A Characterization of Markov Equivalence Classes of Relational Causal Models under Path Semantics

Sanghack Lee and **Vasant Honavar**
Artificial Intelligence Research Laboratory
College of Information Sciences and Technology
The Pennsylvania State University, University Park, PA 16802
{sxl439, vhonavar}@ist.psu.edu

Abstract

Relational Causal Models (RCM) generalize Causal Bayesian Networks so as to extend causal discovery to relational domains. We provide a novel and elegant characterization of the Markov equivalence of RCMs under *path semantics*. We introduce a novel representation of unshielded triples that allows us to efficiently determine whether an RCM is Markov equivalent to another. Under path semantics, we provide a sound and complete algorithm for recovering the structure of an RCM from conditional independence queries. Our analysis also suggests ways to improve the orientation recall of algorithms for learning the structure of RCM under *bridge burning semantics* as well.

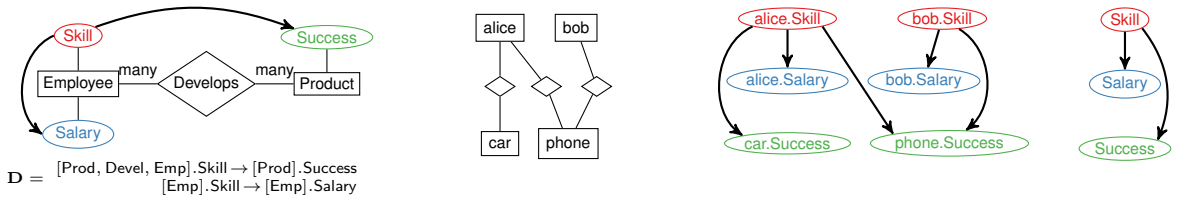
1 INTRODUCTION

The discovery of causal relationships from observational and, when available, experimental data is a central problem in artificial intelligence. Of particular interest is causal discovery in real-world settings consist of inherently interrelated entities and the resulting data exhibit a rich *relational* (Chen, 1976) structure. The past three decades have seen major advances in causal discovery (Pearl, 2000; Spirtes et al., 2000). However, the vast majority of this work has focused on Causal Bayesian Networks (CBN), directed graphical models that model causal relationships between a set of random variables of interest. Such models lack the expressive power to model causal relationships in relational domains.

Maier et al. (2010) showed that the Directed Acyclic Probabilistic Entity-Relationship model (DAPER) (Heckerman et al., 2007) which generalizes both Probabilistic Relational Models (PRM) (Friedman et al., 1999) and plate models (Buntine, 1994) is sufficient to represent causality in relational domains. Maier et al. (2010) proposed Relational PC (RPC), a relational extension of the PC algorithm (Spirtes et al., 2000) for learning the structure of Relational Causal Model (RCM), which is a particular class of DAPER, under *bridge burning semantics* (BBS). However, RPC is not complete, and is prone to erroneous orientation of edges (Maier et al., 2013a). To overcome the limitations of RPC, Maier et al. (2013a) introduced the Relational Causal Discovery (RCD) algorithm which reduces learning the structure of an RCM to learning the structure of Abstract Ground Graph (AGG, Maier et al., 2013b), a directed acyclic graph that is intended to correctly abstract the *ground instances* of the RCM, and Lee and Honavar (2016) proposed RCD-Light, a more efficient alternative to RCD. However, all of existing algorithms for learning RCM are provably *not complete* (Lee and Honavar, 2016).

Against this background, we characterize the *Markov equivalence* of RCMs, an essential step in specifying a provably complete constraint-based algorithms for learning the structure of RCM under *path semantics*, a more elegant alternative to BBS. The key idea is to show that two RCMs are Markov equivalent *if and only if* their corresponding sets of ground instances are Markov equivalent. We introduce *canonical unshielded triples,* a novel graphical construct that can be used to test the Markov equivalence of two RCMs. We provide an efficient algorithm to enumerate a subset of canonical unshielded triples of an RCM that suffice for testing whether an RCM is Markov equivalent to another. Finally, we provide an algorithm to construct a *completed partially-directed RCM*, a unique compact representation of the Markov equivalence class of an RCM.

The main contributions of this paper are: (i) a novel characterization of Markov equivalence of RCMs, using a novel representation of the relational counterparts of unshielded triples and efficient identification thereof; (ii) revelation of problematic behaviors of BBS (Maier et al., 2013a,b; Lee and Honavar, 2015, 2016), and proposal of a viable alternative, namely, *path semantics*, which is more intuitive and retains the desirable properties of BBS while avoiding its drawbacks; and (iii) the first sound and complete algorithm for learning the structure of an RCM under path semantics.



(a) RCM $\mathcal{M} = (\mathcal{S}, \mathbf{D})$ (b) relational skeleton $\sigma \in \Sigma_{\mathcal{S}}$ (c) ground graph $\mathcal{G}_{\sigma}^{\mathcal{M}}$ (d) CDG $\mathcal{G}_{\mathcal{A}}^{\mathcal{M}}$

Figure 1: An example of an RCM drawn over its underlying relational schema together with a relational skeleton, ground graph, and class dependency graph (both bridge burning and path semantics yield the same ground graph.)

2 PRELIMINARIES

We follow the notational conventions for Causal Bayesian Networks (Pearl, 2000; Spirtes et al., 2000) and the literature on RCM (Maier et al., 2013a; Lee and Honavar, 2015). A graph is specified by a set of vertices and the edges that connect them. An edge may be directed ($\rightarrow$) or undirected ($-$), but not both. A partially directed acyclic graph (PDAG) includes undirected as well as directed edges but no directed cycles. A directed acyclic graph (DAG) is a PDAG with no undirected edges. Let $\mathcal{G}$ be a PDAG and X be a vertex in $\mathcal{G}$, i.e., $X \in \mathbb{V}(\mathcal{G})$. Then, parents of X, $pa(\mathcal{G}, X)$, are vertices that have a direct edge towards X. Children (ch) are analogously defined. Neighbors (ne) of X are vertices connected to X via an undirected edge while adjacencies (adj) of X are those connected to X via an edge either directed or undirected. A *walk* on a graph is an ordered sequence of vertices where consecutive vertices in the sequence are adjacent to each other in the graph, and a *path* is a walk in which every vertex is distinct.

Relational Domain A relational domain comprises of entities that are interdependent through relationships. The specification of such relational domain is called a relational schema (schema for short). A schema, denoted by $\mathcal{S}$, is a tuple of entity classes, relationship classes, attribute classes, and cardinality constraints, denoted by $\mathcal{E}$, $\mathcal{R}$, $\mathcal{A}$, and $card$, respectively. For example, *Employee* and *Product* are entity classes in a business domain (Figure 1). *Develops* is a relationship class between them. *Employee* has *Salary* as an attribute class. Each employee may develop *multiple* products; and each product may be developed by *multiple* employees. We collectively call $\mathcal{E}$ and $\mathcal{R}$ item classes. Every item class is associated with a set of attribute classes. We denote $\mathcal{A}(I)$ a set of attribute classes associate with an item class I. A relationship class consists of participating entity classes. We denote $E \in R$ if E is a participating entity class of a relationship class R. For simplicity, we drop role indicators (as in other literature on RCM), which allow participation of an entity class in a relationship class in multiple ways. A cardinality constraint defines how many relationships an entity can participate in. Following RCM literature, $card$ is a partial function from $\mathcal{R} \times \mathcal{E}$ to $\{one, many\}$.

A relational skeleton (skeleton for short) is a particular realization of a schema, which is an undirected graph where vertices are items (i.e., instances of item classes). An edge is defined between a relationship and an entity if the entity participates in the relationship. We denote a skeleton by σ, a member of all possible skeletons $\Sigma_{\mathcal{S}}$. We denote by $\sigma(I)$ the set of items of item class I.

2.1 RELATIONAL CAUSAL MODEL

A relational causal model (RCM) (Maier et al., 2010, 2013a) consists of a set of cause-effect relationships and parameters where the cause and the effect are related in the given relational schema. For example, "the success of a product depends on the skills of employees who develop the product" is encoded as a *relational dependency*, "[Product, Develops, Employee].Skill→[Product].Success". We elaborate on each component of an RCM more precisely in what follows.

A *relational path* is an alternating sequence of entity and relationship classes. The relational path corresponds to a *walk* (with some restrictions) in the given schema where item classes are vertices and the participation of an entity class to a relationship class is an undirected edge between them. A relational path is similar to a slot chain in PRM (Friedman et al., 1999) and a first-order constraint in DAPER (Heckerman et al., 2007). The first and the last item class of a relational path is called *base* and *terminal* item class, respectively. The path explains the relation of the terminal item class from the *perspective* of the base item class. Hence the base item class is also called the *perspective*. A relational path is *canonical* if it is of unit length. A *relational variable* is a pair of a relational path and an attribute class, which belongs to the terminal item class of the path. For example, a relational variable $P.X$ consists of a path P and an attribute class X where X is an attribute class associated with the terminal item class of P. Then, an RCM $\mathcal{M} = (\mathcal{S}, \mathbf{D}, \mathbf{\Theta})$ is a set of relational dependencies $\mathbf{D}$ along with parameters $\mathbf{\Theta}$ given a schema $\mathcal{S}$. A relational dependency $P.Y \rightarrow Q.X$ consists of two relational variables as an effect and its cause where the effect relational variable is canonical, $Q = [I]$ where $X \in \mathcal{A}(I)$, and the base of P is I. To emphasize the use of canonical relational

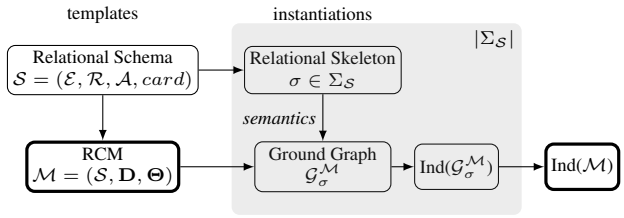


Figure 2: Schematic showing a relational schema, an RCM, and their respective instantiations, i.e., relational skeleton, and ground graphs and the independence relations of the RCM entailed from the independence relations admitted by the ground graphs.

variable, we denote a canonical relational variable with an attribute class X by $\mathcal{V}_X$.

An RCM is said to be acyclic if its *class dependency graph* $\mathcal{G}_\mathcal{A}^\mathcal{M} \triangleq (\mathcal{A}, \{Y \to X \mid P.Y \to \mathcal{V}_X \in \mathbf{D}\})$ is a DAG (see Figure 1(d)). Hence, $\mathcal{A}$ is a partially-ordered set where we denote $Z \prec_\mathcal{A} X$ if there exists a directed path from Z to X in $\mathcal{G}_\mathcal{A}^\mathcal{M}$. This implies that dependencies of the same pair of attribute classes must have the same orientation (i.e., it is impossible to have both $P.Y \to \mathcal{V}_X$ and $Q.X \to \mathcal{V}_Y$).

An RCM (or its partially directed variant) is *not* a traditional graphical model defined over relational variables: edges (i.e., relational dependencies) are only well-defined between a pair of relational variables where one of them is canonical. Hence, graphical relation (i.e., adj, ch, pa, and ne) is well-defined only if its argument is a canonical relational variable. For example, if $P.Y \to \mathcal{V}_X \in \mathcal{M}$, then $pa(\mathcal{M}, \mathcal{V}_X) = \{P.Y\}$ but $ch(\mathcal{M}, P.Y)$ is undefined if P is not canonical.

Relational d-separation An RCM defines a set of dependencies at the schema level. Given a skeleton σ, the RCM $\mathcal{M}$ is realized as a ground graph $\mathcal{G}_\sigma^\mathcal{M}$ (see Figure 1(c)), which is a DAG where vertices are attributes of items in the skeleton (e.g., $i.X$ for $X \in \mathcal{A}(I)$ of an item $i \in \sigma(I)$) and each directed edge is interpreted as a direct cause (e.g., $j.Y \to i.X$). If the RCM is an actual model of given relational data, then the ground graph will correspond to the underlying causal process that governs the attribute values of the items in the skeleton (i.e., relational data). An edge $j.Y \to i.X$ exists if there exists a dependency $P.Y \to \mathcal{V}_X \in \mathbf{D}$ such that j is *reachable* (which we will formally define in Section 3) from i along P in the skeleton. We denote by $P|_i^\sigma$ a *terminal set*, a set of reachable items from i along P in σ, which is determined according to the chosen semantics, e.g., BBS (Maier et al., 2013a; Maier, 2014). For simplicity, we drop σ if it is either unnecessary or can be inferred without ambiguity.

In RCM, we are especially interested in conditional independence between relational variables. We might ask, for example, is the success of a product independent of its developers' salaries given their skills? This conditional independence query can be represented as [Product].Success $\perp\!\!\!\perp$ [Product, Develops, Employee].Salary | [Product, Develops, Employee].Skill. If true, this implies that each product's success is independent of its developers' salary given their skills (in every company). Formally, an independence query is of the form $U \perp\!\!\!\perp V \mid \mathbf{W}$ where $\{U, V\} \cup \mathbf{W}$ is a set of relational variables of the same perspective, say $B \in \mathcal{E} \cup \mathcal{R}$. Then, the query is equivalent to checking

$$\forall_{\sigma \in \Sigma_\mathcal{S}} \forall_{i \in \sigma(B)}\ U|_i^\sigma \perp\!\!\!\perp V|_i^\sigma \mid \mathbf{W}|_i^\sigma, \tag{1}$$

in all of the instantiations of the RCM (Maier et al., 2013b) (see Figure 2). In other words, the existence of a relational skeleton $\sigma \in \Sigma_\mathcal{S}$ and a base item $i \in \sigma(B)$ such that $U|_i^\sigma \not\perp\!\!\!\perp V|_i^\sigma \mid \mathbf{W}|_i^\sigma$ in a ground graph $\mathcal{G}_\sigma^\mathcal{M}$ is the necessary and sufficient condition for $U \not\perp\!\!\!\perp V \mid \mathbf{W}$.

3 RCM SEMANTICS

We proceed to describe two alternative semantics for interpreting relational paths, and hence translating relational dependencies of an RCM into causal relationships on attributes of items of a skeleton. Let P be a relational path of n item classes. We denote the length of P by $|P|$, the reverse of the path P by $\tilde{P}$, the ℓth item class of P by P^ℓ, and the subpath of P from ℓ to m (inclusive) by $P^{\ell:m}$. We might omit the beginning or ending index if the subpath is from the beginning (i.e., prefix) or to the end of the path (i.e., suffix). i.e., $P^{:m} = P^{1:m}$ and $P^{\ell:} = P^{\ell:n}$.

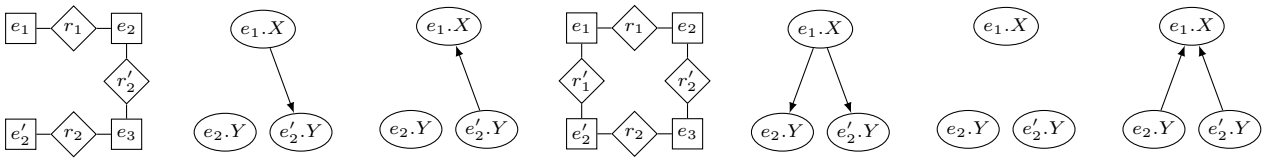
We first introduce *path semantics*, where the term path exactly means what path is defined in graph theory. Let $i \overset{P,\sigma}{\rightsquigarrow} j$ denote the fact that items i and j are connected by a path of items $\mathbf{p}$ from i to j in the given skeleton σ, where the item class of ℓth item of $\mathbf{p}$ is the ℓth item class of P for $1 \le \ell \le |P|$. Then, under path semantics, the terminal set $P|_i^\sigma$ is simply defined as,

$$P|_i^\sigma \triangleq \{j \mid i \overset{P,\sigma}{\rightsquigarrow} j\}.$$

Bridge burning semantics (BBS) (Maier et al., 2013b; Maier, 2014) computes $P|_i^\sigma$ as the set of leaves of the tree obtained by traversing the given skeleton σ along P in breadth-first order starting at i. Formally, BBS defines $P|_i^\sigma$ iteratively as $P^{:1}|_i^\sigma \triangleq \{i\}$ and

$$P^{:m}|_i^\sigma \triangleq \{k \in \sigma(P^m) \cap ne(\sigma, j) \mid j \in P^{:m-1}|_i^\sigma\} \setminus \bigcup_{\ell < m} P^{:\ell}|_i^\sigma$$

The choice of BBS has following implications, which are not fully considered in the existing RCM literature. First, given a more complex relational skeleton, BBS may yield, counterintuitively, a sparser ground graph because, as we can clearly see in the definition, if P' is a proper prefix of P and $j \in P'|_i$, then $j \notin P|_i$ even though there exists a path of items from i to j along P. Compare Figure 3(f) with 3(c). The addition of two edges e_1–r'_1–e'_2



(a) skeleton σ (b) $\mathcal{G}_\sigma^{(\mathcal{S},\{D\})}$ (c) $\mathcal{G}_\sigma^{(\mathcal{S},\{\tilde{D}\})}$ (d) skeleton σ' (e) $\mathcal{G}_{\sigma'}^{(\mathcal{S},\{D\})}$ (f) $\mathcal{G}_{\sigma'}^{(\mathcal{S},\{\tilde{D}\})}$ BBS (g) $\mathcal{G}_{\sigma'}^{(\mathcal{S},\{\tilde{D}\})}$ PS

Figure 3: Comparison of ground graphs under bridge burning semantics and path semantics. Let $\mathcal{S}$ be a schema with $\mathcal{E} = \{E_1, E_2, E_3\}$, $X \in \mathcal{A}(E_1)$, $Y \in \mathcal{A}(E_2)$, and $\mathcal{R} = \{R_1, R_2\}$ where $E_1, E_2 \in R_1$ and $E_2, E_3 \in R_2$ with cardinality greater than 1. $D = [E_2, R_2, E_3, R_2, E_2, R_1, E_1].X \rightarrow [E_2].Y$. Both semantics yield the same ground graphs (b), (c), and (e) for the relational skeleton σ. However, the two semantics yield different ground graphs (f,), (g) for $\mathcal{M} = (\mathcal{S}, \{\tilde{D}\})$ for relational skeleton σ'.

in σ' compared to σ make $e_2' \in [E_1, R_1, E_2]|_{e_1}^{\sigma'}$ and, hence, $e_2' \notin [E_1, R_1, E_2, R_2, E_3, R_2, E_2]|_{e_1}^{\sigma'}$. Second, the two RCMs that differ only with respect to the directionality of their dependencies may have different (undirected) adjacencies in their ground graphs (compare Figure 3(f) with 3(e)). This is because $j \in P|_i$ does not entail $i \in \tilde{P}|_j$ under BBS since the fact that Q is a prefix of P does not necessarily imply that $\tilde{Q}$ is a prefix of $\tilde{P}$.

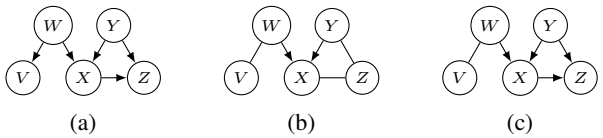
In this paper, we consider RCMs under path semantics, which is an elegant and more intuitive alternative to BBS. Further, path semantics shares the desirable properties of BBS (Maier, 2014): both semantics do not permit revisiting the base item. However, path semantics does not suffer from the counter-intuitive consequences of BBS and easier to analyze as we will see in the rest of paper.

4 MARKOV EQUIVALENCE OF RCMS UNDER PATH SEMANTICS

Recall that, in general, there can be Markov equivalent CBNs that represent a given set of independence relations (Pearl, 2000). Because RCMs are essentially relational counterparts of CBNs, it follows that there can be multiple RCMs that encode a given set of independence relations in relational domains.

Definition 1 (Markov Equivalence of RCMs)**.** Two RCMs are *Markov equivalent* if they entail the same set of relational *d*-separation conditions.

The previous attempts to characterize the Markov equivalence of RCMs under BBS (Maier et al., 2013a; Marazopoulou et al., 2015) had relied on analyses of the Abstract Ground Graph (AGG) representation of an RCM. However, Lee and Honavar (2015) have shown that AGGs cannot faithfully represent the independence relations encoded by RCMs under BBS. Consequently, the RCD algorithm (Maier et al., 2013a), which relies on AGGs to learn the structure of an RCM under BBS is *not* complete Lee and Honavar (2016). Hence, we proceed to characterize the Markov equivalence of RCMs under path semantics.



(a) (b) (c)

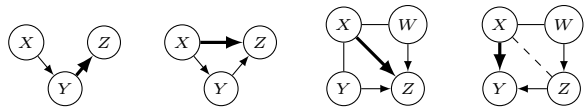
Figure 4: An example of a DAG, its pattern, and its CPDAG where (W, X, Y) is an unshielded collider and (V, W, X) and (W, X, Z) are unshielded non-colliders.

Recall that the relational d-separation $U \perp\!\!\!\perp V \mid \mathbf{W}$ in an RCM is equivalent to $U|_i \perp\!\!\!\perp V|_i \mid \mathbf{W}|_i$ for every base item i in *every* ground graph of the RCM. Hence, a sufficient condition for two RCMs to be Markov equivalent is that, for *every* relational skeleton, the corresponding sets of ground graphs of the two RCMs be Markov equivalent:

$$\forall_{\sigma \in \Sigma_\mathcal{S}} [\mathcal{G}_\sigma^\mathcal{M}] = [\mathcal{G}_\sigma^{\mathcal{M}'}] \Rightarrow [\mathcal{M}] = [\mathcal{M}'] \tag{2}$$

where $[\mathcal{M}]$ and $[\mathcal{G}]$ denote the Markov equivalence class of an RCM and a DAG $\mathcal{G}$, respectively. In Section 4.1, we will demonstrate that the converse of Equation 2 holds as well, thereby establishing a necessary and sufficient condition for two RCMs to be Markov equivalent.

Markov equivalence of DAG First, we recall the characterization of Markov equivalence of DAG (see Figure 4, Verma and Pearl, 1990; Andersson et al., 1997). Let $\mathcal{G}$ be a DAG with random variables $\mathbf{V}$ as vertices. Let X, Y, and Z be in $\mathbf{V}$. A triple (X, Y, Z) is an *unshielded triple* if both X and Z are adjacent to Y but X and Z are not adjacent. It is an *unshielded collider* if they are oriented as $X \rightarrow Y \leftarrow Z$ in the given DAG. Let $\mathcal{G}'$ be a DAG that share the same vertices of $\mathcal{G}$. Then, $\mathcal{G}$ and $\mathcal{G}'$ are said to be *Markov equivalent* if they entail identical independence relations among $\mathbf{V}$. Two DAGs are Markov equivalent if and only if their *patterns* are the same (Verma and Pearl, 1990). The *pattern* of a DAG is a PDAG where all *unshielded colliders* are oriented and the only oriented edges are unshielded colliders. A Markov equivalence class is represented by a *completed PDAG* (CPDAG or *essential graph*), a PDAG in which a directed edge $X \rightarrow Y$ implies that every DAG in the class shares the edge $X \rightarrow Y$ (*compelled edge*) while an undi-



(a) R1 (b) R2 (c) R3 (d) R4

Figure 5: Orientation rules to construct a CPDAG from a pattern and background knowledge where the orientation of a thick edge is determined by the other given edges. The edge between X and Z in R4 might be undirected or directed in any direction.

rected edge $X - Y$ implies that there exist two DAGs in the class where one has $X \to Y$ and the other has $X \leftarrow Y$ (*reversible edge*).

There are at least two systematic methods to discover the CPDAG from a pattern: The first method uses orientation rules (Meek, 1995) (Figure 5). The three rules (R1–R3) are sufficient to discover CPDAG from a pattern, and an additional rule R4 can deal with background knowledge (i.e., known orientations other than those implied by the pattern), if available. The second method exploits an algorithm for *extensibility* of a PDAG (Dor and Tarsi, 1992), which examines whether there exists a DAG which is a *consistent extension* of the PDAG, that is, the DAG shares the same sets of adjacencies, unshielded colliders, and oriented edges (if any) of the PDAG. We proceed to characterize Markov equivalence of RCM by generalizing the notions of unshielded triples, pattern, and CPDAG from the setting of CBNs to the (relational) setting of RCMs.

4.1 THE PATTERN OF AN RCM

We consider unshielded triples in ground graphs of an RCM and relate them to the RCM under path semantics. Let $i.X$, $j.Y$, and $k.Z$ be three different vertices in the ground graph $\mathcal{G}_\sigma^\mathcal{M}$ of an RCM $\mathcal{M}$ for an arbitrary skeleton $\sigma \in \Sigma_\mathcal{S}$. Then, $(i.X, j.Y, k.Z)$ is an unshielded triple in $\mathcal{G}_\sigma^\mathcal{M}$ only if $P.Y \in adj\,(\mathcal{M}, \mathcal{V}_X)$ and $Q.Z \in adj\,(\mathcal{M}, \mathcal{V}_Y)$ where $j \in P|_i^\sigma$ and $k \in Q|_j^\sigma$ for $i.X$ and $k.Z$ to be connected to $j.Y$. Furthermore, $R.Z$ must not be in $adj\,(\mathcal{M}, \mathcal{V}_X)$ for every path R such that $k \in R|_i^\sigma$ for $i.X$ and $k.Z$ to be disconnected in $\mathcal{G}_\sigma^\mathcal{M}$. Then, we define a *canonical unshielded triple* as follows:

Definition 2 (Canonical Unshielded Triple)**.** Let $\mathcal{M}$ be an RCM defined on a relational schema $\mathcal{S}$. Suppose $(i.X, j.Y, k.Z)$ is an unshielded triple (UT) in the ground graph $\mathcal{G}_\sigma^\mathcal{M}$ for some $\sigma \in \Sigma_\mathcal{S}$. There must be two (not necessarily distinct) dependencies $P.Y - \mathcal{V}_X$ and $Q.Z - \mathcal{V}_Y$ of $\mathcal{M}$ (ignoring directions) such that $j \in P|_i^\sigma$ and $k \in Q|_j^\sigma$. Then, we say that $(\mathcal{V}_X, \mathbf{P}.Y, R.Z)$ is a *canonical unshielded triple* (CUT) of $\mathcal{M}$ for every $R \in \{T \mid k \in T|_i^\sigma\}$ where $\mathbf{P} = \{T \mid j \in T|_i^\sigma\}$.

Since whenever $(i.X, j.Y, k.Z)$ is a UT in $\mathcal{G}_\sigma^\mathcal{M}$, so is $(k.Z, j.Y, i.X)$, it follows that whenever $(\mathcal{V}_X, \mathbf{P}.Y, R.Z)$ is a CUT of $\mathcal{M}$, there exists a CUT $(\mathcal{V}_Z, \mathbf{Q}.Y, \tilde{R}.X)$ for some relational paths $\mathbf{Q}$.

Theorem 3. *Two RCMs defined over the same relational schema are Markov equivalent if and only if their ground graphs are Markov equivalent for every relational skeleton of the relational schema:*

$$[\mathcal{M}] = [\mathcal{M}'] \Leftrightarrow \forall_{\sigma \in \Sigma_\mathcal{S}} [\mathcal{G}_\sigma^\mathcal{M}] = [\mathcal{G}_\sigma^{\mathcal{M}'}].$$

Proof. (If part) By the definition of relational d-separation. (Only if part) Let $[\mathcal{G}_\sigma^\mathcal{M}] \neq [\mathcal{G}_\sigma^{\mathcal{M}'}]$ for some $\sigma \in \Sigma_\mathcal{S}$. Then, the two ground graphs $\mathcal{G}_\sigma^\mathcal{M}$and $\mathcal{G}_\sigma^{\mathcal{M}'}$ differ either in their (i) adjacencies or in their (ii) unshielded colliders.
Case (i): There must exist a relational dependency $P.Y \to \mathcal{V}_X$ in $\mathcal{M}$ while both $P.Y \to \mathcal{V}_X$ and $\tilde{P}.X \to \mathcal{V}_Y$ are not in $\mathcal{M}'$ (or vice versa). Then, either $P.Y \perp\!\!\!\perp \mathcal{V}_X \mid pa(\mathcal{M}', \mathcal{V}_X)$ or $\tilde{P}.X \perp\!\!\!\perp \mathcal{V}_Y \mid pa(\mathcal{M}', \mathcal{V}_Y)$ hold in $\mathcal{M}'$ by causal Markov condition. However, both tests will be false in $\mathcal{M}$ since there exists a relational skeleton σ yielding $i.X \to j.Y$ in $\mathcal{G}_\sigma^\mathcal{M}$ where $\{P\} = \{T \mid i \in T|_j^\sigma\}$ while $P.Y \notin pa(\mathcal{M}', \mathcal{V}_X)$ and $\tilde{P}.X \notin pa(\mathcal{M}', \mathcal{V}_Y)$.
Case (ii): There must exist a CUT $(\mathcal{V}_X, \mathbf{P}.Y, R.Z)$ corresponding to an unshielded triple $(i.X, j.Y, k.Z)$, which is an unshielded collider in $\mathcal{G}_\sigma^\mathcal{M}$ and unshielded non-collider in $\mathcal{G}_\sigma^{\mathcal{M}'}$ (or vice versa). Because $R.Z \notin adj\,(\mathcal{M}, \mathcal{V}_X)$ for every $R \in \{T \mid k \in T|_i^\sigma\}$, there must exist a separating set $\mathbf{S} \subseteq adj(\mathcal{M}, \mathcal{V}_X)$ such that $\mathcal{V}_X \perp\!\!\!\perp R.Z \mid \mathbf{S}$ in $\mathcal{M}$ assuming $X \not\prec_\mathcal{A} Z$ without loss of generality.[1] By the definition of relational d-separation, $\mathbf{S}$ must be disjoint with $\mathbf{P}.Y$. However, in $\mathcal{M}'$, $\mathcal{V}_X \not\perp\!\!\!\perp R.Z \mid \mathbf{S}$ since $\mathbf{S}$ is disjoint from $\mathbf{P}.Y$, and $i.X$ and $k.Z$ are d-connected with $j.Y$ unblocked. □

We derive the definition of the *pattern* of an RCM taking into account the fact that acyclicity of an RCM is defined at an attribute class level.

Definition 4 (Pattern of RCM)**.** Let $\mathcal{M} = (\mathcal{S}, \mathbf{D})$ be an RCM and $\mathfrak{C}^\mathcal{M}$ be all canonical unshielded colliders of $\mathcal{M}$. We define the set of *attribute class level colliders* as

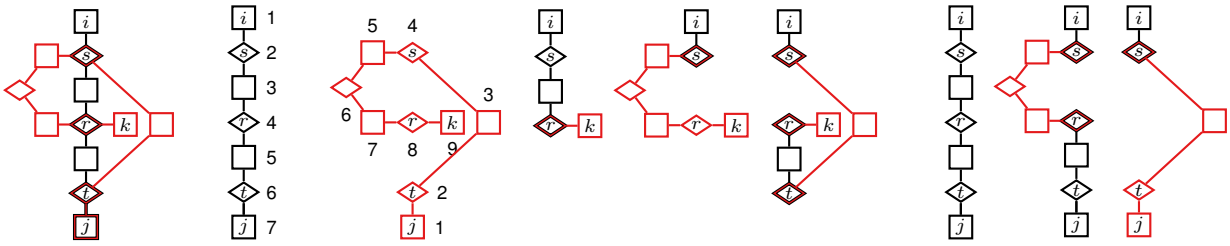
$$\mathfrak{C}_\mathcal{A}^\mathcal{M} \triangleq \{(X, Y, Z) \mid (\mathcal{V}_X, \mathbf{P}.Y, R.Z) \in \mathfrak{C}^\mathcal{M}\}.$$

Then, the *pattern* of $\mathcal{M}$, $pattern\,(\mathcal{M})$, is a partially-directed RCM $(\mathcal{S}, \mathbf{D}' \cup \mathbf{D}'')$ where $\mathbf{D}' = \{Q.X \to \mathcal{V}_Y \in \mathbf{D} \mid (X, Y, Z) \in \mathfrak{C}_\mathcal{A}^\mathcal{M}\}$ and $\mathbf{D}'' = \{P.Y - \mathcal{V}_X \mid P.Y \to \mathcal{V}_X \in \mathbf{D} \setminus \mathbf{D}'\}$.

Lemma 5. $[\mathcal{M}] = [\mathcal{M}'] \Leftrightarrow pattern(\mathcal{M}) = pattern(\mathcal{M}')$.

Proof. The proof follows from Theorem 3. □

[1] Otherwise, the proof can be obtained using $(\mathcal{V}_Z, \tilde{\mathbf{Q}}.Y, \tilde{R}.X)$.



(a) σ (b) $\mathbf{p}$ (c) $\mathbf{q}$ (d) paths of items for R_r, R_s, and R_t (e) paths of items for $\mathbf{P}$

Figure 6: Illustration of key concepts used to characterize CUTs for a hypothetical RCM $\mathcal{M}$ with $P.Y \in adj\,(\mathcal{M}, \mathcal{V}_X)$ and $Q.Z \in adj\,(\mathcal{M}, \mathcal{V}_Y)$ yielding a UT $(i.X, j.Y, k.Z)$ in $\mathcal{G}_\sigma^{\mathcal{M}}$ where P and Q correspond to item classes of $\mathbf{p}$ and $\mathbf{q}$, respectively.

Unlike in the case of DAGs, it is not immediately obvious how to identify all CUTs of an RCM. Fortunately, to discover the pattern of an RCM, it suffices to identify only one CUT from the set of CUTs for each triple of attribute classes if exists.

4.1.1 Characterization of Canonical Unshielded Triples for Pattern of RCM

How can we identify a subset of CUTs of an RCM that is sufficient to identify the pattern of the RCM? One approach is to enumerate all relational skeletons, identify the UTs in the corresponding ground graphs, and the corresponding CUTs. Because such an approach is not computationally tractable, we consider the following alternative: enumerate the skeletons that are just large enough to include a UT in the corresponding ground graph, and the corresponding CUT. We first investigate conditions under which a relational skeleton includes a UT in the corresponding ground graph, and then provide a characterization of CUTs in terms of such UTs, which leads to an efficient CUT enumeration algorithm whose time complexity is polynomial in the number of dependencies, $|\mathbf{D}|$, and $\max\{|P| \mid P.X \to \mathcal{V}_Y \in \mathbf{D}\}$ (the maximum length of dependencies, which is typically bounded by a small constant).

Relational Skeleton of an Unshielded Triple Each shielded or unshielded triple of item-attributes associates with two dependencies non-exclusively. Consider a triple $(i.X, j.Y, k.Z)$ in some skeleton $\sigma \in \Sigma_{\mathcal{S}}$. Let $P.Y \in adj\,(\mathcal{M}, \mathcal{V}_X)$ and $Q.Z \in adj\,(\mathcal{M}, \mathcal{V}_Y)$ (the two are the same if $Q.Z = \tilde{P}.X$) that admit the triple, that is, $j \in P|_i^\sigma$ and $k \in Q|_j^\sigma$. Let $\mathbf{p} = [i, \ldots, j]$ and $\mathbf{q} = [j, \ldots, k]$ be paths of items from i to j along P and j to k along Q, respectively. Since $\mathbf{p}$ and $\mathbf{q}$ must share at least one item j, there must be a non-empty set of items shared by $\mathbf{p}$ and $\mathbf{q}$. We define *anchors*, denoted by $\mathbf{J}_{\mathbf{p},\mathbf{q}}$, to be the set of pairs of indices of items shared by $\mathbf{p}$ and $\mathbf{q}$:

$$\mathbf{J}_{\mathbf{p},\mathbf{q}} \triangleq \{(a, b) \mid \mathbf{p}_a = \mathbf{q}_b\}.$$

For example, $\mathbf{J}_{\mathbf{p},\mathbf{q}} = \{(2,4), (4,8), (6,2), (7,1)\}$ in Figure 6. Anchors permit us to construct a small relational skeleton made of items for P and Q. Thus, we can enumerate the candidate anchors and verify if they are indeed anchors by constructing a relational skeleton that conforms to the equalities implied by $\mathbf{J}_{\mathbf{p},\mathbf{q}}$.

Characteristic Anchors We consider anchors that allow us to efficiently enumerate a subset of CUTs that suffice to identify the pattern of an RCM $\mathcal{M}$. We identify three special anchors (a_r, b_r), (a_s, b_s), and (a_t, b_t) among the anchors in $\mathbf{J}_{\mathbf{p},\mathbf{q}}$, and derive three relational paths R_r, R_s, and R_t from the special anchors.

Consider the item j that is the last shared item of $\mathbf{p}$ and the first shared item of $\mathbf{q}$such that $(|P|, 1) \in \mathbf{J}_{\mathbf{p},\mathbf{q}}$. Since $\mathbf{J}_{\mathbf{p},\mathbf{q}}$ is not empty, there must be a last shared item for $\mathbf{q}$ at the following anchor:

$$(a_r, b_r) \triangleq \arg\max_{(a,b) \in \mathbf{J}_{\mathbf{p},\mathbf{q}}} b.$$

No item in $\mathbf{p}_{:a_r}$ and $\mathbf{q}_{b_r:}$ is shared other than the item at the anchor (a_r, b_r) and, hence, there exists a path of items from i to k. We define $R_r \triangleq P^{:a_r} \bowtie Q^{b_r:}$ where the symbol "$\bowtie$" is a path concatenation operator (e.g., $[E_1, R_1, E_2] \bowtie [E_2, R_2] = [E_1, R_1, E_2, R_2]$). We can infer that $R_r.Z \notin adj\,(\mathcal{M}, \mathcal{V}_X)$ since $i.X$ and $k.Z$ are disconnected. Next, we define an anchor for the first shared item of $\mathbf{p}$:

$$(a_s, b_s) \triangleq \arg\min_{(a,b) \in \mathbf{J}_{\mathbf{p},\mathbf{q}}} a.$$

We characterize the given unshielded triple by considering following two cases where (a_s, b_s) is identical to (a_r, b_r) and where it is not.

Case $(a_r, b_r) = (a_s, b_s)$: A path of items corresponding to R_r is the *only* path from i to k that consists of items only in $\mathbf{p}$ and $\mathbf{q}$, and $\{R_r\} \subseteq \{T \mid k \in T|_i^\sigma\}$.

Case $(a_r, b_r) \neq (a_s, b_s)$: In a similar manner we define R_r with (a_r, b_r), we define $R_s \triangleq P^{:a_s} \bowtie Q^{b_s:}$, which satisfies $k \in R_s|_i^\sigma$. Note that $R_r = R_s$ if $P^{a_s:a_r} = Q^{b_s:b_r}$. Observing that $a_s < a_r \leq |P|$ and $1 \leq b_s < b_r$, we infer that $(|P|, 1)$ can be neither (a_r, b_r) nor (a_s, b_s).

Hence, there must exist at least three distinct anchors in $\mathbf{J}_{\mathbf{p},\mathbf{q}}$: $\{(|P|,1),(a_r,b_r),(a_s,b_s)\} \subseteq \mathbf{J}_{\mathbf{p},\mathbf{q}}$. The existence of characteristic anchors further implies that there must be an anchor (a,b) such that $a_r < a \leq |P|$ and $1 \leq b < b_s$. Among such anchors, if $\mathbf{p}_{a_r:a:-1}$ and $\mathbf{q}_{b:b_s:-1}$ do not share any items except the item at (a,b), then there exists a path of items from i to k, $\mathbf{p}_{:a_s} \bowtie \mathbf{q}_{b:b_s:-1} \bowtie \mathbf{p}_{a_r:a:-1} \bowtie \mathbf{q}_{b_r:}$, where the subpath with ": -1" represents the reverse of the subpath. There do exist such anchors:

$$(a_t, b_t) \triangleq \arg\max_{(a,b)\in\mathbf{J}_{\mathbf{p},\mathbf{q}}, a_r<a, b<b_s} b,$$

and we likewise define $R_t \triangleq P^{:a_s} \bowtie Q^{b_t:b_s:-1} \bowtie P^{a_r:a_t:-1} \bowtie Q^{b_r:}$. We call such a set of anchors, *characteristic anchors*. Given the characteristic anchors $\{(a_r,b_r),(a_s,b_s),(a_t,b_t)\} \subseteq \mathbf{J}_{\mathbf{p},\mathbf{q}}$, we retrieve three relational paths, R_r, R_s, and R_t, such that $\{R_r,R_s,R_t\} \subseteq \{T \mid k \in T|_i^\sigma\}$. See Figure 6(d) for characteristic anchors $(a_s,b_s)=(2,4)$, $(a_r,b_r)=(4,8)$, and $(a_t,b_t)=(6,2)$, and for paths of items corresponding to R_r, R_s, and R_t.

Construction of CUTs with Characteristic Anchors The characteristic anchors permit the construction of a relational skeleton σ such that the corresponding ground graph $\mathcal{G}_\sigma^\mathcal{M}$ includes a UT. First, since all triples characterized by a given characteristic anchor share common relational path(s) from i to k, the existence of a dependency $R_r.Z - \mathcal{V}_X$ (ignoring its direction) makes the triple "shielded" if $(a_r,b_r) = (a_s,b_s)$. Similarly, we can test "shieldedness" in the case of $(a_r,b_r) \neq (a_s,b_s)$ by checking $adj\,(\mathcal{M},\mathcal{V}_X) \cap \{R_r,R_s,R_t\}.Z$ is non-empty. Second, we can devise an efficient and complete procedure that (virtually) constructs a relational skeleton σ that includes an unshielded triple in $\mathcal{G}_\sigma^\mathcal{M}$. Hence, characteristic anchors can be used to identify the CUTs of an RCM without enumerating the entire set of anchors $\mathbf{J}_{\mathbf{p},\mathbf{q}}$.

We proceed to outline an algorithm (see supplementary material for details) that, given a pair of dependencies of an RCM, constructs a CUT. The algorithm initialize candidate anchors $\mathbf{J}_{\mathbf{p},\mathbf{q}}$ by checking pairs of indices (a,b) where $P^a = Q^b$. Then, the algorithm picks an anchor as (a_r,b_r), checks whether (a_r,b_r) can be (a_s,b_s) and yields a UT. Then, it outputs a CUT $(\mathcal{V}_X, \{P.Y, (P^{:a_r} \bowtie Q^{:b_r:-1}).Y\}, R_r.Z)$ where the (virtually constructed) relational skeleton σ' satisfies $\{R_r\} = \{T \mid k \in T|_i^{\sigma'}\}$ and $\{P, P^{:a_r} \bowtie Q^{:b_r:-1}\} = \{T \mid j \in T|_i^{\sigma'}\}$. If (a_r,b_r) must differ from (a_s,b_s), then the algorithm explores valid candidates for (a_s,b_s) and (a_t,b_t). If all necessary conditions are passed, then it yields a CUT from among the following: $(\mathcal{V}_X,\mathbf{P}.Y,R_r.Z)$, $(\mathcal{V}_X,\mathbf{P}.Y,R_s.Z)$, and $(\mathcal{V}_X,\mathbf{P}.Y,R_t.Z)$ where σ' satisfies $\{R_r,R_s,R_t\} = \{T \mid k \in T|_i^{\sigma'}\}$ and $\mathbf{P} = \{T \mid j \in T|_i^{\sigma'}\}$, which consists of at most six relational paths[2]. For example, paths of items in Figure 6(e) correspond to three distinct relational paths of $\mathbf{P}$.

[2] P, $P^{:a_w} \bowtie Q^{:b_w:-1}$, $P^{:a_s} \bowtie Q^{:b_s:-1}$, $P^{:a_s} \bowtie Q^{b_t:b_s:-1} \bowtie P^{a_t:}$, $P^{:a_s} \bowtie Q^{b_s:b_r} \bowtie P^{a_r:}$, and $P^{:a_s} \bowtie Q^{b_s:b_r} \bowtie P^{a_r:a_w} \bowtie Q^{:b_w:-1}$ with $a_w \triangleq a_t - \gamma + 1$ and $b_w \triangleq b_t - \gamma + 1$ where $\gamma = LLRSP(P^{a_r:a_t:-1}, Q^{:b_t:-1})$ (see Lee and Honavar, 2015).

4.2 COMPLETED PARTIALLY-DIRECTED RCM

The pattern of an RCM is a partially-directed RCM (PRCM) wherein each directed dependency is covered by some CUT of the RCM. Completed PRCM (CPRCM) is a PRCM where a dependency is directed if and only if all valid RCMs with the same pattern have the dependency oriented in the same direction as in the CPRCM. Since acyclicity of RCM is defined at the attribute class level, we orient edges on a partially-directed class dependency graph $\mathcal{G}_\mathcal{A}$ (initialized with $\mathcal{G}_\mathcal{A}^{pattern(\mathcal{M})}$) with a set of attribute class level non-colliders, denoted by $\mathfrak{N}_\mathcal{A}^\mathcal{M}$ ($\mathfrak{N}$ for short), derived from canonical unshielded non-colliders obtained as a byproduct of discovering the pattern of an RCM. Then, orientations from *completed* partially-directed CDG are used to orient undirected dependencies in the pattern of RCM resulting the CPRCM.

Given a canonical unshielded non-collider $(\mathcal{V}_X,\mathbf{P}.Y,R.Z)$, corresponding attribute class level non-collider is (X,Y,Z). It is the case that $X = Z$, that is, $(X,Y,X) \in \mathfrak{N}$. Then, we can orient as $Y \to X$, which corresponds to Relational Bivariate Orientation (RBO, Maier et al., 2013a). For simplicity, we assume that all edges of $\mathcal{G}_\mathcal{A}$ that can be oriented using RBO have been oriented, and we exclude them (e.g., (X,Y,X)) from $\mathfrak{N}$. Otherwise if $X \neq Z$, then X and Z may be connected making (X,Y,Z) *shielded*. This is why the term "unshielded" is dropped in attribute class level non-colliders. To obtain the CPRCM given the pattern of an RCM, we provide a *sound* set of rules and a *sound* and *complete* extensibility-based method. The former can be used even when the set of non-colliders is not complete whereas the latter requires a complete set of non-colliders. Before we proceed, we characterize $\mathcal{G}_\mathcal{A}^{pattern(\mathcal{M})}$ and $\mathfrak{N}_\mathcal{A}^\mathcal{M}$:

Proposition 6. *Let (X,Y,Z) be an unshielded collider in $\mathcal{G}_\mathcal{A}^\mathcal{M}$, then $X \to Y \leftarrow Z$ in $\mathcal{G}_\mathcal{A}^{pattern(\mathcal{M})}$.*

Proof. This follows from Lemma 4.4.1 in (Maier, 2014) for the existence of a triple. Since there is no dependency between X and Z, the triple must be unshielded. □

Corollary 7. *For every unshielded non-collider $(X,Y,Z) \in \mathcal{G}_\mathcal{A}^\mathcal{M}$, $(X,Y,Z) \in \mathfrak{N}_\mathcal{A}^\mathcal{M}$.*

Hence, $\mathfrak{N}$ is simply a set of non-colliders that includes all unshielded non-colliders.

Sound Rules The four rules in Figure 5 can be used to correctly orient the edges in a partially-directed CDG $\mathcal{G}_\mathcal{A}$ (Corollary 7). We provide three additional rules that make use of $\mathfrak{N}$. First, if $(X,Y,Z) \in \mathfrak{N}$ and $X \to Y$, then $Y \to Z$. This can be viewed as a generalization of R1 that

Algorithm 1 Completing a PDAG given non-colliders.

```
1: procedure completes(PDAG 𝒢, non-colliders 𝔑)
2:    U := {X→Y, Y→X}_{X−Y∈𝒢}
3:    for X→Y in U do
4:       𝒢′ := (𝒢 \ {X − Y}) ∪ {X → Y}
5:       if ∀_{V∈pa(𝒢,Y)}(X, Y, V) ∉ 𝔑 and ext(𝒢′, 𝔑) then
6:          remove edges of 𝒢′ from U
7:       else orient Y→X in 𝒢, remove Y→X from U

8: procedure ext(𝒢, 𝔑)
9:    ℋ := copy(𝒢)
10:   repeat
11:      for X in 𝕍(ℋ) such that ch(ℋ, X)=∅ do
12:         if (V1, X, V2) ∉ 𝔑 for every V1, V2 ∈ adj(ℋ, X)
13:            orient Y→X in 𝒢 for every Y∈ne(ℋ, X)
14:            ℋ := ℋ \ {X}
15:            break
16:      else return False
17:   until ℋ is empty
18:   return True
```

avoids checking unshieldedness. Second, if $(X, Y, Z) \in \mathfrak{N}$ and $X \rightarrow Z$, then $Y \rightarrow Z$. This is similar to R4 in the sense that $Y \rightarrow Z$ is a common orientation among possible orientations of a non-collider that does not create a directed cycle. Finally, we can identify a shielded collider from the fact that there must be a sink in any undirected cycle. If there exists an undirected cycle of length $n \geq 3$ where every subsequent triple in the cycle except one is non-collider, then the triple that is not a *non-collider* must be a collider. The preceding rules are clearly sound. However, without further characterization of non-colliders $\mathfrak{N}$ in a partially-directed CDG, we cannot prove that they are complete for learning the structure of an RCM.

Extensibility with Shielded Non-Colliders We generalize the algorithm for determining whether a PDAG admits an oriented extension (PDAG extensibility) (Dor and Tarsi, 1992) to work with a set of non-colliders that may be, but not necessarily, shielded. The original PDAG extensibility algorithm finds a vertex without outgoing edges where all undirected edges on the vertex can be oriented towards the vertex (i.e., sinkable) without creating new unshielded colliders. If such a vertex is found, the undirected edges between it and its neighbors are oriented towards it. The preceding steps are repeated after removing the vertex from the PDAG. The algorithm returns failure if some edges remain undirected in the PDAG and no sinkable vertex can be found. The original algorithm exploits the observation that a sinkable vertex cannot be "the middle of unshielded non-colliders", which we generalize to "the middle of non-colliders $\mathfrak{N}$". Because the unshieldedness of non-colliders plays no role in the proof of correctness of the original algorithm, the proof holds for the modified algorithm (Algorithm 1).

Theorem 8. *Let $\mathcal{G}$ be a PDAG. Let $\mathfrak{N}$ be a set of non-colliders which includes all unshielded non-colliders in $\mathcal{G}$. Then, algorithm ext correctly decides whether there exists a DAG that is a consistent extension of $\mathcal{G}$ satisfying constraints imposed by $\mathfrak{N}$.*

Proof. Let $ce(\mathcal{G}, \mathfrak{N})$ be a set of DAGs that consistently extend $\mathcal{G}$ for a given set of attribute level non-colliders $\mathfrak{N}$. Let $\mathfrak{N}(\mathcal{G}) = \{(X, Y, Z) \in \mathfrak{N} \mid \{X, Y, Z\} \subseteq \mathbb{V}(\mathcal{G})\}$ be a set of *induced* non-colliders. Whenever there exists a DAG $\mathcal{G}' \in ce(\mathcal{G}, \mathfrak{N})$, there must exist X, a sink of $\mathcal{G}$, such that $ce(\mathcal{G} - X, \mathfrak{N}(\mathcal{G} - X))$ is non-empty since $\mathcal{G}' - X$ satisfies $\mathfrak{N}(\mathcal{G} - X)$. Thus the algorithm 1 will maximally orient the PDAG and return True.

Let $\mathcal{G}''$ be a DAG in $ce(\mathcal{G} - X, \mathfrak{N}(\mathcal{G} - X))$ and $\mathcal{G}'''$ be a *reconstructed* graph $\mathcal{G}'' \cup \{X\} \cup \{Y \rightarrow X \mid Y \in ne(\mathcal{G}, X)\}$. Then, $\mathcal{G}'''$ is in $ce(\mathcal{G}, \mathfrak{N})$: (i) $\mathcal{G}'''$ is a DAG since adding a vertex as a sink to a DAG results a DAG; and (ii) $\mathcal{G}'''$ satisfies $\mathfrak{N}(\mathcal{G}) \setminus \mathfrak{N}(\mathcal{G} - X)$ since, for every reconstructed (shielded or unshielded) collider $Y \rightarrow X \leftarrow Z$, $(Y, X, Z) \notin \mathfrak{N}$ (by the definition of sinkable vertex). Therefore, ext finds a DAG in $ce(\mathcal{G}, \mathfrak{N})$ and returns True whenever $\mathcal{G}$ is extensible; and returns False otherwise. □

5 CAUSAL DISCOVERY ALGORITHM

We proceed to present RpCD, a sound and complete causal discovery algorithm for RCM under path semantics under the usual assumptions namely, causal Markov condition, sufficiency, and faithfulness (Spirtes et al., 2000), that allow us to interpret every ground graph of RCM as a CBN. We also assume access to an independence oracle that correctly answers independence queries with respect to the RCM. We further assume, as in (Maier et al., 2013a), that the maximum hop length of dependencies is known *a priori* which ensures that only a finite number of candidate dependencies need to be considered.

RpCD (see Algorithm 2) extends the key ideas of the PC algorithm (Spirtes et al., 2000) to the relational domain. Phase I of RpCD identifies adjacencies (Lines 1–11) and phase II orients the dependencies (Lines 12–23). The phase I is nearly identical to that of RCD (Maier et al., 2013a). Given a maximum hop threshold h, all candidate dependencies are enumerated. Then, spurious dependencies are removed through conditional independence tests. In Lines 12–23, it orients undirected dependencies through conditional independence tests on CUTs. Redundant tests are avoided by skipping (i) already known non-colliders (Line 15), (ii) already oriented edges (Line 16), and (iii) inactive non-colliders (Line 17). At an attribute class level, edges are oriented if forming a collider (Line 19) or forming a non-collider having the same attribute classes on its flanking elements (Line 20, RBO). All orientations that can be inferred from the sound orientation rules (see Section 4.2) are enforced (Line 22). Finally, Line 23 maximally-orients partially-directed class dependency graph with a complete

Algorithm 2 RpCD

Input: $\mathcal{S}$ schema, $\mathcal{O}$ independence tester, h hop threshold

1: initialize $\mathbf{D}$ with candidate dependencies up to h hops.
2: initialize an undirected graph $\mathcal{G}$ with undirected $\mathbf{D}$.
3: $\ell := 0$
4: **repeat**
5: **for** every ordered pair $(P.Y,\ \mathcal{V}_X)$ **s.t.** $P.Y - \mathcal{V}_X \in \mathcal{G}$ **do**
6: **for** every $\mathbf{S} \subseteq ne(\mathcal{G}, \mathcal{V}_X) \setminus \{P.Y\}$ **s.t.** $|\mathbf{S}| = \ell$ **do**
7: **if** $\mathcal{V}_X \perp\!\!\!\perp P.Y \mid \mathbf{S}$ **then**
8: remove $\{P.Y - \mathcal{V}_X, \tilde{P}.X - \mathcal{V}_Y\}$ from $\mathcal{G}$.
9: **break**
10: $\ell := \ell + 1$
11: **until** $|ne(\mathcal{G}, \mathcal{V}_X)| - 1 < \ell$ for every $X \in \mathcal{A}$
12: initialize $\mathfrak{U}$ with canonical unshielded triples from $\mathcal{G}$.
13: $\mathfrak{N} := \emptyset$, $\mathcal{H} := \langle \mathcal{A}, \{X - Y \mid P.Y - \mathcal{V}_X \in \mathcal{G}\} \rangle$
14: **for** every $(\mathcal{V}_X, \mathbf{P}.Y, R.Z) \in \mathfrak{U}$ **do**
15: **continue if** $(X, Y, Z) \in \mathfrak{N}$ **or**
16: $\{X, Z\} \cap ne(\mathcal{H}, Y) = \emptyset$ **or**
17: $\{X, Z\} \cap ch(\mathcal{H}, Y) \neq \emptyset$
18: **if** exists $\mathbf{S} \subseteq adj(\mathcal{G}, \mathcal{V}_X)$ **s.t.** $R.Z \perp\!\!\!\perp \mathcal{V}_X \mid \mathbf{S}$ **then**
19: **if** $\mathbf{S} \cap \mathbf{P}.Y = \emptyset$ **then** orient $X \to Y \leftarrow Z$ in $\mathcal{H}$
20: **else if** $X = Z$ **then** orient $Y \to X$ in $\mathcal{H}$
21: **else** add (X, Y, Z) to $\mathfrak{N}$
22: orient edges in $\mathcal{H}$ with sound rules with $\mathfrak{N}$.
23: $completes\,(\mathcal{H}, \mathfrak{N})$
24: **return** $\bigcup_{P.Y - \mathcal{V}_X \in \mathcal{G}} \begin{cases} P.Y \to \mathcal{V}_X & Y \to X \in \mathcal{H} \\ P.Y - \mathcal{V}_X & Y - X \in \mathcal{H} \end{cases}$

set of attribute class level non-colliders $\mathfrak{N}$ (except the inactive ones that play no role in the orientation of the edges). RpCD outputs undirected and directed dependencies reflecting orientations recovered from Phase II (Line 24).

Theorem 9 (Soundness and Completeness)**.** *Let $\mathcal{M}$ be an RCM whose maximum hop length of dependencies is less than or equal to h. Given access to an independence oracle and h, RpCD is sound and complete for learning the structure of the RCM under path semantics.*

Proof. The proof follows from (Maier et al., 2013a) for Phase I and, for Phase II, from the Markov equivalence of RCMs (Theorem 3) with the completeness of (i) CUTs for UTs, (ii) the CUT-enumerating algorithm for (non-)colliders ($\mathfrak{C}_{\mathcal{A}}^{\mathcal{M}}$ and $\mathfrak{N}_{\mathcal{A}}^{\mathcal{M}}$), and (iii) generalized extensibility (Theorem 8). □

Causal Discovery of RCM under BBS It is easy to see that a modification of RCD (Maier et al., 2013a) to take advantage of valid CUTs under BBS will improve the orientation recall of RCD. Given the implications of BBS (Section 3), one can check whether a CUT of an RCM under path semantics correspond to a UT in some ground graph of the RCM under BBS. The "valid" CUTs under BBS can then replace UTs of AGG (Maier et al., 2013b; Lee and Honavar, 2016) used by RCD for orientating the edges. Note that each UT of AGG is a special case of CUT where $(a_r, b_r) = (a_s, b_s)$ with $P^{a_r:} = Q^{:b_r:-1}$.

6 SUMMARY AND DISCUSSION

Relational causal models (RCM) offer an attractive approach to modeling causality in real world settings that are modeled by relational domains. Previous studies of RCM have assumed bridge burning semantics (BBS). A careful examination of RCM under BBS reveals its counter-intuitive behavior. We consider RCM under path semantics which offers a viable alternative to BBS while preserving its desirable properties while avoiding its counterintuitive consequences. We introduced canonical unshielded triples, a novel graphical construct that we use to characterize Markov equivalence of RCM under path semantics. We described RpCD, a sound and complete algorithm for recovering the structure of an RCM under path semantics from conditional independence queries. We also suggested ways to improve the orientation recall of algorithms for learning the structure of RCM under BBS.

We conclude by listing some promising directions for further research: (i) Our analysis is based on perfect independence tests. In practice, the reliability of independence tests depends on the accuracy of parametric assumption for the underlying distribution, and the quantity of available data. Many methods have been developed to make the structure learning algorithm for causal Bayesian networks (CBNs) robust to such errors including adjacency-conservative (Lemeire et al., 2012), orientation-conservative (Ramsey et al., 2006) and order-independent (Colombo and Maathuis, 2014) PC algorithms. It would be interesting to consider variants of RpCD that incorporate such approaches in the relational setting. (ii) There are variants of CBN that relax some of its restrictive assumptions (Richardson and Spirtes, 2002). Similar extensions of RCMs would be interesting to consider. (iii) It would be interesting to consider methods for estimating a spillover effect in the presence of interference (Tchetgen Tchetgen and VanderWeele, 2012) due to violation of stable unit treatment variable assumption. (iv) RCM currently does not allow class dependency graph level cycles even if such cycles are guaranteed to not introduce a cycle in any of the ground graphs of the model. For example, a person's traits are inherited from those of his/her biological parents. We can consider relaxing the acyclicity assumptions underlying RCM to permit such cycles.

Acknowledgments

The authors are grateful to UAI 2016 anonymous reviewers for their thorough reviews. This research was supported in part by the Edward Frymoyer Endowed Professorship held by Vasant Honavar and in part by the Center for Big Data Analytics and Discovery Informatics which is co-sponsored by the Institute for Cyberscience, the Huck Institutes of the Life Sciences, and the Social Science Research Institute at the Pennsylvania State University.

References

Andersson, S. A., Madigan, D., and Perlman, M. D. (1997). A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541.

Buntine, W. L. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225.

Chen, P. P.-S. (1976). The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36.

Colombo, D. and Maathuis, M. H. (2014). Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15:3921–3962.

Dor, D. and Tarsi, M. (1992). A simple algorithm to construct a consistent extension of a partially oriented graph. Technical Report R-185, Cognitive Systems Laboratory, UCLA.

Friedman, N., Getoor, L., Koller, D., and Pfeffer, A. (1999). Learning probabilistic relational models. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1300–1309. Morgan Kaufmann.

Heckerman, D., Meek, C., and Koller, D. (2007). Probabilistic entity-relationship models, PRMs, and plate models. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning*, pages 201–238. MIT Press.

Lee, S. and Honavar, V. (2015). Lifted representation of relational causal models revisited: Implications for reasoning and structure learning. In *Proceedings of the UAI 2015 Workshop on Advances in Causal Inference*, pages 56–65. CEUR-WS.

Lee, S. and Honavar, V. (2016). On learning causal models from relational data. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press.

Lemeire, J., Meganck, S., Cartella, F., and Liu, T. (2012). Conservative independence-based causal structure learning in absence of adjacency faithfulness. *International Journal of Approximate Reasoning*, 53(9):1305–1325.

Maier, M., Marazopoulou, K., Arbour, D., and Jensen, D. (2013a). A sound and complete algorithm for learning causal models from relational data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 371–380. AUAI Press.

Maier, M., Marazopoulou, K., and Jensen, D. (2013b). Reasoning about independence in probabilistic models of relational data. In *Approaches to Causal Structure Learning Workshop, UAI-13*.

Maier, M., Taylor, B., and Jensen, D. (2010). Learning causal models of relational domains. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 531–538. AAAI Press.

Maier, M. E. (2014). *Causal Discovery for Relational Domains: Representation, Reasoning, and Learning*. PhD thesis, University of Massachusetts Amherst.

Marazopoulou, K., Maier, M., and Jensen, D. (2015). Learning the structure of causal models with relational and temporal dependence. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 572–581.

Meek, C. (1995). Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 403–410. Morgan Kaufmann.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

Ramsey, J., Zhang, J., and Spirtes, P. L. (2006). Adjacency-faithfulness and conservative causal inference. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 401–408. AUAI Press.

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

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT Press.

Tchetgen Tchetgen, E. J. and VanderWeele, T. J. (2012). On causal inference in the presence of interference. *Statistical Methods in Medical Research*, 21(1):55–75.

Verma, T. and Pearl, J. (1990). Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 220–227. AUAI Press.