

The same expression can be obtained manually by applying the front-door criterion.

We provide another example on Corollary 13 with a slightly more complicated graph. We are interested in the causal effect of X on Y in graph G of Figure 6(a) We obtain a formula for the causal effect using the ID algorithm



(a) Graph G.

(b) Subgraph $G[\{X, Y, Z_1, Z_2\}$.

Figure 6: Second example of Corollary 13.

$$\frac{\sum_{w_2,z_2} P(y|w_1, w_2, z_2, z_1, x) P(x|w_1, w_2, z_2, z_1) P(z_2|w_1, w_2) P(w_2|w_1)}{\left(\sum_{w_2,z_2,y} P(y|w_1, w_2, z_2, z_1, x) P(x|w_1, w_2, z_2, z_1) P(z_2|w_1, w_2) P(w_2|w_1) \right)}.$$

Vertices W_1 and W_2 are connected to other vertices only through Z_2 which allows us to remove them from the graph. We obtain a simpler formula for the causal effect from the subgraph $G[\{X, Y, Z_1, Z_2\}]$ of Figure 6(b)

$$\frac{\sum_{z_2} P(y|z_2, z_1, x) P(x|z_2, z_1) P(z_2)}{\sum_{z_2,y} P(y|z_2, z_1, x) P(x|z_2, z_1) P(z_2)}.$$

The previous results have allowed us to completely remove specific vertices from the graph. Next we will consider cases where a vertex is present in the graph, but it is not necessary to observe it. This means that instead of the original graph we may consider identifiability in the corresponding latent projection, as characterized by the following lemma.

Lemma 14 *Let $G = \langle \mathbf{V}, \mathbf{E} \rangle$ be an SMG and let $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ be disjoint sets of variables. Let $P(\mathbf{V})$ be the joint distribution of $\mathbf{V}$. Then the causal effect of $\mathbf{X}$ on $\mathbf{Y}$ is identifiable from P' in the latent projection $L(G, \mathbf{V} \setminus \mathbf{Z})$ where $P' = P(\mathbf{V} \setminus \mathbf{Z})$, if and only if it is identifiable from P' in G.*

Proof Tian (2002) showed that the latent projection has the same topological relations over the observables and that it has the same set of maximal C-components. Thus if $P_{\mathbf{x}}(\mathbf{y})$

is identifiable from P' in G it is also identifiable from P' in $L(G, \mathbf{V} \setminus \mathbf{Z})$ with the same expression and vice versa. ■

In situations where $\mathbf{X}$ is a singleton we can exploit the following sufficient condition for identifiability by Tian and Pearl (2002).

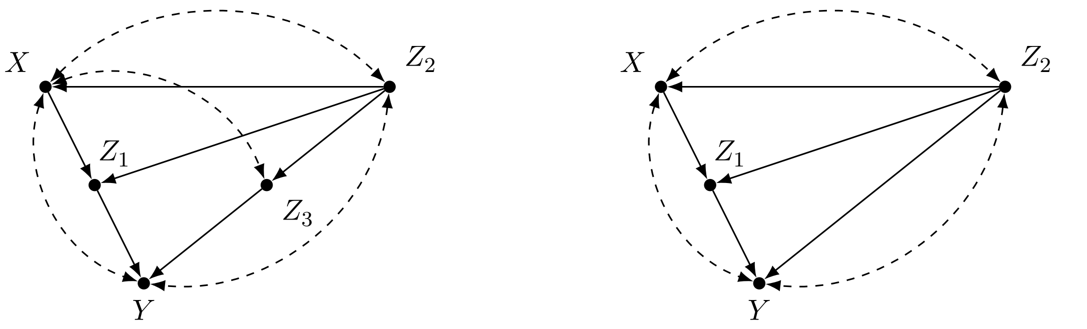
Theorem 15 *The causal effect $P_x(\mathbf{y})$ is identifiable if there is no bidirected path between X and any of its children in $G[An(\mathbf{Y})_G]$.*

We can regard any variable as latent when $\mathbf{X}$ is a singleton if the respective latent projection does not induce such a bidirected path.

Corollary 16 *Let G be an SMG and let $\mathbf{Y}$ be a set of vertices. Let $X, W \in \mathbf{V}$ such that $X \neq W$, $W \notin \mathbf{Y}$ and $X, W \in An(\mathbf{Y})_G$. The causal effect $P_x(\mathbf{y})$ obtained from P in G is equal to $P'_x(\mathbf{y})$ obtained from $P' = P(An(\mathbf{Y})_G \setminus \{w\})$ and $G' = L(G[An(\mathbf{Y})_G], An(\mathbf{Y})_G \setminus \{W\})$ if there is no bidirected path from X to any of its children in G'.*

Proof By Theorem 15 the causal effect $P'_x(\mathbf{y})$ is identifiable from P' in G'. By Lemma 14 $P'_x(\mathbf{y})$ is now identifiable from P' in G. $P_x(y)$ obtained from P' in G is equal to $P^*_x(y)$ obtained from $P^* = P(\mathrm{An}(\mathbf{Y})_G)$ in $G[\mathrm{An}(\mathbf{Y})]$ since identifiability from P' implies identifiability from P. Finally, $P^*_x(\mathbf{y})$ obtained from $P^* = P(\mathrm{An}(\mathbf{Y})_G)$ in $G[\mathrm{An}(\mathbf{Y})_G]$ is equal to $P_x(y)$ obtained from P in G. ■

We continue with an example on how Corollary 16 can be applied in practice. We consider the causal effect of X on Y in the graph G of Figure 7(a). The causal effect is identifiable and the output of the ID algorithm is



(a) Graph G.

(b) Latent projection $L(G, \{X, Y, Z_1, Z_2\})$.

Figure 7: A graph for an example where Corollary 16 allows us to make variable Z_3 latent.

$$\sum_{z_2,z_3,z_1} P(z_1|z_2,x)\frac{\left(\sum_{x'} P(y|z_2,x',z_3,z_1)P(z_3|z_2,x')P(x'|z_2)P(z_2)\right)}{\left(\sum_{x',y'} P(y'|z_2,x',z_3,z_1)P(z_3|z_2,x')P(x'|z_2)P(z_2)\right)}\times$$
$$\left(\sum_{x',z_3,y'} P(y'|z_2,x',z_3,z_1)P(z_3|z_2,x')P(x'|z_2)P(z_2)\right)P(z_3|z_2).$$

We may apply Corollary 16 by noting that $G = G[\mathrm{An}(Y)_G]$ and that there is no bidirected path between X and its only child Z_1 in the latent projection $L(G, \mathbf{V} \setminus \{Z_3\})$ as depicted in Figure 7(b).

Running the ID algorithm in $L(G, \mathbf{V} \setminus \{Z_3\})$ results in the following expression

$$\sum_{z_2,z_1} \left(\sum_{x'} P(y|z_2, x', z_1) P(x'|z_2) P(z_2) \right) P(z_1|z_2, x).$$

5. Pruning Identifiability Algorithm

Corollaries 11, 13 and 16 can be implemented as additional steps for the ID algorithm. For Algorithm 2, line 3 implements Corollary 11, line 4 implements Corollary 13 and line 5 implements Corollary 16. Other lines are identical to the ID algorithm. This algorithm is provided by the R package *causaleffect* which implements various causal inference algorithms such as the original ID algorithm (Tikka and Karvanen, 2017a).

The ordering of the variables in the loop on line 4 has no effect on the resulting expression, but the ordering does matter on line 5. Choosing a different ordering may lead to a different expression. For example, when identifying the causal effect of X on Y in graph G of Figure 8 one may obtain either the back-door formula or the front-door formula by proceeding in either the topological ordering or the reverse-topological ordering of the vertices, respectively.

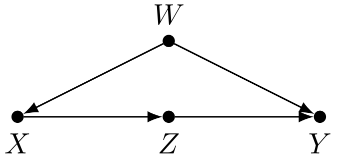


Figure 8: Graph G where different latent projections lead to different expressions for $P_x(y)$.

Algorithm 2 A pruning identifiability algorithm (PID) for causal effects.

INPUT: Value assignments $\mathbf{x}$ and $\mathbf{y}$, joint distribution $P(\mathbf{v})$ and an SMG $G = \langle \mathbf{V}, \mathbf{E} \rangle$. G is an I-map of P.

OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y})$ in terms of $P(\mathbf{v})$ or $\mathbf{FAIL}(F, F')$.

```
    function PID($\mathbf{y}, \mathbf{x}, P, G$)
 1: if $\mathbf{x} = \emptyset$,
       return $\sum_{v \in \mathbf{v} \setminus \mathbf{y}} P(\mathbf{v})$.
 2: if $\mathbf{V} \neq \mathrm{An}(\mathbf{Y})_G$,
       return $\mathbf{PID}(\mathbf{y}, \mathbf{x} \cap \mathrm{An}(\mathbf{y})_G, P(\mathrm{An}(\mathbf{Y})_G), G[\mathrm{An}(\mathbf{Y})_G])$
 3: let $\mathbf{Z} = \mathrm{An}(\mathbf{Y})_G \setminus \mathrm{Co}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$.
    if $\mathbf{Z} \neq \emptyset$ and $G[\mathbf{V} \setminus \mathbf{Z}] = L(G, \mathbf{V} \setminus \mathbf{Z})$,
       return $\mathbf{PID}(\mathbf{y}, \mathbf{x} \setminus \mathbf{z}, P(\mathbf{V} \setminus \mathbf{Z}), G[\mathbf{V} \setminus \mathbf{Z}])$
 4: for $W \in \mathbf{V} \setminus \mathbf{X}$ do
       let $\mathbf{R} = \mathrm{An}(W)_{G_{\overline{\mathbf{X}}}} \setminus \mathrm{De}(\mathbf{X})_G$.
       let $\mathbf{T} = \mathbf{T} \cup (\mathbf{R} \setminus \mathrm{Co}(\mathbf{V} \setminus \mathbf{R})_{G_{\overline{W}}})$.

    if $\mathbf{T} \neq \emptyset$,
       return $\mathbf{PID}(\mathbf{y}, \mathbf{x}, P(\mathbf{V} \setminus \mathbf{T}), G[\mathbf{V} \setminus \mathbf{T}])$
 5: if $\mathbf{X} = \{X\}$,
       let $G[\mathbf{S}_X] \in C(G), X \in \mathbf{S}_X$.
       if $\mathrm{Ch}(\mathbf{X})_{G[\mathbf{S}_X]} \setminus \mathbf{X} = \emptyset$,
          for $W \in \mathbf{V} \setminus (\mathbf{Y} \cup \mathbf{X})$ do
             let $G' = L(G, \mathbf{V} \setminus \{W\})$.
             let $G'[\mathbf{S}'_X] \in C(G'), X \in \mathbf{S}'_X$.
             if $\mathrm{Ch}(\mathbf{X})_{G'[\mathbf{S}'_X]} \setminus \mathbf{X} = \emptyset$,
                $P \leftarrow P(\mathbf{V} \setminus \{W\})$.
                $G \leftarrow G'$.
                $\mathbf{V} \leftarrow \mathbf{V} \setminus \{W\}$.
 6: let $\mathbf{W} = (\mathbf{V} \setminus \mathbf{X}) \setminus \mathrm{An}(\mathbf{Y})_{G_{\overline{\mathbf{X}}}}$.
    if $\mathbf{W} \neq \emptyset$,
       return $\mathbf{PID}(\mathbf{y}, \mathbf{x} \cup \mathbf{w}, P, G)$.
 7: if $C(G[\mathbf{V} \setminus \mathbf{X}]) = \{G[\mathbf{S}_1], \ldots, G[\mathbf{S}_k]\}$,
       return $\sum_{v \in \mathbf{v} \setminus (\mathbf{y} \cup \mathbf{x})} \prod_{i=1}^{k} \mathbf{PID}(\mathbf{s}_i, \mathbf{v} \setminus \mathbf{s}_i, P, G)$.
    if $C(G[\mathbf{V} \setminus \mathbf{X}]) = \{G[\mathbf{S}]\}$,
 8:    if $C(G) = \{G\}$,
          throw $\mathbf{FAIL}(G, G[\mathbf{S}])$.
 9:    if $G[\mathbf{S}] \in C(G)$,
          return $\sum_{v \in \mathbf{s} \setminus \mathbf{y}} \prod_{V_i \in \mathbf{S}} P(v_i | v_\pi^{(i-1)})$.
10:    if $(\exists \mathbf{S}') \mathbf{S} \subset \mathbf{S}'$ such that $G[\mathbf{S}'] \in C(G)$,
          return $\mathbf{PID}(\mathbf{y}, \mathbf{x} \cap \mathbf{s}', \prod_{V_i \in \mathbf{S}'} P(V_i | V_\pi^{(i-1)} \cap \mathbf{S}', v_\pi^{(i-1)} \setminus \mathbf{s}'), G[\mathbf{S}'])$.
```

On line 5 we first check whether the set $\mathbf{X}$ is a singleton. If so, we determine whether any children of X belong the same C-component as X. If no such children exist, we iterate over the possible latent projections in an attempt to find one that does not induce a bidirected path between X and any of its children in the projection. After the new pruning steps have been carried out, we attempt identification using the original formulation of the ID algorithm.

We return to the example presented in the introduction and show how Algorithm 2 operates to derive the expression for $P_x(y)$ in the graph of Figure 1(a). We choose the topological ordering to be $Y > Z_1 > X > Z_3 > W_2 > Z_4 > Z_2 > W_1$. We begin on line 3, since $\mathbf{Z} = \mathrm{An}(Y)_G \setminus \mathrm{Co}(Y)_{G_{\overline{X}}} = \mathbf{V} \setminus \{W_1, W_2\}$ and continue by calling $\mathrm{PID}(y, x, P(\mathbf{V} \setminus \{W_1, W_2\}), G[\mathbf{V} \setminus \{W_1, W_2\}])$. In the presentation below, $\mathbf{V}$ and G refer to the set of vertices and graph in the current call of PID, respectively.

Next we enter the loop on line 4. When $W = Z_1$ we obtain $\mathbf{R} = \mathrm{An}(Z_1)_{G_{\overline{X}}} \setminus \mathrm{De}(X)_G = \{Z_2, Z_4\}$ and $\mathbf{R} \setminus \mathrm{Co}(\mathbf{V} \setminus \mathbf{R})_{G_{\overline{Z_1}}} = \{Z_2, Z_4\} \setminus \{Z_2\} = \{Z_4\}$ since Z_4 is an ancestor of Z_1 and it is disconnected from other vertices in $G_{\overline{Z_1}}$. Other choices of W result in an empty set. When the loop is completed we have $\mathbf{T} = \{Z_4\}$ and we continue by calling $\mathrm{PID}(y, x, P(\mathbf{V} \setminus \{Z_4\}), G[\mathbf{V} \setminus \{Z_4\}])$.

Since $\mathbf{X}$ is a singleton we end up on line 5 and find no children of X in the same C-component as X. We assume a reverse-topological ordering for the loop and begin with the latent projection $L(G, \mathbf{V} \setminus \{Z_2\})$. This projection creates a bidirected edge between X and Z_1 bringing them into the same C-component in the projection. Thus we continue with $L(G, \mathbf{V} \setminus \{Z_1\})$. This projection is also unsuitable, since Y is a child of X in the projection and there is a bidirected arc connecting them. We continue with $L(G, \mathbf{V} \setminus \{Z_3\})$ and find that X is not connected to its children via bidirected paths in the projection. Thus we set $P \leftarrow P(\mathbf{V} \setminus \{Z_3\})$, $G \leftarrow L(G, \mathbf{V} \setminus \{Z_3\})$ and $\mathbf{V} \leftarrow \mathbf{V} \setminus \{Z_3\}$. This projection is identical to the graph depicted in Figure 7(b). After these steps, only lines of the original ID algorithm are called, which results in the expression

$$\sum_{z_2, z_1} \left(\sum_{x'} P(y|z_2, z_1, x') P(x'|z_2) P(z_2) \right) P(z_1|z_2, x).$$

6. Examples on Recursive Pruning

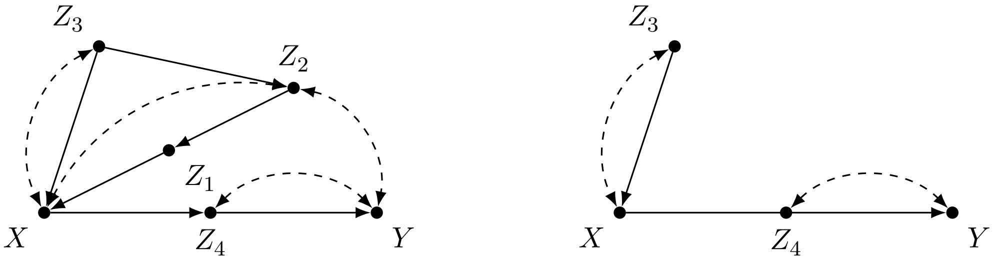
Corollaries 10, 13 and 16 often provide direct benefits when applied before the ID algorithm. The following examples show why they are also useful as recursive steps as implemented in the PID algorithm.

We are tasked with identifying the causal effect of X on Y in graph G depicted in Figure 9(a)

Initially there are no vertices that are connected to other vertices only through X. As a recursive step of the ID algorithm, we are tasked with identifying $P'_{z_3,x}(y)$ from P', where

$$P' = \sum_{z_2} P(y|z_3, z_2, z_1, x) P(x|z_3, z_2, z_1) P(z_2|z_3) P(z_3),$$

in the subgraph $G[\{Z_3, X, Y\}]$ shown in Figure 9(b). In this graph Z_3 is connected to other vertices only through X and it can be removed according to Corollary 11, since the corresponding latent projection is the subgraph $G[\{X, Y\}]$. Thus we sum out Z_3 from P'



(a) Graph G.

(b) Subgraph $G[\{Z_3, X, Y\}]$.

Figure 9: A graph for the example of recursive application of Corollary 11 within the ID algorithm.

and the resulting expression for the causal effect is

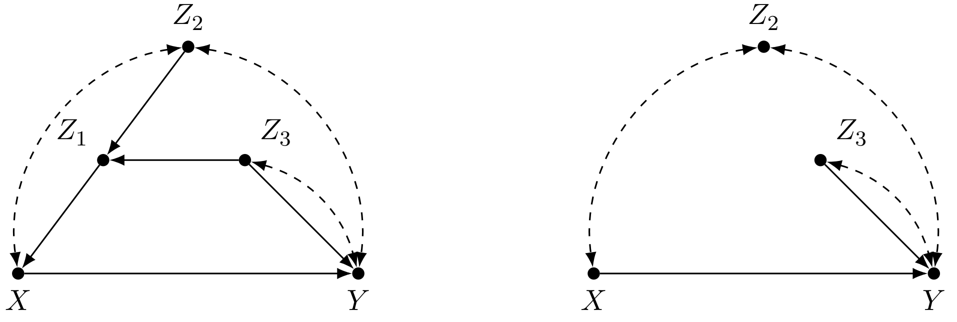
$$\sum_{z_4} \frac{\sum_{z_2,z_3} P(y|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)}{\sum_{z_2,z_3,y'} P(y'|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)}.$$

If Corollary 11 is not applied at this stage, the final expression is instead

$$\sum_{z_4} \Bigg(\frac{\sum_{z_2} P(y|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)}{\sum_{z_2,y'} P(y'|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)} \times \\ \frac{\sum_{z_2,y'} P(y'|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)}{\sum_{z_2,y',z_4} P(y'|z_3,z_2,z_1,x,z_4)P(z_4|z_3,z_2,z_1,x)P(x|z_3,z_2,z_1)P(z_2|z_3)P(z_3)} \Bigg).$$

Using Corollary 11 provides us with a simpler expression in this situation by completely removing the second term from the product inside the summation.

Next we show how Corollary 13 can also be applied at a recursive step. Our interest lies in the causal effect of X on Y in graph G of Figure 10(a)



(a) Graph G.

(b) Subgraph $G[\{Z_2, Z_3, X, Y\}]$.

Figure 10: A graph for the example of recursive application of Corollary 13 within the ID algorithm.

There are no vertices that are connected to other vertices in the graph via a single vertex. When the ID algorithm is applied we eventually reach a step where the causal effect of

$P'_{z_2,x}(y)$ is to be identified from P', where

$$P' = P(y|z_2, z_3, z_1, x)P(x|z_2, z_3, z_1)P(z_3|z_2)P(z_2),$$

in the subgraph $G[\{Z_2, Z_3, X, Y\}]$ which is depicted in Figure 10(b).

In this graph Z_3 can be removed according to Corollary 13, since Z_3 is connected to other vertices of the subgraph only through Y. The resulting distribution is obtained by summing out Z_3 from P'. The final expression for the causal effect is

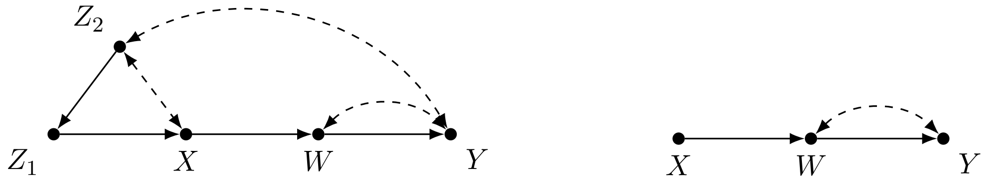
$$\frac{\sum_{z_2,z_3} P(y|z_2, z_3, z_1, x)P(x|z_2, z_3, z_1)P(z_3|z_2)P(z_2)}{\sum_{z_2,z_3,y'} P(y'|z_2, z_3, z_1, x)P(x|z_2, z_3, z_1)P(z_3|z_2)P(z_2)}. \tag{1}$$

If Corollary 13 is not applied, the resulting expression is instead

$$\sum_{z_3} \left(\frac{\sum_{z_2} P(y|z_2, z_3, z_1, x)P(x|z_2, z_3, z_1)P(z_3|z_2)P(z_2)}{\sum_{z_2,y'} P(y'|z_2, z_3, z_1, x)P(x|z_2, z_3, z_1)P(z_3|z_2)P(z_2)} \times \right.$$
$$\left. \sum_{z_2,x,y'} P(y'|z_2, z_3, z_1, x)P(x|z_2, z_3, z_1)P(z_3|z_2)P(z_2) \right).$$

As in the previous example, the benefit of applying Corollary 13 is apparent.

We can also take advantage of latent projections recursively via Corollary 16 as shown in the next example. Our interest lies in the causal effect of X on Y in graph G of Figure 11.



(a) Graph G.

(b) Subgraph $G[\{X, W, Y\}]$.

Figure 11: A graph for the example of recursive application of Corollary 16 within the ID algorithm.

Here X is connected to its child W via a bidirected path and thus they belong to the same C-component rendering Corollary 16 unusable at this time. However, as a recursive step of the ID algorithm we have to identify $P'_x(y)$ from P', where

$$P' = \sum_{z_2} P(y|z_2, z_1, x, w)P(w|z_2, z_1, x)P(x|z_2, z_1)P(z_2)$$

in the subgraph $G[\{X, W, Y\}]$. In this subgraph X is not connected to its child W via a bidirected path. We also find that the latent projection $L(G[\{X, W, Y\}], \{X, Y\})$ does not induce such a path between X and Y. We can now continue identification in this latent projection and sum out W from P'. The resulting expression for the causal effect is

$$\frac{\sum_{z_2,w} P(y|z_2, z_1, x, w)P(w|z_2, z_1, x)P(x|z_2, z_1)P(z_2)}{\sum_{z_2,w,y'} P(y'|z_2, z_1, x, w)P(w|z_2, z_1, x)P(x|z_2, z_1)P(z_2)},$$

whereas the expression without applying the corollary is instead

$$\sum_{w}\left(\frac{\sum_{z_2} P(y|z_2,z_1,x,w)P(w|z_2,z_1,x)P(x|z_2,z_1)P(z_2)}{\sum_{z_2,y'} P(y'|z_2,z_1,x,w)P(w|z_2,z_1,x)P(x|z_2,z_1)P(z_2)} \times \right.$$
$$\left.\frac{\sum_{z_2,y'} P(y'|z_2,z_1,x,w)P(w|z_2,z_1,x)P(x|z_2,z_1)P(z_2)}{\sum_{z_2,w',y'} P(y'|z_2,z_1,x,w)P(w'|z_2,z_1,x)P(x|z_2,z_1)P(z_2)}\right).$$

Additional examples are provided as an R script (R Core Team, 2017) at the JMLR online paper repository. The script also includes all of the examples presented in this paper.

An interesting question is how pruning works together with simplification presented in (Tikka and Karvanen, 2017b). We return to the example on identifying the causal effect of X on Y in the graph of Figure 10. If we apply the ID algorithm without pruning and perform simplification as a post-processing step, then the resulting expression is

$$\sum_{z_3}\frac{\sum_{z_2} P(y|z_2,z_3,z_1,x)P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2)}{\sum_{z_2} P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2)}P(z_3|z_2). \tag{2}$$

This expression is in some aspects simpler than expression (1) obtained using pruning alone, but does contain a sum over Z_3 that was not originally present.

When pruning is introduced to the ID algorithm and simplification is again applied, the resulting expression is instead

$$\frac{\sum_{z_3,z_2} P(y|z_2,z_3,z_1,x)P(x|z_2,z_3,z_1)P(z_3|z_2)P(z_2)}{\sum_{z_2} P(x|z_2,z_1)P(z_2)},$$

which is noticeably simpler than expressions (1) and (2). This example shows, that when pruning methods are employed together with simplification, a simpler expression can be reached than what is possible with either pruning or simplification alone.

7. Discussion

We have presented criteria for removing variables from causal models that are not necessary to achieve identifiability for a given causal effect, and showed how these criteria can be applied in practice. We integrated our results into a new version of the ID algorithm called PID as presented in Algorithm 2 to facilitate automatic processing of identifiability queries. It should be noted that the ID algorithm already performs some pruning such as removing non-ancestors of $\mathbf{Y}$.

The pruning operations carried out by Algorithm 2 can significantly simplify the resulting expression compared to the traditional ID algorithm. Benefits of simplification can be realized in various settings. A simpler expression is easier to understand and to evaluate, since the dimensionality of the problem has been reduced. This is especially true for settings where the expression has to be evaluated repeatedly. Simplification can also help dealing with data where some variables are affected by bias or contain missing data. Obtaining an expression that does not contain these variables has clear advantages. It may also be of interest to obtain a different expression for the same causal effect.

The choice of variable ordering on line 5 of Algorithm 2 is not arbitrary. However, available external knowledge may guide our selection to prefer certain orderings. For example, in a situation where two latent projections are mutually exclusive, we may prefer an ordering where the variable that is associated with the smallest cost, or of which we have the most accurate measurements is not considered latent.

When PID is applied in conjunction with simplification methods described in (Tikka and Karvanen, 2017b), various situations that lead to complex expressions can be taken into account. These methods complement each other, since the results in this paper deal with completely removing variables from the resulting expression, whereas the simplification methods focus on symbolic summation of so-called atomic expressions, which are expression consisting of a single sum and a number of product terms. An expression for a causal effect may consist of multiple atomic expressions, some of which can be simplified and some of which can not.

We showed via examples that our improvements are not simply pre-processing steps to be carried out before calling the ID algorithm, but actually provide significant benefits when applied recursively. As the ID algorithm manipulates the original graph it often enables the application of our results as well. As hedges characterize identifiability, it is possible to consider latent projections in a more general manner, but this is not necessarily beneficial for simplification. One could construct an algorithm that performs a search over the possible subsets of $\mathbf{V}$, and checks whether identifiability is retained in the corresponding latent projection. However, as we have shown via examples, this may not be enough to obtain a simpler expression, and the recursive structure of the ID algorithm needs to be taken advantage of. Instead, we could consider a variant of the PID algorithm, where line 5 is replaced by this procedure. However, one must be careful when applying this method, so that the computation does not become intractable when the number of vertices increases due to the complexity of the search.

Acknowledgments

We wish to thank Professor Jukka Nyblom for his comments that greatly helped to improve this paper. We also thank the anonymous reviewers for their insightful feedback. The work belongs to the profiling area "Decision analytics utilizing causal models and multiobjective optimization" (DEMO) supported by Academy of Finland (grant number 311877).

References

A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society B*, 41:1–31, 1979.

D. Geiger, T. Verma, and J. Pearl. Identifying independence in Bayesian networks. *Networks*, 20(5):507–534, 1990.

Y. Huang and M. Valtorta. Pearl's calculus of intervention is complete. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pages 217–224. AUAI Press, 2006.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82:669–688, 1995.

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

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2017. URL `https://www.R-project.org/`.

I. Shpitser and J. Pearl. Identification of joint interventional distributions in recursive semi-Markovian causal models. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, pages 1219–1226. AAAI Press, 2006.

I. Shpitser and J. Pearl. Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research*, 9:1941–1979, 2008. ISSN 1532-4435.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2nd edition, 2000.

J. Tian. *Studies in Causal Reasoning and Learning*. PhD thesis, Department of Computer Science, University of California, Los Angeles, 2002.

J. Tian and J. Pearl. A general identification condition for causal effects. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 567–573. AAAI Press, 2002.

S. Tikka and J. Karvanen. Identifying causal effects with the R package causaleffect. *Journal of Statistical Software*, 76(12):1–30, 2017a.

S. Tikka and J. Karvanen. Simplifying probabilistic expressions in causal inference. *Journal of Machine Learning Research*, 18(36):1–30, 2017b. URL `http://jmlr.org/papers/v18/16-166.html`.

T. S. Verma. Graphical aspects of causal models. Technical report, Department of Computer Science, University of California, Los Angeles, 1993. R-191.