On Learning Causal Models from Relational Data

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

Many applications call for learning causal models from relational data. We investigate Relational Causal Models (RCM) under relational counterparts of adjacency-faithfulness and orientation-faithfulness, yielding a simple approach to identifying a subset of relational *d*-separation queries needed for determining the structure of an RCM using *d*-separation against an unrolled DAG representation of the RCM. We provide original theoretical analysis that offers the basis of a sound and efficient algorithm for learning the structure of an RCM from relational data. We describe RCD-Light, a sound and efficient constraint-based algorithm that is guaranteed to yield a correct partially-directed RCM structure with at least as many edges oriented as in that produced by RCD, the only other existing algorithm for learning RCM. We show that unlike RCD, which requires exponential time and space, RCD-Light requires only polynomial time and space to orient the dependencies of a sparse RCM.

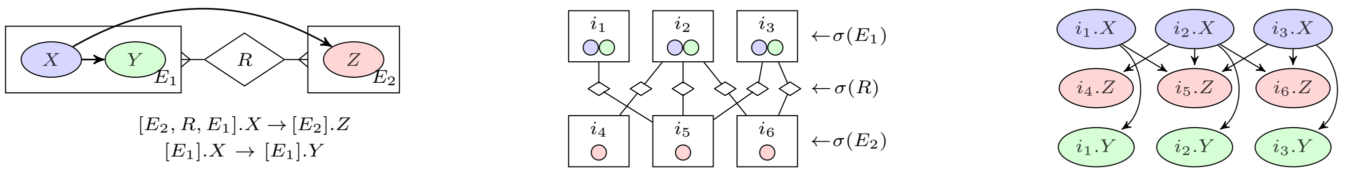
Introduction

Discovering causal relationships from observations and experiments is one of the hallmarks of intelligence. Applications of causal inference span virtually every area of human endeavor. There has been considerable progress on algorithms for eliciting causal relationships from data (Pearl 2000; Spirtes, Glymour, and Scheines 2000; Shimizu et al. 2006). Most of this work relies on Causal Bayesian Networks (CBN), directed acyclic graph (DAG)-structured probabilistic models of *propositional* data. However, in many real-world settings, the data exhibit a relational structure. Such settings call for probabilistic models of relational data (Getoor and Taskar 2007; Friedman et al. 1999; Richardson and Domingos 2006). Existing work on such models has largely focused on learning models that maximize the likelihood of the data as opposed to discovering causal relationships using independence relations from data.

It is against this background that Maier et al. (2010) introduced RPC, an extension of the PC algorithm (Spirtes, Glymour, and Scheines 2000), to the relational setting for learning causal relationships from relational data. RPC uses directed acyclic probabilistic entity-relationship (DAPER) model (Heckerman, Meek, and Koller 2007), which extends the standard entity-relationship (ER) model (Chen 1976) to incorporate probabilistic dependencies. DAPER unifies and offers more expressive power than several models of relational data, including probabilistic relational models (Getoor and Taskar 2007) and plate models (Buntine 1994). Maier et al. (2013) demonstrated the lack of completeness of RPC for learning causal models from relational data (which we will refer to as relational causal models or RCM) and introduced Relational Causal Discovery (RCD) as an alternative to RPC. RCD employs a constraint-based approach (testing conditional independencies (CI) and reasoning about them to determine the direction of causal dependencies) in an RCM. Maier, Marazopoulou, and Jensen (2013) introduced *relational d*-separation*,* the *relational* counterpart of *d*-separation (Pearl 2000) (graphical criteria for deriving CI that hold in a CBN), and introduced *abstract ground graph* (AGG), for algorithmic derivation of CI that hold in an RCM by applying (traditional) *d*-separation criteria to AGG.

The proof of correctness of RCD (Maier et al. 2013) relies on the soundness and completeness of AGG for relational *d*-separation, which in turn requires that the AGG is a DAG that represents exactly the edges that could appear in all possible ground graphs, i.e., instances of the RCM in question. However, our recent work (Lee and Honavar 2015) has called into question the completeness of AGG for relational *d*-separation: In short, there exist cases in which *d*-separation on an AGG does not yield CI that hold in the corresponding RCM. Moreover, in general, AGG can contain an infinite number of vertices and edges: It is not immediately obvious whether the practical implementation of RCD that work with a finite subgraph of the AGG inherits the theoretical guarantees of RCD based on the purported soundness and completeness of AGG for relational *d*-separation. Furthermore, RCD orients causal relationships based on the acyclicity of AGG which, in general, does not guarantee the maximal-orientedness of the resulting RCM.

Against this background, we revisit the problems of learning an RCM from relational data. The main contributions of this paper are: (i) An investigation of RCMs under two weaker faithfulness conditions (Ramsey, Zhang, and Spirtes 2006), leading to a simple approach to identifying a subset of relational *d*-separation queries that needed for learning the structure of an RCM; (ii) An original theoretical analysis that provides the basis of a provably sound al-



(a) an RCM $\mathcal{M}$ over a schema $\mathcal{S}$ (b) a skeleton $\sigma \in \Sigma_{\mathcal{S}}$ (c) a ground graph $\mathcal{GG}_{\mathcal{M}\sigma}$

Figure 1: A simple example of an RCM $\mathcal{M}$ over a schema $\mathcal{S}$, a skeleton $\sigma \in \Sigma_{\mathcal{S}}$, and a ground graph $\mathcal{GG}_{\mathcal{M}\sigma}$.

gorithm for learning the structure of an RCM from relational data; (iii) RCD-Light (RCDℓ), a sound and efficient constraint-based algorithm that, when given access to a conditional independence oracle, is guaranteed to yield a correct partially-directed RCM structure. Unlike RCD, the only other existing algorithm for learning RCM from relational data, RCDℓ does not require the construction and manipulation of AGGs, and orients dependencies more efficiently.

Preliminaries

We follow notational conventions introduced in the literature on RCM (Maier et al. 2013; Maier 2014) and on causality (Pearl 2000; Spirtes, Glymour, and Scheines 2000). An entity-relationship model (Chen 1976) abstracts the entities (e.g., employee, product) and relationships (e.g., develops) between entities in a domain using a relational schema. A skeleton is an instantiation of the schema wherein entities form a network of relationships (e.g., Quinn-develops-Laptop). Entities and relationships have attributes (e.g., salary of employees). The relationships associate with entities under cardinality constraints (e.g., many employees can develop a product). The following definitions in this section are taken from (Maier 2014):

Definition 1. A *relational schema* $\mathcal{S}$ is a tuple $\langle \mathbf{E}, \mathbf{R}, \mathbf{A}, \mathsf{card} \rangle$: a set of entity classes $\mathbf{E}$; a set of relationship classes $\mathbf{R}$; attribute classes $\mathbf{A}$ where $\mathbf{A}(I)$ is a set of attribute classes of $I \in \mathbf{E} \cup \mathbf{R}$; and cardinalities $\mathsf{card} : \mathbf{R} \times \mathbf{E} \nrightarrow \{\mathsf{one}, \mathsf{many}\}$.

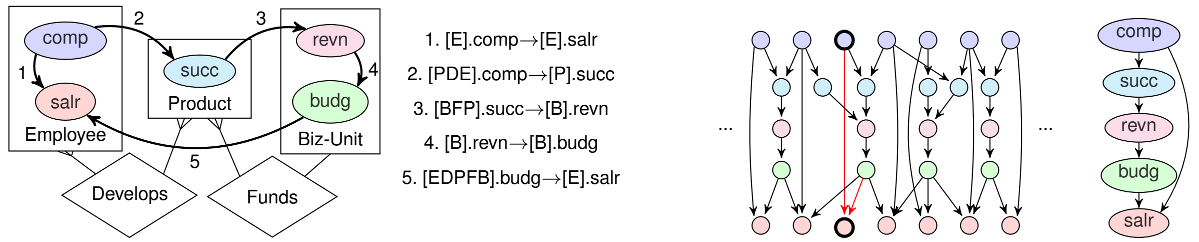
Every relationship class has two or more participating entity classes. Participation of an entity class E in a relationship class R is denoted by $E \in R$. In general, the same entity class can participate in a relationship class in two or more different roles. Although it is straightforward to introduce role indicators in the schema, for simplicity, we consider only relationship classes with distinct entity classes as in (Maier et al. 2013). We denote by $\mathbf{I}$ all item classes $\mathbf{E} \cup \mathbf{R}$. We denote by I_X an item class that has an attribute class X assuming, without loss of generality, that the attribute classes of different item classes are disjoint.

A *relational skeleton* $\sigma \in \Sigma_{\mathcal{S}}$ is an instantiation of relational schema $\mathcal{S}$ and is represented by a graph of entities and relationships where $\Sigma_{\mathcal{S}}$ represents all possible instantiations of $\mathcal{S}$. We denote by $\sigma(I)$ a set of items in σ of an item class $I \in \mathbf{I}$. The structure of skeleton and attribute values of items in the skeleton comprise a relational data, which is to be modeled by an RCM.

A *Relational Causal Model* (Maier et al. 2010), denoted by $\mathcal{M}$, consists of a set of causal relationships $\mathbf{D}$ where causes and their effects are related given an underlying relational schema $\mathcal{S}$ (see Figure 1(a)). A relational path $P = [I_j, \ldots, I_k]$ is sequence in which entity class and relationship class alternate. In the relational path P, I_j is called a *base class* or *perspective* and I_k is called a *terminal class*. A relational path corresponds to a *walk* through the schema, and shows how its terminal class is related to the base class. A relational variable $P.X$ is a pair of a relational path P and an attribute class X of the terminal class of P. A relational variable is said to be *canonical* if its relational path has length equal to 1. A relational dependency specifies a cause and its effect. Thus, relational dependency is of the form $[I_j, \ldots, I_k].Y \rightarrow [I_j].X$ i.e., its cause and effect share the same base class and its effect is canonical. For example, the success of a product depends on the competence of employees who develop the product. We represent such a relational dependency as: [Product, Develops, Employee].Competence$\rightarrow$[Product].Success.

An RCM is said to be acyclic if there is a partial order, denoted by π, over the attribute classes $\mathbf{A}$ where the order is based on cause and effect relationships in $\mathbf{D}$. An acyclic RCM does not allow dependencies that connect an attribute class to itself (similar to traditional CBN). We can parameterize an acyclic RCM $\mathcal{M}$ to obtain $\mathcal{M}_{\mathbf{\Theta}}$ by associating with the parameters $\mathbf{\Theta}$ which define the conditional distributions $\Pr([I_X].X|\mathrm{Pa}([I_X].X))$ for each attribute class X, where $\mathrm{Pa}([I_X].X)$ denotes the set of causes of $[I_X].X$. Since our focus here is on learning the structure of an RCM from relational data, we drop the parameters $\mathbf{\Theta}$ in $\mathcal{M}_{\mathbf{\Theta}}$.

A *ground graph* is an instantiation of the underlying RCM given a skeleton (see Figure 1(b) and 1(c)). It is obtained by interpreting the causes of dependencies of the RCM on the skeleton using the *terminal sets* of each of the items in the skeleton. Given a relational skeleton σ, the terminal set of a relational path P given a base item $b \in \sigma(P_1)$, denoted by $P|_b$, is items reachable from b when we traverse the skeleton along P without revisiting any items that are previously visited. The *bridge burning semantics* (Maier 2014) restricts the traversals so as not to revisit any previously visited items. Note that in order for a relational path to be valid it must yield a non-empty terminal set for some skeleton and some base item. We denote by $\mathcal{GG}_{\mathcal{M}\sigma}$ a ground graph of an RCM $\mathcal{M}$ on a skeleton σ. The vertices of $\mathcal{GG}_{\mathcal{M}\sigma}$ are labeled by pairs of items and their attributes. There exists an edge $i_j.X \rightarrow i_k.Y$ in $\mathcal{GG}_{\mathcal{M}\sigma}$ if and only if there exists a dependency $[I_k, \ldots, I_j].X \rightarrow [I_k].Y$ such that $i_j \in \sigma(I_j)$



(a) an RCM $\mathcal{M}$ over a schema $\mathcal{S}$ (b) a partial view of $\mathcal{G}_{\mathcal{M}}$ (c) a CDG $\mathcal{G}_{\pi}$

Figure 2: (a) an RCM $\mathcal{M}$ excerpted from Maier (2014). (b) a finite subgraph of $\mathcal{G}_{\mathcal{M}}$ for a base class Employee. Two thick circles correspond to canonical variables [E] .comp and [E] .salr. Two red lines correspond to dependencies [E].comp → [E].salr and [EDPFB].budg → [E].salr. (c) a class dependency graph $\mathcal{G}_{\pi}$.

is reachable from $i_k \in \sigma(I_k)$ along the relational path $[I_k, \ldots, I_j]$. Throughout this paper, unless specified otherwise, we assume a relational schema $\mathcal{S}$, a set of relational dependencies $\mathbf{D}$, and an RCM $\mathcal{M} = \langle \mathcal{S}, \mathbf{D} \rangle$. Also, we often drop 'relational' in referring to schema, skeleton, path, variables, or dependencies.

Conditional Independence of an RCM

An RCM $\mathcal{M}$ can be seen as a meta causal model defined on a schema $\mathcal{S}$. Given a skeleton σ of the schema, the RCM is instantiated into a ground graph $\mathcal{GG}_{\mathcal{M}\sigma}$, which corresponds to a DAG-structured CBN. Given the attribute values of items and the structure of the skeleton, CI that hold in the relational data (i.e., probability distribution) are equivalent to those entailed by d-separation from $\mathcal{GG}_{\mathcal{M}\sigma}$ under the causal Markov condition and the faithfulness condition (Pearl 2000; Spirtes, Glymour, and Scheines 2000). Relational counterpart of d-separation in $\mathcal{M}$ can be reduced to traditional d-separation on all of its instantiations as follows (Maier et al. 2013):

Definition 2. Let $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ be three disjoint sets of relational variables of the same perspective B defined over relational schema $\mathcal{S}$. Then, for relational model $\mathcal{M}$, $\mathbf{U}$ and $\mathbf{V}$ are *relational d-separated* by $\mathbf{W}$ if and only if, for every skeleton $\sigma \in \Sigma_{\mathcal{S}}$, $\mathbf{U}|_b$ and $\mathbf{V}|_b$ are d-separated by $\mathbf{W}|_b$ in a ground graph $\mathcal{GG}_{\mathcal{M}\sigma}$ for every $b \in \sigma(B)$.

It is non-trivial to determine whether relational d-separation holds for two reasons: i) *All-ground-graphs semantics* that relational d-separation requires d-separation to hold over all instantiations of the model $\mathcal{M}$; ii) *Intersectability* of relational variables implies that their terminal sets may have a non-empty overlap. Maier, Marazopoulou, and Jensen (2013) devised an abstract ground graph (AGG) and a mechanism that answers a relational d-separation query against an RCM by reducing it to a d-separation query against the corresponding AGG. However, as already noted, we have shown that AGG is, in general, *not* complete for relational d-separation (Lee and Honavar 2015). Hence, we proceed to propose a new DAG representation for an RCM, which serves to represent and reason about the CI that hold in an RCM for the purpose of learning the structure of an RCM from relational data.

Unrolled Graph of an RCM

We *unroll* an RCM to obtain a DAG which allows us to make explicit, the CI that are implicit in an RCM. The resulting DAG allows us to appeal to graph-theoretic notions, e.g., parents, children, descendants, unshielded triples, (traditional) d-separation, etc., to characterize relational d-separation in the RCM in terms of (traditional) d-separation in a DAG-structured probabilistic model (see Figure 2(b)):

Definition 3 (Unrolled Graph)**.** Given an RCM $\mathcal{M} = \langle \mathcal{S}, \mathbf{D} \rangle$, an *unrolled graph* of $\mathcal{M}$ is a DAG denoted by $\mathcal{G}_{\mathcal{M}}$ where vertices are all relational variables of $\mathcal{S}$, and an edge $P.X \rightarrow Q.Y$ exists if and only if there exists a dependency $R.X \rightarrow [I_Y].Y \in \mathbf{D}$ such that $P \in \mathsf{extend}(Q, R)$.

The function extend (Maier, Marazopoulou, and Jensen 2013) yields relational paths that cover some of possible concatenations of Q and R (a formal definition in Appendix). For example, extending $[E_1, R, E_2]$ and $[E_2, R, E_1]$ yields $\{[E_1, R, E_2, R, E_1], [E_1]\}$ if E_2 participates in R with many cardinality. The first path will not be included if E_2 participates in R with one cardinality since valid paths should yield a non-empty terminal set under bridge burning semantics. Note that $[E_1, R, E_1]$ is not a valid relational path. The unrolled graph is a graph union of AGGs from each perspective without including intersection variables and intersection variable edges, which are known to be imprecisely defined (Lee and Honavar 2015).

Weak Faithfulness of $\mathcal{M}$ with Respect to $\mathcal{G}_{\mathcal{M}}$ A probability distribution over a set of variables is said to be *faithful* to a DAG over the same set of variables if and only if every CI valid in the probability distribution is *entailed by* the DAG. It is easy to show that an RCM $\mathcal{M}$ is *not faithful* to its AGGs (Lee and Honavar 2015). However, we will show that an RCM $\mathcal{M}$ satisfies two weaker notions of faithfulness, namely, *adjacency-faithfulness* and *orientation-faithfulness* (Ramsey, Zhang, and Spirtes 2006) with respect to its unrolled graph $\mathcal{G}_{\mathcal{M}}$, thereby setting the stage for a new, provably correct algorithm for learning a partially-directed structure of an RCM from relational data. We denote conditional independence by '$\perp$' in general (e.g., RCM or probability distribution). We use '$\perp\!\!\!\perp$' to represent (traditional) d-separation on a DAG (e.g., $\mathcal{G}_{\mathcal{M}}$ or $\mathcal{GG}_{\mathcal{M}\sigma}$). Furthermore, we use a subscript to specify, if necessary, the scope of CI. Let $\mathcal{V}_B$ be all relational variables of base class B.

Lemma 1 (Adjacency-Faithfulness)**.** *Let U, V be two relational variables of the same perspective B. If U, V are adjacent in $\mathcal{G}_{\mathcal{M}}$, then they are dependent conditional on any subset of $\mathcal{V}_B \setminus \{U, V\}$.*

Proof. Let $\mathbf{W} \subseteq \mathcal{V}_B \setminus \{U, V\}$. Let $U \to V$ be an edge in $\mathcal{G}_{\mathcal{M}}$, which is due to a dependency $D \in \mathbf{D}$. We can prove that $(U \not\perp V \mid \mathbf{W})_{\mathcal{M}}$ by constructing a skeleton $\sigma \in \Sigma_{\mathcal{S}}$ where $\mathcal{GG}_{\mathcal{M}\sigma}$ satisfies $(U|_b \not\perp V|_b \mid \mathbf{W}|_b)_{\mathcal{GG}_{\mathcal{M}\sigma}}$. Maier (2014) described a method to construct a skeleton (Lemma 4.4.1) to only to represent U and V with respect to D. This ensures that: i) $\{u\} = U|_b$, $\{v\} = V|_b$, and $\{u\} = D|_v$ are singletons; ii) $u \neq v$; and iii) $\mathbf{W}|_b \cap \{u, v\} = \emptyset$. Since $\mathcal{GG}_{\mathcal{M}\sigma}$ contains $u \to v$, and both u and v cannot be conditioned by $\mathbf{W}$, it satisfies $(U|_b \not\perp V|_b \mid \mathbf{W}|_b)_{\mathcal{GG}_{\mathcal{M}\sigma}}$. ☐

The following lemma deals with the orientation of a pair of dependencies that form an unshielded triple. We refer to a triple of vertices $\langle U, V, W\rangle$ in a DAG as an *unshielded triple* if both U and W are connected to V but are disconnected from each other. An unshielded triple of the form $U \to V \leftarrow W$, is called an unshielded collider.

Lemma 2 (Orientation-Faithfulness)**.** *Let U, V, and W be distinct relational variables of the same perspective B and $\langle U, V, W\rangle$ be an unshielded triple in $\mathcal{G}_{\mathcal{M}}$ where U and W are not intersectable.*
(O1) *if $U \to V \leftarrow W$, then U and W are dependent given any subset of $\mathcal{V}_B \setminus \{U, W\}$ that contains V.*
(O2) *otherwise, U and W are dependent given on any subset of $\mathcal{V}_B \setminus \{U, V, W\}$.*

Proof. The method in Lemma 1 can be modified to construct a skeleton σ for U, V, and W. One can add unique items for W, which are not already a part of items for V. The resulting skeleton σ for U, V, and W guarantees that no $T \in \mathcal{V}_B \setminus \{U, V, W\}$ can represent any items in $\{u, v, w\} = \{U, V, W\}|_b$. Then, the resulting ground graph $\mathcal{GG}_{\mathcal{M}\sigma}$ has an unshielded triple of items $\{u, v, w\}$ with directions corresponding to those between U, V, and W in $\mathcal{G}_{\mathcal{M}}$. Hence, the existence (or absence) of V in the conditional determines dependence for O1 (or O2) in $\mathcal{GG}_{\mathcal{M}\sigma}$. ☐

Note, however, that orientation-faithfulness does not imply whether every unshielded triple in ground graphs can be represented as an unshielded triple of relational variables. Adjacency- and orientation-faithfulness of RCM with respect to its unrolled DAG, provides a sound basis for answering relational *d*-separation queries against an RCM.

Learning an RCM

The preceding results set the stage for an algorithm for correctly identifying undirected dependencies and orienting them through unshielded colliders in the DAG representation. Let D be $P.X \to [I_Y].Y$. The reverse of P is denoted by $\tilde{P}$. We denote $\tilde{P}.Y \to [I_X].X$ by $\tilde{D}$, which is a dependency of an opposite direction. A dependency is said to be *undirected* if both D and $\tilde{D}$ are considered valid candidates.

We will use an accent *hat* to differentiate an intermediate varying structure (e.g., $\hat{\mathbf{D}}$ and $\hat{\mathcal{M}}$) from the true structure. A graph is called partially directed acyclic graph (PDAG) if edges are either undirected or directed and there is no directed cycle. We denote $X \prec Y$ if there exists a directed path from X to Y in an underlying (P)DAG or, similarly, if X precedes Y in the given partial order (e.g., $X \prec_\pi Y$ for a partial order π). A function Pa is a set of parents of given vertices in an underlying (P)DAG. In the context of an RCM, Pa is a set of causes for a canonical variable, which is identical to the use of Pa in its corresponding unrolled graph. We often specify the scope using a superscript (i.e., $\mathrm{Pa}^{\mathcal{G}_{\mathcal{M}}}$) when it is not obviously inferred from the context. We provide a proposition that minimally-generalizes the existence of a separating set to a relational setting.

Proposition 1 (Existence of a Separating Set)**.** *Let $[B].X$ and $Q.Y$ be two different relational variables of the same perspective B where Y is not a descendant of X in the partial order of attribute classes induced from RCM $\mathcal{M}$. Then, $[B].X$ and $Q.Y$ are relational d-separated by Pa $([B].X)$ if and only if $Q.Y \to [B].X$ or $\tilde{Q}.X \to [I_Y].Y$ is not in $\mathcal{M}$.*

Phase I: Identifying Undirected Dependencies

We first identify all undirected dependencies. Recall that CI-based algorithms for learning the structure of a causal model start by enumerating all possible candidate dependencies (Spirtes, Glymour, and Scheines 2000). Unlike in the propositional setting where the number of variables is fixed and finite, the number of relational variables is, in general, infinite. It is therefore impossible to enumerate all possible dependencies for learning the structure of an RCM. Hence, as in (Maier et al. 2013), we assume that the number of dependencies in the RCM to be learned is finite and that the maximum number of hops (i.e., path length) of dependencies, denoted by h, is known a *priori*. This allows us to enumerate candidate dependencies that include all true dependencies (Maier et al. 2013). Then, we can identify and orient true undirected dependencies among the candidates.

Lemma 3. *Let D be $P.X \to [I_Y].Y$. Then, $(P.X \perp [I_Y].Y \mid Pa([I_Y].Y))_{\mathcal{M}}$ or $(\tilde{P}.Y \perp [I_X].X \mid Pa([I_X].X))_{\mathcal{M}}$ if and only if both D and $\tilde{D}$ are not in $\mathbf{D}$.*

Proof. (If) In $\mathcal{G}_{\mathcal{M}}$, there is no edge between $P.X$ and $[I_Y].Y$ and $\tilde{P}.Y$ and $[I_X].X$ by definition of extend. By Proposition 1, a separating set exists for at least one of the two CI tests since either $X \not\prec_\pi Y$ or $Y \not\prec_\pi X$. (Only if) It follows from adjacency-faithfulness (Lemma 1). ☐

Phase II: Orienting Dependencies Using CI Tests

Let $\mathcal{G}_{\hat{\mathcal{M}}}$ be a partially directed unrolled graph from $\hat{\mathcal{M}} = \langle \mathcal{S}, \hat{\mathbf{D}}\rangle$ where $\hat{\mathbf{D}} = \{D, \tilde{D}\}_{D \in \mathbf{D}}$ after Phase I (currently, no edge is directed). We orient the undirected dependencies that correspond to unshielded triples using Lemma 2. The following lemma shows how to perform collider detection:

Lemma 4 (Collider Detection)**.** *Let $\langle U, V, W\rangle$ be an unshielded triple in $\mathcal{G}_{\hat{\mathcal{M}}}$. If a separating set $\mathbf{S}$ exists such that $(U \perp W \mid \mathbf{S})_{\mathcal{M}}$ and $V \notin \mathbf{S}$, then $U \to V \leftarrow W$ in $\mathcal{G}_{\mathcal{M}}$.*

Unfortunately, since $\mathcal{G}_{\hat{\mathcal{M}}}$ is an infinite graph, we cannot naively apply the collider detection (CD) on $\mathcal{G}_{\hat{\mathcal{M}}}$. Fortunately, we can prove that for each unshielded triple in $\mathcal{G}_{\hat{\mathcal{M}}}$, there exists a *representative unshielded triple* (see Figure 2(b), red lines correspond to a representative unshielded triple), such that orienting the representative unshielded triples is equivalent to orienting all unshielded triples in $\mathcal{G}_{\hat{\mathcal{M}}}$.

Lemma 5 (Representative Unshielded Triple)**.** *Let $\langle P'.X,\, Q'.Y,\, R'.Z\rangle$ be an unshielded triple in $\mathcal{G}_{\hat{\mathcal{M}}}$ where X can be Z. Then, there exists a representative unshielded triple $\langle [I_X].X,\, Q.Y,\, R.Z\rangle$ in $\mathcal{G}_{\hat{\mathcal{M}}}$.*

Proof. see Appendix. □

The lemma frees us from the need to check whether two flanking elements of a given triple are non-intersectable (see Lemma 2): Because of bridge burning semantics, a canonical variable is not intersectable with any other relational variables of the same base class. The existence of representative unshielded triples permits us to orient relational dependencies in unshielded colliders of the RCM, unlike RCD, without needing search for the unshielded triples over AGGs. We can enumerate all representative unshielded triples totaling $O(|\mathbf{D}|^2\, h)$. We now proceed to pull together Lemma 5 and Proposition 1:

Corollary 1. *Let $\langle [I_X].X, Q.Y, R.Z\rangle$ be an unshielded triple in $\mathcal{G}_{\hat{\mathcal{M}}}$ where $X \not\prec_\pi Z$. Then, $([I_X].X \perp R.Z \mid Pa([I_X].X))_{\mathcal{M}}$.*

Since the existence of an unshielded collider $\langle [I_X].X,\, Q.Y,\, R.Z\rangle$ implies (by construction) the existence of another unshielded collider $\langle [I_Z].Z,\, \tilde{D}_2.Y,\, \tilde{R}.X\rangle$ where $D_2.Z \to [I_Y].Y$ in $\hat{\mathbf{D}}$ and $R \in \mathsf{extend}(Q, D_2)$, one can always orient dependencies between X and Y and X and Z without regard to $X \not\prec_\pi Z$.

Lemma 5 indirectly indicates that RCD is not complete. There might exist an unshielded triple $\langle i.X, j.Y, k.Z\rangle$ in some ground graph although no representative unshielded triple exists in $\mathcal{G}_{\mathcal{M}}$. Lemma 5 shows that no unshielded triple in $\mathcal{G}_{\mathcal{M}}$ or AGGs represents $\langle i.X, j.Y, k.Z\rangle$. Example 1 in Appendix shows that RCD is not complete because it fails to identify some of such unshielded colliders.

Phase III: Maximally-Orienting the Dependencies

Phase II not only orients dependencies that form unshielded colliders in the unrolled DAG representation, but also creates constraints on the pairs of dependencies that form unshielded triples, which turn out to be unshielded non-colliders. We maximally-orient the undirected dependencies based on these constraints, together with the acyclicity of RCM. There are systematic ways for finding a maximally-oriented graph in the case of conventional causal Bayesian networks (Dor and Tarsi 1992; Meek 1995; Chickering 1995), including methods that leverage background knowledge (Meek 1995). It is appealing to consider applying known rules (Meek 1995) on the unrolled graph. However, direct application of such rules is not feasible since the unrolled graph is an infinite graph. Furthermore, it is not obvious how to apply such rules on the unrolled graph whose vertices are relational variables while the acyclicity of RCM is defined at the level of attribute classes.

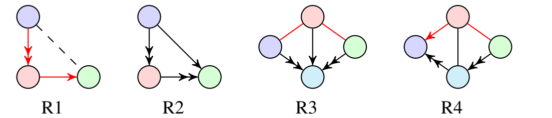


Figure 3: Orientation rules for $\mathcal{G}_{\hat{\pi}}$. Double arrow edges represent given oriented edges and single arrow edges are inferred by each rule. Dashed edge in R1 might be present or not. A pair of red edges represents a non-collider.

Hence, we translate the information expressed using relational variables (i.e., dependencies and unshielded non-colliders) into information described using the attribute classes. We first represent unshielded non-colliders (e.g., $\langle [I_X].X,\, Q.Y,\, R.Z\rangle$) as attribute class non-colliders (e.g., $\langle X, Y, Z\rangle$) when $X \neq Z$. If $X = Z$, we can immediately orient every dependency between X and Y to $Y \to X$ given the non-collider constraints and acyclicity. This corresponds to Relational Bivariate Orientation (RBO) (Maier et al. 2013). Let $\hat{\mathcal{M}} = \langle \mathcal{S}, \hat{\mathbf{D}}\rangle$ where $\hat{\mathbf{D}}$ reflects orientations through collider detection and orientations from unshielded non-colliders with the same attribute class (RBO). We then introduce a *class dependency graph* $\mathcal{G}_{\hat{\pi}}$ over $\mathbf{A}$ (see Figure 2(c) for a true graph $\mathcal{G}_\pi$), a PDAG that represents $\hat{\pi}$, an inferred partial order of $\mathbf{A}$. There exists an edge in $\mathcal{G}_{\hat{\pi}}$ between X and Y if there exists a dependency between X and Y. It is directed as $X \to Y$ if there exists an oriented dependency $P.X \to [I_Y].Y$ in $\hat{\mathbf{D}}$. Otherwise, it is undirected.

Characterization of Class Dependency Graph It is not immediately obvious whether applying the rules (Meek 1995) on $\mathcal{G}_{\hat{\pi}}$ will yield a maximally-oriented $\mathcal{G}_{\hat{\pi}}$ since $\mathcal{G}_{\hat{\pi}}$ and attribute class non-colliders, denoted by $\boldsymbol{\mathcal{N}}$, do not directly match the conditions under which the completeness of the rules is proved: i) All unshielded colliders are oriented in a PDAG; ii) There exists a set of known oriented edges, constituting background knowledge, denoted by $\boldsymbol{\mathcal{K}}$; and iii) All known non-colliders are unshielded in the PDAG.

Hence, we proceed to characterize $\mathcal{G}_{\hat{\pi}}$ and $\boldsymbol{\mathcal{N}}$ with respect to these conditions. First, $\mathcal{G}_{\hat{\pi}}$ satisfies the first condition that all edges involved in unshielded colliders in $\mathcal{G}_\pi$ are correctly oriented in $\mathcal{G}_{\hat{\pi}}$. It is possible, by construction, that an unshielded collider of $\mathcal{G}_{\hat{\mathcal{M}}}$, e.g., $\langle [I_X].X,\, Q.Y,\, R.Z\rangle$, can be shielded in $\mathcal{G}_{\hat{\pi}}$, e.g., $X \to Y \leftarrow Z$ where X and Z are adjacent. Such a shielded collider is treated as two oriented edges as a part of background knowledge $\boldsymbol{\mathcal{K}}$. If an unshielded collider has the same attribute class on the flanking elements of the triple, e.g., $\langle [I_X].X,\, Q.Y,\, R.X\rangle$, then $X \to Y$ can be regarded as $\boldsymbol{\mathcal{K}}$. Note that it is a shielded triple in $\mathcal{G}_\pi$ that Phase II might fail to orient. Second, every edge oriented by RBO is also a part of $\boldsymbol{\mathcal{K}}$. Finally, let us examine $\boldsymbol{\mathcal{N}}$ under the third condition. It may be possible that X and Z can be adjacent in $\mathcal{G}_{\hat{\pi}}$ for some $\langle X, Y, Z\rangle \in \boldsymbol{\mathcal{N}}$, which violates the third condition. We prove that such shielded attribute class non-colliders can be encoded as background knowledge.

Lemma 6. *Let $\mathcal{G}_{\hat{\pi}}$ be a PDAG as defined above. Let X, Y,* *and Z be three different attribute classes connected to each other in $\mathcal{G}_{\hat{\pi}}$. If $\langle X, Y, Z\rangle$ is a non-collider, then either an edge between X and Y or Y and Z is oriented.*

Proof. Since $\langle X, Y, Z\rangle$ is a shielded non-collider, there must be an unshielded triple $\langle P'.X, Q'.Y, R'.Z\rangle$. Let $D_1.Y - [I_X].X$ and $D_2.Y - [I_Z].Z$ be two corresponding undirected dependencies (directionality does not affect the proof) in $\mathbf{D}$. Since $\langle X, Y, Z\rangle$ is shielded, there must be a dependency $D_3.Z - [I_X].X \in \mathbf{D}$. By lemma of representative unshielded triple, there must be a representative unshielded triple $\langle [I_X].X, D_1.Y, R.Z\rangle$ such that $R \neq D_3$.

(If $D_3.Z - D_1.Y$) $\langle D_3.Z, D_1.Y, R.Z\rangle$ forms an unshielded triple, and $Z - Y$ will be oriented by either CD or RBO with its representative unshielded triple.

(Otherwise) Because of the dependency $D_2.Y - [I_Z].Z$, there must exist $Q.Y$ such that $D_3.Z - Q.Y, Q \in D_3 \bowtie D_2$, and $Q \neq D_1$. Consider the following cases.

- If $Q.Y - [I_X].X$, then $\langle Q.Y, [I_X].X, D_1.Y\rangle$ is an unshielded triple, and $X - Y$ will be oriented by CD or RBO.
- Otherwise, following $\langle Q.Y, D_3.Z, [I_X].X\rangle$, $\langle D_3.Z, [I_X].X, D_1.Y\rangle$, and $\langle [I_X].X, D_1.Y, R.Z\rangle$ are unshielded triples, and one of them is an unshielded collider, which must be oriented by CD.

Finally, either an edge between X and Y or Y and Z must be oriented for any shielded non-collider $\langle X, Y, Z\rangle$. □

For a non-collider $\langle X, Y, Z\rangle$, we can orient $Y \to Z$ if $X \to Y$ (i.e., colliding to Y). If an oriented edge is $Y \to X$ (i.e., diverging from Y), then the non-collider constraint is inactive and the other edge $Y – Z$ can be oriented in either direction. This implies that all shielded non-colliders in $\mathcal{G}_{\hat{\pi}}$ can be encoded in background knowledge as either two oriented edges or one edge oriented in a direction pointing away from Y. Finally, given a sound, but not necessarily complete, list of non-colliders, the four rules suffice to maximally-orient the rest of undirected dependencies in a partially-directed RCM resulting from Phase II. Figure 3 shows the four rules (Meek 1995) where R1 is generalized so that it can orient edges for shielded non-colliders as well (Lemma 6).

RCDℓ, a Sound Algorithm for RCM

The preceding results provide a sound theoretical basis for identifying and orienting dependencies in an RCM. We proceed to describe RCDℓ (Algorithm 1), a sound algorithm for learning an RCM. Lines 1–8 enumerate candidate dependencies and refine them with CI tests increasing the size of separating sets. Lines 9–20 test whether a representative unshielded triple is a collider or not, orient if it is either a collider or orientable by RBO, or record as a non-collider, otherwise. RCDℓ minimizes the number of CI tests by simultaneously orienting edges of the class dependency graph using four rules in Line 10 and 20. Lines 12–14 serve to avoid unnecessary tests and Line 19 serves to record the ancestral relationship between X and Z based on Proposition 1. Finally, Line 21 orients dependencies based on the orientation of the class dependency graph. Because the four orientation

Algorithm 1 RCDℓ: RCD-Light

Input: $\mathcal{S}$: a relational schema; p: a distribution; h: a maximum hop threshold; $\boldsymbol{\mathcal{K}}$: background knowledge
Output: a correct partially-directed relational causal model

1: initialize $\mathbf{D}$ with candidate dependencies up to h
2: $d = 0$
3: **repeat**
4:     **for** $D = U \to V$ in $\mathbf{D}$ **do**
5:         **if** $(U \perp V \mid \mathbf{S})_p$ **s.t.** $\mathbf{S} \subseteq \mathrm{Pa}(V) \setminus \{U\}$, $|\mathbf{S}| = d$ **then**
6:             remove $\{D, \tilde{D}\}$ from $\mathbf{D}$
7:     $d = d + 1$
8: **until** $|\mathrm{Pa}(V)| < d$ for every $U \to V \in \mathbf{D}$
9: $\boldsymbol{\mathcal{N}} = \emptyset$; initialize $\mathcal{G}_\pi$ with $\mathbf{D}$ and $\boldsymbol{\mathcal{K}}$
10: apply orientation rules on $\mathcal{G}_\pi$ with $\boldsymbol{\mathcal{N}}$ if $\boldsymbol{\mathcal{K}}$ is non-empty.
11: **for** representative unshielded triple $\langle [I_X].X, Q.Y, R.Z\rangle$ **do**
12:     **if** $X \prec_\pi Z$, **then continue**
13:     **if** $X - Y$ and $Y - Z$ are oriented, **then continue**
14:     **if** $\langle X, Y, Z\rangle$ in $\boldsymbol{\mathcal{N}}$ or $X \leftarrow Y$ or $Y \to Z$, **then continue**
15:     **if** $([I_X].X \perp R.Z \mid \mathbf{S})_p$ **s.t.** $\mathbf{S} \subseteq \mathrm{Pa}([I_X].X)$ **then**
16:         **if** $Q.Y \notin \mathbf{S}$ **then** orient $X \to Y$, $Z \to Y$
17:         **else if** $X = Z$ **then** orient $Y \to X$
18:         **else** add $\langle X, Y, Z\rangle$ to $\boldsymbol{\mathcal{N}}$
19:     **else** set $X \to Z$
20:     apply orientation rules on $\mathcal{G}_\pi$ with $\boldsymbol{\mathcal{N}}$
21: orient $\mathbf{D}$ as directed in $\mathcal{G}_\pi$
22: **return** $\langle \mathcal{S}, \mathbf{D}\rangle$

rules suffice to maximally-orient the rest of undirected dependencies given a list of non-colliders, RCDℓ produces a partially directed RCM with at least as many edges oriented as in that produced by RCD. It is easy to prove that RCDℓ requires time that is a polynomial function of $|\mathbf{D}|$ and h for orientation of an RCM of fixed degree where the degree of an effect is the number of its causes and the degree of an RCM is the degree of the effect with the largest degree.

Theorem 1. *Given access to the conditional independence oracle for an RCM $\mathcal{M}$, RCDℓ offers a sound procedure for learning the structure of the RCM whose maximum number of hops of dependencies is bounded by h.*

Proof. This follows from: (i) Lemma 3 for identifying correct undirected dependencies; (ii) Corollary 1 for sound orientation of undirected dependencies through CI tests; and (iii) Lemma 6 and the soundness and completeness of four rules (Meek 1995) for maximally-orienting undirected dependencies given a set of non-colliders. □

Empirical Comparison to RCD on Synthetic Models We compared the time and space used by RCD and RCDℓ (built on RCD codebase) on 100 RCMs with h ranging from 1 to 4. We generated schemas with 3 entity and 3 binary relationship classes with 2 and 1 attribute classes per entity and relationship class, respectively, with random cardinality. Given the schema, we generated an RCM with 10 dependencies of length up to h and maximum degree of 3. We followed settings in Maier et al. (2013): (i) RCD uses AGGs whose hop length is limited to $2h$ for practical reasons; and (ii) AGGs with $2h$ is adopted as a CI oracle.

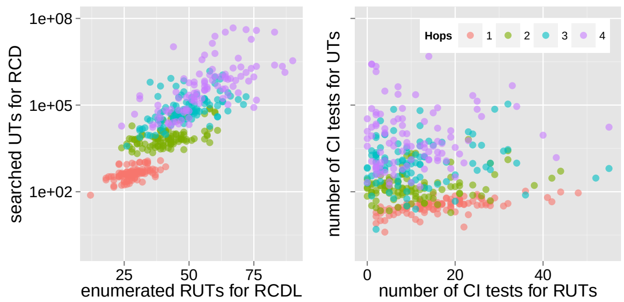


Figure 4: Empirical comparison of RCD and RCDℓ. Both y-axes are on the same logarithmic scale.

Our experiments confirm that RCDℓ is substantially more efficient than RCD with respect to its space and time requirements (see Figure 4). RCDℓ takes 70 seconds on average learning an RCM given $h = 4$ while RCD takes 50 minutes. These efficiency gains are due to the fact that RCDℓ, unlike RCD, is able to avoid redundant CI tests and has no need to construct or manipulate AGGs or an unrolled graph. Because the number of searched unshielded triples (UT) grows with the size of AGGs, RCD refines them to test CI on a small number of UTs close to the number of enumerated representative UTs. However, the number of CI tests on the selected UTs grows exponentially with h. This may be due to the fact that a separating set is sought from the neighbors of both flanking elements of each UT.

Summary and Discussion

We have presented a new theoretical analysis that (i) shows that RCD, the only other existing algorithm for learning an RCM, is *not* complete; and (ii) suggests the design of RCD-Light (RCDℓ), a sound and efficient algorithm for learning an RCM from relational data. Unlike RCD, RCDℓ requires only polynomial time and space to orient the dependencies for a sparse RCM. Our result also suggests CI tests that can be used to define an *RCM pattern* where two RCMs of the same pattern share the same set of independence relations. RCM pattern can be seen as the relational counterpart of a pattern originally defined for causal Bayesian networks by Verma and Pearl (1991).

Our analysis (as in the case of all constraint-based structure learning algorithms) assumes that the algorithm has access to a CI oracle. In practice, the reliability of tests depends on the accuracy of the parametric form assumed for the underlying distribution, and the quantity of available data. Work in progress aims to design of a complete algorithm and extend the algorithm to learn (i) temporal RCMs (Marazopoulou, Maier, and Jensen 2015) (ii) variants of RCMs that allow dependencies between the same attributes (Friedman et al. 1999) (iii) accurate models in real-world settings where CI tests are necessarily imperfect, e.g., by developing the relational counterparts of methods developed in the propositional setting (Colombo and Maathuis 2014).

Acknowledgments

The authors are grateful to AAAI 2016 anonymous reviewers for their thorough reviews, and members of the Pennsylvania State University Artificial Intelligence Research Laboratory for their useful feedback. This research was supported in part by the National Science Foundation Grant CCF 1518732, 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.

Appendix

Function extend of two relational paths Q and R is defined as $\{Q^{1:|Q|-i}+R^{i:}|i\in\mathsf{pivots}(\tilde{Q},R)\}\cap\mathbf{P}_{\mathcal{S}}$ where $Q^{m:n}$ represents a subpath of Q (inclusive), $\tilde{Q}$ is the reverse of Q, $\mathsf{pivots}(S,T)=\{i|S^{1:i}=T^{1:i}\}$, $\mathbf{P}_{\mathcal{S}}$ is a set of all valid relational paths, and '$+$' is a concatenation operator. We will use a *join* operator '$\bowtie$' for extend and denote $Q^{1:|Q|-i}+R^{i:}$ by $Q\bowtie_i R$ for a pivot i. We list several important properties of extend (i.e., $\bowtie$ operator). Let P, Q, $R\in\mathbf{P}_{\mathcal{S}}$ and $P_{|P|}=Q_1$. Then, following hold: (nonemptiness) $P\bowtie Q\neq\emptyset$; (maximal pivots) $\mathsf{pivots}(\tilde{P},Q)\subseteq\mathsf{pivots}(\tilde{Q},Q)$; (triangle symmetry) if $R\in P\bowtie Q$, then $P\in R\bowtie\tilde{Q}$; (singleton) if $|\tilde{Q}\bowtie Q|=1$, then $|P\bowtie Q|=1$; (canonicality) if $[B]\in P\bowtie Q$, then $Q=\tilde{P}$. The properties can be inferred solely from the definition of extend, and we omit proof for the properties.

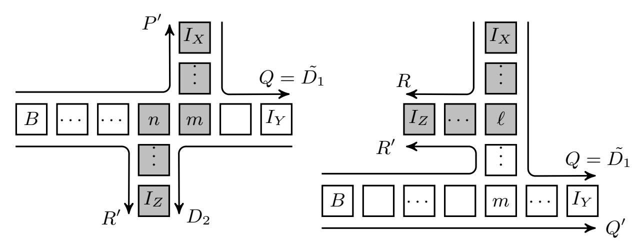


Figure 5: Schematic examples for $m\neq n$ (left) and $m=n$ (right). Simple walks from B to I_X, I_Y, and I_Z are P', Q', and R'; from I_X to I_Y and I_Z are Q and R; from I_Y to I_Z is D_2, respectively. Gray color highlights R from I_X to I_Z.

Lemma (Representative Unshielded Triple)**.** *Let* $\langle P'.X, Q'.Y, R'.Z\rangle$ *be an unshielded triple in* $\mathcal{G}_{\hat{\mathcal{M}}}$ *where* X *can be* Z*. Then, there exists a representative unshielded triple* $\langle[I_X].X, Q.Y, R.Z\rangle$ *in* $\mathcal{G}_{\hat{\mathcal{M}}}$.

Proof. (If $X=Z$) For $P'.X-Q'.Y$ and $Q'.Y-R'.X$, there must exist dependencies $D_1.X\rightarrow[I_Y].Y$ and $D_2.X\rightarrow[I_Y].Y$, respectively, in $\hat{\mathbf{D}}$. Then, $Q=\tilde{D}_1$ and we need to select R from $Q\bowtie D_2$ such that $R\neq[I_X]$ to satisfy the definition of an unshielded triple. Since $[I_X]$ is canonical, $[I_X]\neq R$ implies $[I_X].X$ and $R.X$ are not intersectable because of the bridge burning semantics. Then, we have to prove $\{[I_X]\}\neq Q\bowtie D_2=\tilde{D}_1\bowtie D_2$. Suppose for the sake

of contradiction that $R = [I_X]$ be the only path of $Q \bowtie D_2$. This implies that $D_1 = D_2$ by *canonicality* of extend. Due to the singleton property of extend, $\{P'\} = \{R'\} = Q' \bowtie D_2$, which contradicts that $\langle P'.X,\, Q'.Y,\, R'.X\rangle$ be an unshielded triple. Hence, $\{[I_X]\} \subsetneq Q \bowtie D_2$.
(Otherwise, $X \neq Z$) Similarly, there exist dependencies $D_1.X \to [I_Y].Y$ and $D_2.Z \to [I_Y].Y$ in $\hat{\mathbf{D}}$. We set $Q = \tilde{D_1}$ and choose R from $Q \bowtie D_2$ such that there is no edge from $R.Z$ to $[I_X].X$. Let m and n be pivots for P' and R' relative to Q' so that $P' = Q' \bowtie_m D_1$ and $R' = Q' \bowtie_n D_2$. If $m \neq n$, then, let $R = Q \bowtie_{\min(m,n)} D_2$. Otherwise, any R in $Q^{1:|Q|-m+1} \bowtie D_2^{m:}$ will satisfy our purpose. Let ℓ be the selected pivot so that $R = Q^{1:|Q|-m+1} \bowtie_\ell D_2^{m:}$. We can see that $R' \in P' \bowtie R$ (see Figure 5). If $m \neq n$, then pivot is $|Q| - m + |m-n| + 1$ and $|Q| - m - \ell + 2$, otherwise. Suppose $R.Z$ and $[I_X].X$ are adjacent. Since $R' \in P' \bowtie R$, there must be an edge $R'.Z \to P'.X$, which contradicts that $\langle P'.X,\, Q'.Y,\, R'.Z\rangle$ is an unshielded triple. □

Example 1. Let $\mathcal{S}$ be a schema with $\mathbf{E} = \{E_i\}_{i=1}^4$ and $\mathbf{R} = \{R_i\}_{i=1}^4$ where $R_1 : \{E_1, E_2\}$, $R_2 : \{E_2, E_3\}$, $R_3 : \{E_2, E_3\}$, and $R_4 : \{E_2, E_4\}$ with every cardinality is 'one'. Let R_1, R_2, and E_2 have an attribute class X, Y, and Z, respectively. Let $\mathbf{D} = \{D_1, D_2, D_3\}$ where $D_1 : [R_2, E_2, R_1].X \to [I_Y].Y$, $D_2 : [R_2, E_3, R_3, E_2].Z \to [I_Y].Y$, and $D_3 : [R_1, E_2, R_2, E_3, R_3, E_2].Z \to [I_X].X$. In AGGs, any relational variables $P.X$ and $R.Z$ that are adjacent to some $Q.Y$ are connected. That is, there is no unshielded triple of the form $\langle P.X, Q.Y, R.Z\rangle$ in AGGs. However, there exists a skeleton σ such that $e_1 - r_1 - e_2 - r_2 - e_3 - r_3 - e_2 \in \sigma$ where two e_2 are the same item. Then, $r_1.X \to r_2.Y$, $e_2.Z \to r_2.Y$, and $r_1.X$ and $e_2.Z$ are disconnected in its ground graph. Hence, $\langle r_1.X, r_2.Y, e_2.Z\rangle$ is an unshielded collider where a separating set between $[I_X].X$ and $[R_1, E_2].Z$ must not include $[R_1, E_2, R_2].Y$. Such independence test leads to the orientation of D_1 and D_2. RCD does not check such case and will leave all dependencies undirected (RCD cannot orient D_1, D_2, and D_3 via RBO).

References

Buntine, W. L. 1994. Operations for learning with graphical models. *The 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 (TODS)* 1(1):9–36.

Chickering, D. M. 1995. A transformational characterization of equivalent Bayesian network structures. In *UAI '95 Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, Montreal, QU*, 87–98. San Francisco, CA: Morgan Kaufmann.

Colombo, D., and Maathuis, M. H. 2014. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research* 15(1):3741–3782.

Dor, D., and Tarsi, M. 1992. A simple algorithm to construct a consistent extension of a partially oriented graph. *Technicial 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*, 1300–1309. San Francisco, CA: Morgan Kaufmann.

Getoor, L., and Taskar, B. 2007. *Introduction to statistical relational learning*. MIT press.

Heckerman, D.; Meek, C.; and Koller, D. 2007. Probabilistic entity-relationship models, prms, and plate models. In Getoor, L., and Taskar, B., eds., *Introduction to statistical relational learning*. MIT Press. 201–238.

Lee, S., and Honavar, V. 2015. Lifted representation of relational causal models revisited: Implications for reasoning and structure learning. In *Proceedings of Advances in Causal Inference Workshop co-located with 31st Conference on Uncertainty in Artificial Intelligence (UAI 2015)*, 56–65.

Maier, M.; Taylor, B.; Oktay, H.; and Jensen, D. 2010. Learning causal models of relational domains. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, Atlanta, GA*, 531–538. Menlo Park, California: AAAI Press.

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

Maier, M.; Marazopoulou, K.; and Jensen, D. 2013. Reasoning about independence in probabilistic models of relational data. *Approaches to Causal Structure Learning Workshop, UAI 2013*.

Maier, M. 2014. *Causal Discovery for Relational Domains: Representation, Reasoning, and Learning*. Ph.D. Dissertation, 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*, 572–581.

Meek, C. 1995. Causal inference and causal explanation with background knowledge. In *UAI '95 Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, Montreal, QU*, 403–410. San Francisco, CA: Morgan Kaufmann.

Pearl, J. 2000. *Causality: models, reasoning and inference*. Cambridge University Press.

Ramsey, J.; Zhang, J.; and Spirtes, P. 2006. Adjacency-faithfulness and conservative causal inference. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence (UAI-06), Cambridge, MA*, 401–408. Arlington, VA: AUAI Press.

Richardson, M., and Domingos, P. M. 2006. Markov logic networks. *Machine Learning* 62(1-2):107–136.

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

Spirtes, P.; Glymour, C. N.; and Scheines, R. 2000. *Causation, prediction, and search*. MIT press.

Verma, T., and Pearl, J. 1991. Equivalence and synthesis of causal models. In *UAI '90 Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence, Cambridge, MA*, 255–270. Amsterdam, NL: Elsevier Science.