The Pennsylvania State University
The Graduate School
College of Information Sciences and Technology

CAUSAL DISCOVERY FROM RELATIONAL DATA: THEORY AND PRACTICE

A Dissertation in
Information Sciences and Technology
by
Sanghack Lee

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2018

The dissertation of Sanghack Lee was reviewed and approved* by the following:

Vasant G. Honavar
Professor of Information Sciences and Technology
Dissertation Advisor, Chair of Committee

C. Lee Giles
David Reese Professor of Information Sciences and Technology

John Yen
Professor of Information Sciences and Technology

Bharath K. Sriperumbudur
Assistant Professor of Statistics

Andrea H. Tapia
Associate Professor of Information Sciences and Technology
Director of Graduate Programs, Information Sciences and Technology

*Signatures are on file in the Graduate School.

Abstract

Discovery of causal relationships from observational and experimental data is a central problem with applications across multiple areas of scientific endeavor. There has been considerable progress over the past decades on algorithms for eliciting causal relationships through a set of conditional independence queries from data. Much of this work assumes that the data instances are independent and identically distributed (iid). However, in many real-world applications, because the underlying data exhibits a relational structure of the sort that is modeled in practice by an entity-relationship model, the iid assumption is violated. Motivated by the limitations of traditional approaches to learning causal relationships from relational data, a relational causal model is recently introduced. The key idea behind the relational causal model is that a cause and its effects are in a direct or indirect relationship that is reflected in the relational data. Traditional approaches for reasoning with and learning causal models from iid data cannot be trivially applied in the relational setting. Against this background, this dissertation investigates a set of closely related research problems having to do with causal inference with relational data: (i) characterizing the conditional independence relations that hold in a given relational causal model, (ii) sound and complete learning of the structure of a relational causal model using an independence oracle, (iii) measuring the strength of conditional dependence and testing conditional independence among relational variables from relational data, and (iv) robustly learning the structure of a relational causal model from relational data.

Table of Contents

List of Figures

List of Tables

Acknowledgments

This dissertation would not have been possible without the support from many individuals I met through my unique, unusually long journey with *local minimas*.

First, I would like to express my deep gratitude and appreciation to my adviser, Dr. Vasant Honavar, for his continuous support during my graduate studies. He encouraged me to keep motivated with valuable suggestions and insights. His thoughtful questions and comments made me think more critically about my research. He let me explore subject matters even though he might not be able to see progress for a long time. He helped me advance my career. He is a great role model for me.

I would like to express my thanks to lab mates in Artificial Intelligence Research Laboratory at Pennsylvania State University and at Iowa State University for healthy, helpful discussion on various problems each of us faced. I sincerely give thanks to Marc Maier and his colleagues at the University of Massachusetts, Amherst, who first formulated relational causal model and put tremendous effort into devising non-traditional, important concepts to characterize it. Although my research so far lies in a computational aspect of artificial intelligence, faculty members and colleagues I met in College of Information Sciences and Technology shaped me to think more about humanities and societal consequences of what I am doing and what I will do through my career. I also thank many Korean friends and brothers in Ames and State College for making my life as a foreign student a lot easier.

I would not be who I am now without endless support from my family in Korea. My parents, Choongwoo and Boorim, and my sister, Hwaryoung, always believed in me. Words cannot fully express my feelings to them. Finally, I would like to thank my dear wife and son, Jungmin and Doyun. They have been always on my side with their unconditional love and endless support through my pursuit of Ph.D.

The research described in this dissertation was supported by grants from the National Science Foundation (NSF 1636795), the National Center for Advancing Translational Sciences, National Institutes of Health (UL1 TR002014), the Center for Big Data Analytics and Discovery Informatics and the College of Information

Sciences and Technology at the Pennsylvania State University, and the Edward Frymoyer Endowed Chair in Information Sciences and Technology at the Pennsylvania State University, and the Sudha Murty Distinguished Visiting Chair in Neurocomputing and Data Science at the Indian Institute of Science (both held by Dr. Vasant Honavar). The support provided by the Department of Computer Science at Iowa State University during to Sanghack Lee is also gratefully acknowledged.

Dedication

To my parents, sister, wife, and son

Chapter 1
Introduction

The growth of information generated by people and devices makes people think about whether data alone can provide substantive knowledge for decision making and scientific explorations (Bareinboim and Pearl, 2016). The last three decades have also seen advances in statistics, artificial intelligence, and machine learning helping us understanding underlying nature of such massive and complex data. Among many approaches and methods, *causal discovery*, identifying causal relationships and their directions, and *causal inference*, estimating effects of treatments, are at the heart of such understanding since they allow users to reason about the consequences of actions and policies beyond predictions — predictive approaches could not tell possible outcomes of actions, treatments, or change of policies.

The discovery of causal relationships from observational and, when available, experimental data is a central problem in artificial intelligence. There have been many approaches to causal inference or causal discovery, e.g., potential outcome framework by Rubin (1974), structural causal model by Pearl (1995), inductive causation (IC) algorithm by Pearl (2000), and PC algorithm by Spirtes et al. (2000). Such approaches often based on the *independent and identically distributed* (IID or iid) assumption where individuals and samples are independently acquired from a joint distribution. However, such assumption often does not hold in the real world, and there are many circumstances where data, regardless of whether it is generated from nature or human, exhibits non-iidness, which is dubbed as *relational data*.

Relational data is characterized by heterogeneous entities interrelated to each other through a diverse set of relationships, which might be modeled by graphs, entity-relationships (Chen, 1976), first-order logic knowledge base. Examples of relational data include genomic structures, hyperlinked pages in the WWW, epidemic disease data, and data generated through the interactions among people.

Relational data have been studied by diverse scientific communities. For example, social scientists studied phenomena in social networks and characterized networks by reciprocity, transitivity, homophily, etc. Epidemiologists studied how disease spread across the population of interest, and tried to reveal the causes of disease that can help designing better public health interventions.

Relational data requires much more attention especially in the problem of causal inference and discovery. Naively applying iid-based methods to non-iid relational data causes biases in causal effect estimates and erroneous causal structures, which might result, for example, inappropriate public policies. Hence, it is of great interest for many researchers to extend existing methods of causal inference, causal discovery, and causal models to capable of relational data. For example, Tchetgen Tchetgen and VanderWeele (2012) formally addressed the issue of *interference* (Cox, 1958) in observed (social) network data.

In regards to models, causal models have been developed almost exclusively for iid cases (Spirtes et al., 2000). 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. However, such models lack the expressive power to model causal relationships in relational domains. Only recently, Maier and his colleagues (2010; 2013a) proposed new representation called Relational Causal Model (RCM), which enables the representation of "relational causal dependencies", reasoning about "relational conditional independence", and learning the "relational causal structure of data". However, we found that there are more questions unanswered than answered. We first review how researchers conduct causal inference or model causality of the domain of interests from observational data.

1.1 Causal Inference

The primary task in causal inference is predicting the outcome of an intervention. Randomized controlled trial is a gold standard (Holland, 1986) to conduct an experiment to identify the effect of interventions by comparing the outcomes of experimental (i.e., treatment) and control groups. Random allocation of units to each group ensures that treatment will not be confounded with both measured and unmeasured covariates. However, there are numerous circumstances where such

randomized study is impossible for financial, ethical, or technical reasons. Therefore, there are reasons to develop methods for inferring causal effects from observational data.

Potential outcome framework (also known as Rubin causal model or counterfactual framework) (Rubin, 1974; Holland, 1986) has its root in the ideas of Cox (1958) and Fisher (1958) and mathematical expression by Neyman et al. (1990). Rubin popularized the framework by connecting the definition of a causal effect as in an experiment to observational studies. In this framework, an individual causal effect is defined as the difference between the *potential* outcomes of an individual (i.e., unit) under two different exposure conditions, assuming binary exposure without loss of generality. Formally, there is an exposure or treatment variable $w_i \in \{0, 1\}$ for each unit i (e.g., a person in population). There are two potential outcomes $Y_i(0)$ and $Y_i(1)$. It is potential because we are, in general, able to observe only one output. We may want to measure *average treatment effect*, $\text{ATE} := \mathbb{E}[Y(1) - Y(0)]$, *average treatment effect among treated*, $\text{ATT} := \mathbb{E}[Y(1) - Y(0) \mid W = 1]$, or focus on *intent-to-treat analysis* if there are units non-compliant to treatments they received.

Instead of explicitly representing relationships among variables as in causal models, computation of causal estimands is based on assumptions for conditionality, e.g., whether one can represent probability in a succinct way by taking out some conditionals. Some of well-known tools and methods for potential outcome framework are *propensity score matching* or *inverse probability weighting* (Rosenbaum and Rubin, 1983) for alleviating differences in tendencies of units for being received the treatment, i.e., $\text{P}(w \mid X)$, *principal stratification* (Frangakis and Rubin, 2002) for adjusting results for post-treatment covariates, and *instrumental variables* (Angrist et al., 1996) for non-compliance in treatment.

Most of causal inference literature assumes there is no *interference* under the name of Stable Unit Treatment Value Assumption (SUTVA). In a network setting (especially social ones), there do exist *interference* (often termed with peer-effect or spillover) where a treatment received by an individual might affect other individuals (under treatment or control) through social interactions. Hence, a naive way to applying existing methods of causal inference results biases in estimates. Very recently, researchers develop principled ways for causal inference in the presence of interference. Some researchers assume that there are multiple independent groups. This type of study *intentionally* limits interactions between groups while

allowing within groups (Sobel, 2006; Rosenbaum, 2007; Hudgens and Halloran, 2008; Tchetgen Tchetgen and VanderWeele, 2012). For example, Tchetgen Tchetgen and VanderWeele (2012) introduced an inverse probability weighting approach to causal inference under interference in observational study. Some recent studies are free from such assumption (Aronow and Samii, 2013; Toulis and Kao, 2013; Ugander et al., 2013). Rather than pursuing unbiased estimator for causal effect under implausible assumptions, Toulis and Kao (2013) and Ugander et al. (2013) recognized that estimators might be biased. Causal inference in the presence of interference is relatively new, and there remains considerable work to be done.

1.2 Causal Models

Causality among variables of interests can be often represented with *directed acyclic graph* (DAG) where nodes correspond to variables and a directed edge represents a *direct* cause-effect relationship between a pair of variables. Such representation is at least about one century old. Wright (1921) illustrated diagrams to represent direct influence among variables, and Haavelmo (1943) introduced *structural equation models* (SEM) in Economics. In Statistics, the conditional independence structure of a set of random variables is studied (Lauritzen and Wermuth, 1989). Among possible structures, much attention was paid to undirected graphs (Markov random field, MRF) and directed acyclic graphs (also called Bayesian network) although Studeny (2005) showed that the notion of conditional independence is not exclusively represented by such graphs.

Bayesian networks (Pearl, 1988; Lauritzen and Wermuth, 1989; Spirtes et al., 2000; Neapolitan, 2003; Koller and Friedman, 2009) provide a compact representation of relationships among random variables. Formally, given a set of variables $\mathbf{X} := \{X_i\}_{i=1}^n$, a Bayesian network is a DAG $\mathcal{G}$ where nodes are variables $\mathbf{X}$ and a directed edge from X_i to X_j represents X_i is the direct cause of X_j. For each variable X_i, a conditional probability distribution is associated to describe $\mathrm{P}\left(X_i \mid pa\left(X_i; \mathcal{G}\right)\right)$ where $pa\left(X_i; \mathcal{G}\right)$ denotes parents (direct causes) of X_i in the DAG $\mathcal{G}$.

When it is equipped with a few assumptions that let us interpret given Bayesian network *causal*, we explicitly call such a mathematical object as a *causal Bayesian network* or just a causal network. In brief, *causal sufficiency condition* says common causes of measured variables are all measured, *causal Markov condition* says a

variable is independent of non-descendant variables conditional on all of its parents (direct causes). In addition, *faithfulness* asserts that there is no independence relations other than those entailed by d-separation (Pearl, 1988): X and Y are conditionally independent given Z, denoted by $X \perp\!\!\!\perp Y \mid Z$, if and only if there exists no unblocked path from X to Y in the underlying graph conditional on Z.

Causal models allow users to compute causal effects. Given a fully-specified model structure and parameters (i.e., a conditional probability distribution $\mathrm{P}\left(V \mid pa\left(V; \mathcal{G}\right)\right)$ for every random variable V in the model), $\mathrm{P}\left(Y \mid do\left(X = x\right)\right)$, interventional distribution of Y given X fixed to x, is inferred by manipulating its causal graph $\mathcal{G}$ by removing influence from parents of X (by simulating value of X held to x) on X. Formally,

$$\mathrm{P}\left(Y \mid do\left(X = x\right)\right) := \sum_{\mathbf{V}\setminus\{Y,X\}} \prod_{V \in \mathbf{V}\setminus\{X\}} \mathrm{P}\left(V \mid pa\left(V; \mathcal{G}\right)\right)$$

where a constant 1 replaces $\mathrm{P}\left(X = x \mid pa\left(X; \mathcal{G}\right)\right)$ since $X = x$ is *true* irrelevant to what values of causes of X are. Note that the distribution $\mathrm{P}\left(Y \mid do\left(X = x\right)\right)$ in $\mathcal{G}$ is equivalent to $\mathrm{P}\left(Y \mid X = x\right)$ in a mutilated graph $\mathcal{G}_{\bar{X}}$, the causal model by removing edges between X and its parents.

The structure of (causal) Bayesian networks can be learned, up to its Markov equivalence (without assuming functional form of random variables), from given observational data, and experimental data if available. For example, Bayesian networks have been used to interpret and discover gene regulatory pathways (Friedman et al., 2000). There are many approaches to learning the structure and parameters of a Bayesian network which can be mainly categorized into two: constraint-based and score-based.

Constraint-based (or non-Bayesian method) algorithm is based on a conditional independence structure characterized by an original DAG structure (Pearl, 2000; Spirtes et al., 2000; Margaritis and Thrun, 2000; Pellet and Elisseeff, 2008). This approach carries out conditional independence tests on a given data and constructs a Bayesian network that conforms to the test results. Polynomial time algorithms exist under the assumption of limited number of parents (the maximum number of causes) (Claassen et al., 2013).

Alternatively, a score-based searching method defines a measure, which evaluates a given Bayesian network model (both structure and parameters), as a function of

	Non-relational	**Relational**
Associational	(acausal) Bayesian network, Markov network	Plate, probabilistic relational model (PRM), directed acyclic probabilistic entity relationship (DAPER), Markov logic network (MLN),
Causal	causal Bayesian network (CBN), semi-Markov causal models or structural causal models, ancestral graph	**relational causal model (RCM)**

Table 1.1: A partial list of models based on four categories

the given data. Then, algorithms search for the (global or local) optimal Bayesian network in the space of structures and its parameters. This approach often works well with small data compared to constraint-based methods. However, it is well-known that learning optimal structure of Bayesian network is NP-hard (Chickering et al., 2004). Some of notable scores and algorithms are based on defining posterior of structure given data and prior knowledge (Cooper and Herskovits, 1992; Heckerman, 1995; Heckerman et al., 1995; Chickering and Meek, 2002) and some information-theoretic scoring functions based on AIC and MDL/BIC. Other existing approaches try to combine both approaches in a principled and effective way (Tsamardinos et al., 2006; Claassen and Heskes, 2013).

Bayesian network is one of the simplest form to represent causality. The following are models that subsume (causal) Bayesian network as a special case. *Structural Causal Model* (Pearl, 1995; Tian and Pearl, 2002; Shpitser and Pearl, 2008) combines features of the structural equation models (Haavelmo, 1943), graphical models for probabilistic reasoning, and causal analysis. Ancestral graph (Spirtes et al., 2000; Richardson and Spirtes, 2002; Zhang, 2008) is capable of representing independence relations even when there are hidden variables (latent confounding variables or selection biases). *Maximal Ancestral Graph* (MAG) generalizes the class of DAGs that is closed under marginalization and selection.

1.3 Statistical Relational Models and Relational Causal Model

Relational data can be represented as, but not exclusively, following three forms — graphs, relational databases, and first-order logic knowledge base. Statistical relational models focused on modeling the joint distribution of attributes of entities and relationships in a given data. These models include Plate models (Buntine, 1994), Probabilistic Relational Model (PRM) (Friedman et al., 1999), Directed Acyclic Probabilistic Entity-Relationship (DAPER) (Heckerman et al., 2007), Relational Dependency Network (RDN) (Neville and Jensen, 2007), etc. They extend traditional Bayesian networks to relational settings by relaxing the assumption about iid instances. These models have been successfully applied in several relational domains, e.g., genomic data (Friedman et al., 2000), scientific literature, and the WWW. Alternatively, other group of researchers focused on combining first-order logic with probabilities (e.g., Markov logic network by Richardson and Domingos, 2006). Further, there are "statistical network models" in the field of (social) network analysis, where researchers interested in the formation of a single network (cross-sectionally observed) or dynamics thereof.

In this dissertation, we are interested in representing causality among attributes of objects comprising relational data (see Table 1.1). For the purpose of representing causality in relational data, we will focus on generalization of Bayesian network to relational settings. Maier et al. (2010) showed that the Directed Acyclic Probabilistic Entity-Relationship model (DAPER) (Heckerman et al., 2007) which generalizes both Probabilistic Relational Model (PRM) (Friedman et al., 1999) and plate model (Buntine, 1994) is sufficient to represent causality in relational domains (in the sense that DAPER yields a ground graph which can be viewed as a Bayesian network over attributes of items.) However, these models and learning algorithms are not intended for learning causal models where learning algorithm focused on finding a structure better *fitting* a given data. Hence, Maier et al. (2010, 2013a) developed a relational causal model which is rooted in the intersection of PRM and DAPER with causal assumptions.

In brief, a *relational schema* is used to formally describe the class of entities and relationships in the domain of interest (e.g., companies). A *relational skeleton*

	Theory	Practice
Conditional Independence	Can we correctly, efficiently infer a relational version of conditional independence from a given RCM?	Can we efficiently, accurately measure and test relational version of conditional (in)dependence from a relational data?
Structure Learning	Can we correctly, completely learn the structure of an underlying RCM given access to an independence oracle?	Can we efficiently, accurately learn the structure (and parameters) of an underlying RCM from a relational data?

Table 1.2: Four questions themed around relational causal model

is an instance of relational schema. Relational causal model defined on a relational schema is composed of a set of causal dependencies between attributes of possibly interrelated entities and relationships. For example, 'the success of a product depends on the competences of employees who develop the product'. These relational dependencies are *translated* into a given relational skeleton and produces a ground graph, which is a Bayesian network on attributes of items.

1.4 Four Fundamental Research Questions on Relational Causal Model

In this dissertation, we first address theoretical characteristics of relational causal models with particular emphasis on what conditional independence statements a model implies and how we can reconstruct the model structure from an observational relational data given an independence oracle. Then, we address issues arose in practice, e.g., testing conditional independence and learning the structure of the model when the given relational data is not in an idealistic condition where test results are not fully reliable. We outline in Table 1.2 the two tasks of relational causal model from the aspects of both theory and practice. We summarize each topic below.

1.4.1 Relational d-Separation: Relational Conditional Independence of Relational Causal Model

Conditional independence is an important notion in many machine learning algorithms and statistical models dealing with multiple variables. Furthermore, causal discovery heavily relies on conditional independence statements obtained from a given observational data. Formally, a joint probability distribution P over variables X, Y, and Z satisfies a *conditional independence* relationship $X \perp\!\!\!\perp Y \mid Z$, which reads as X is conditionally independent of Y given Z, if the joint distribution is decomposed as $\mathrm{P}(X, Y, Z) = \mathrm{P}(X \mid Z)\,\mathrm{P}(Y \mid Z)\,\mathrm{P}(Z)$ assuming the existence of conditional distributions $\mathrm{P}(X \mid Z)$ and $\mathrm{P}(Y \mid Z)$ with $\mathrm{P}(Z = z)$ being positive.

In RCM, we are interested in answering conditional independence among relational variables, which generalizes the conditional independence among non-relational variables. For example, "is success of a product independent from its developers' salaries given their competencies?" However, in general, relational data is non-independently and non-identically distributed. Hence, the notion of CI needs to be generalized appropriately to address issues in relational settings. Against this background, Maier et al. (2013a) defined relational CI as "a collection of individuals' (i.e., item's) CI statements for every possible instantiation (e.g., company) of a given relational schema and for every base item (e.g., every employee)". A relational CI statement is true if and only if d-separation (a graphical criterion to check conditional independence under Markov condition and faithfulness, (Geiger et al., 1989)) holds on every base item in every possible ground graph, which we call *relational d-separation*. It is non-trivial to determine whether relational d-separation holds for two reasons:

- *All-ground-graphs semantics* that relational d-separation requires d-separation to hold over all instantiations of the model;
- *Intersectability* of relational variables that their terminal sets may overlap.

Maier et al. (2013a) devised a lifted representation of all ground graphs, Abstract Ground Graph (AGG), and a mechanism that answers a relational CI query against an RCM by reducing the query to a traditional CI query against the corresponding AGG. However, we demonstrated that both their representation and mechanism are problematic (Lee and Honavar, 2015).

1.4.2 Learning Structure of Relational Causal Model with Independence Oracle

Causal discovery refers reconstruction of causality among variables from observational and interventional data, if available. In RCM, we are interested in learning the structure of an RCM (whether a relational dependency exists and its direction), a causal process, which might have generated a given data. In a theoretical aspect, we focus on how we can completely and efficiently (i.e., avoiding unnecessary CI queries) learn the structure of an RCM. By *complete* we mean that there is no other learning algorithm that can outperform in identifying the existence of and orientation of relational dependencies without any further assumptions. To examine the completeness of an algorithm, one must characterize the *Markov equivalence* of an RCM — whether two RCMs yield the same CI relations — since no algorithm will be able to discriminate two Markov equivalent RCMs without further assumptions. Existing algorithms (Maier et al., 2013a; Marazopoulou et al., 2015; Lee and Honavar, 2016b) before our paper (Lee and Honavar, 2016a) are known to be not complete (Lee and Honavar, 2016b).

1.4.3 Testing Relational Conditional Independence from Relational Data

Unlike *reasoning* about CI relations from a known RCM, another important task is *measuring* the strength of and *testing* conditional (in)dependence of relational variables from relational data that might have been generated from an RCM. In the case of non-relational settings, there are many existing approaches from Pearson product-moment correlation coefficient, Spearman's rank correlation coefficient, and partial correlation to nonparametric kernel-based conditional independence test (Fukumizu et al., 2008; Zhang et al., 2011).

In relational settings, we have a relational data from a single relational skeleton, where we focus on testing relational skeleton-specific relational CI. Relational data can be viewed as a single sample generated from a ground graph, that is, the snapshot of a given relational skeleton. In a relational data, instances are non-independently and non-identically distributed. Existing approaches (Maier et al., 2013a; Marazopoulou et al., 2015; Arbour et al., 2016b) flattened a given relational

data and applied aggregation (e.g., average), which ignores the nature of relational data.

1.4.4 Robustly Learning Relational Causal Model from Relational Data

In the real world, assumptions are often violated — samples are finite and data is noisy. This yields faulty relational CI tests which make learning algorithms designed under the perfect test assumption perform poorly given a real-world or simulated dataset. Previous existing work (Maier et al., 2013a; Maier, 2014) simply adopted model averaging approach — averaged RCMs learned from the same data with different variable orderings (since the performance of *their* algorithm is affected by such ordering). More recent work (Arbour et al., 2016b) studied conditions under which we can orient the direction of a single relational dependency with very strong assumptions.

1.5 Dissertation Structure

The rest of this dissertation is structured as follows. We define relational causal model and relevant concepts in Chapter 2 including our new semantics, path semantics. Next four chapters correspond to the four research questions on RCM. We focus on answering relational conditional independence given a relational causal model (Chapter 3). In Chapter 4, we characterize the Markov equivalence class of a relational causal model, and devise a sound and complete structure learning algorithm for relational causal model given an independence oracle. We move to practical questions where questions and answers are specific to a given relational data. In Chapter 5, we study how naively adopting iid-based conditional independence test can be wrong and devise a kernel-based conditional independence test for relational data. We then build a practical, robust structure learning algorithm for relational causal model from relational data (Chapter 6). We finally recapitulate our contributions and provide conclusions with future research directions (Chapter 7).

Chapter 2
Causal Model for Relational Domain

This chapter describes the definition of Relational Causal Model (RCM) and its relevant concepts initially developed by Maier and his colleagues (Maier et al., 2010, 2013b,a; Maier, 2014). We also introduce new semantics called path semantics for RCM.

2.1 Relational Domain

A Relational Causal Model (RCM) tries to model causality arising in a data where entities consisting of such data interact with each other. A relational domain comprises of entities that are interdependent through relationships. A specification of such relational domain is called *relational schema*.

2.1.1 Relational Schema

Among numerous ways by which relational data can be represented, we develop RCMs on the top of entity-relationship modeling, which is popular in database community. Following definition is adopted from (Maier, 2014).

Definition 1 (Relational Schema)**.** A relational schema

$$\mathcal{S} := \langle \mathbf{E}, \mathbf{R}, \mathbf{A}, \mathsf{card} \rangle$$

consists of a set of entity classes $\mathbf{E}$; a set of relationship classes $\mathbf{R}$; a set of attribute classes $\mathbf{A}$; and cardinality constraints card on a relationship class R and its participating entity class E: $\mathsf{card} : \mathbf{R} \times \mathbf{E} \to \{\mathsf{one}, \mathsf{many}\}$.

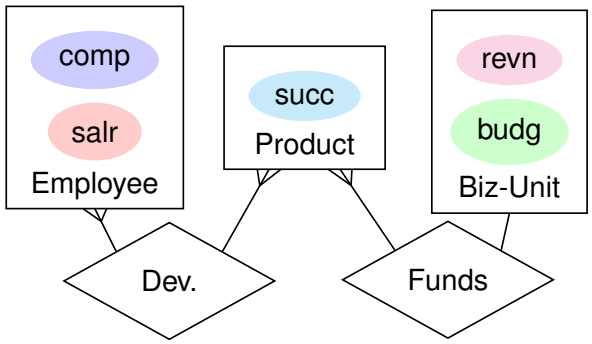


Figure 2.1: An example of a relational schema (Maier, 2014) for companies. There are three entity classes (Employee, Product, Business-unit) and two relationship classes (Develops and Funds) with attribute classes (Competence, Salary, Success, Revenue, and Budget) where every product is funded by at most only one business-unit.

We reproduce a relational schema for companies used in (Maier, 2014) in Figure 2.1. A relationship class $R \in \mathbf{R}$ consists of a set of participating entity classes. There can be more than two participating entity classes meaning that a relationship class can be binary, ternary, or more. We assume that participating entity classes are unique so that we can avoid labeling participating entity classes with 'roles'. Further, self-relationship is not allowed. We denote by $E \in R$ that an entity class E participates in a relationship class R. For example, a relationship class Advises can be represented as $\{\mathsf{Professor} : \mathsf{many},\ \mathsf{Students} : \mathsf{many}\}$ if a student can be advised by multiple professors and a professor can advise multiple students in a university domain.

Entity and relationship classes are also called item classes denoted by $\mathbf{I}$. Every item class can have multiple attribute classes. $\mathbf{A}(I) \subseteq \mathbf{A}$ is a set of attribute classes of an item class $I \in \mathbf{I}$. For example, $\{\mathsf{Age}, \mathsf{Rank}\} \subseteq \mathbf{A}(\mathsf{Professor})$. We say two item classes are *adjacent* to each other interpreting the relational schema as an undirected bipartite graph of entity and relationship classes where an edge represents the participation of an entity class to a relationship class. For example, a relationship class Advises is adjacent to both Professor and Student.

Without loss of generality, we assume that sets of attribute classes of different item classes are mutually exclusive, that is, $\mathbf{A}(I) \cap \mathbf{A}(J) = \emptyset$ for every item class $I \neq J \in \mathbf{I}$. Hence, an item class, which associates with an attribute class, is uniquely identified. We denote I_X an item class $\mathbf{A}^{-1}(X)$.

2.1.2 Relational Skeleton

A relational schema provides how entities corresponding to entity classes will interact each other through relationships specified by relationship classes and cardinality constraints. A relational skeleton (skeleton for short) is a particular realization of the given relational schema. We denote a relational skeleton by σ, a member of all possible relational skeletons $\Sigma_{\mathcal{S}}$.

Definition 2 (Relational Skeleton (adopted from (Maier, 2014)))**.** A relational skeleton σ for relational schema $\mathcal{S} = \langle \mathbf{E}, \mathbf{R}, \mathbf{A}, \mathsf{card} \rangle$ specifies a set of entities $\sigma(E)$ for each $E \in \mathbf{E}$, relationships $\sigma(R)$ for each $R \in \mathbf{R}$, and a set of participation of entities to relationships, $e \in r$ only if $E \in R$ where $e \in \sigma(E)$ and $r \in \sigma(R)$ for some $E \in \mathbf{E}$ and $R \in \mathbf{R}$. Relational skeleton adheres the cardinality constraints:

$$\forall_{r \in \sigma(R)} |ne(r; \sigma) \cap \sigma(E)| \leq 1$$

if $\mathsf{card}(R, E) = \mathsf{one}$ for every $R \in \mathbf{R}$, $E \in R$, and $\sigma \in \Sigma_{\mathcal{S}}$

A relational skeleton can be viewed as an undirected bipartite graph where vertices are items (i.e., instances of item classes). An edge $e - - r$ is defined between a relationship r and an entity e if the entity e participates in the relationship r. Oftentimes, we represent a relational skeleton as an undirected graph of entities when an edge represents that two of its ends (entities) participate in a relationship, which is implicit in the edge. To be clear, two entities can have multiple relationships of the same relationship class if both entity classes participate the relationship class with many cardinality.

A relational skeleton corresponds to a relational data, which consists of (i) the relational structure of items and (ii) item attributes of each item, e.g., $\{i.X \mid I \in \mathbf{I}, i \in \sigma(I), X \in \mathbf{A}(I)\}$.

2.2 Relational Causal Model

A relational causal model (RCM) (Maier et al., 2010, 2013a) consists of a set of cause-effect relationships and a set of parameters where the causes and an effect are related in the given relational schema. For example, "the success of a product

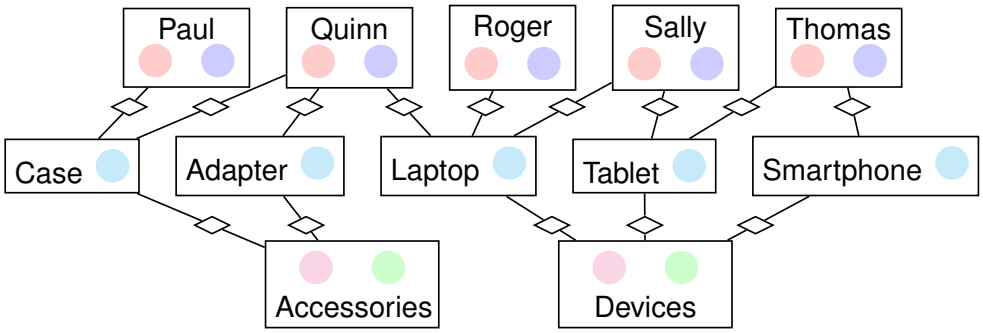


Figure 2.2: An example of a relational skeleton (Maier, 2014) corresponding to the company schema $\mathcal{S}$

depends on the competences of employees who develop the product" is encoded as a *relational dependency*,

$$[\mathsf{Product}, \mathsf{Develops}, \mathsf{Employee}].\mathsf{Competence} \rightarrow [\mathsf{Product}].\mathsf{Success}.$$

We elaborate on each component of an RCM more precisely in what follows.

2.2.1 Relational Path, Terminal Set, and Semantics

As shown in the example of a relational dependency, we want to establish dependency between Competence and Success by putting constraints on the relationship between who and what (i.e., who's competence to what's success). Among several possible ways, RCM defined over a relational schema $\mathcal{S}$ uses a (constrained) walk on the given relational schema $\mathcal{S}$. Such a walk is named Relational Path after (Maier et al., 2013a).

Definition 3 (Relational Path)**.** A relational path for relational schema $\mathcal{S}$ is an alternating sequence of entity and relationship classes $\mathbf{E} \cup \mathbf{R}$ where every consecutive item classes are adjacent in $\mathcal{S}$.

The above definition is modified from the original definition by Maier et al. (2013a) (the original definition imposes two more constraints). Their original definition is incorporated with *Bridge-Burning Semantics* which will be described later. We separate semantics from the definition of relational path, and present a different semantics, called *Path Semantics*. 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).

We call the beginning item class of a relational path as a *base item class* (originally 'base item') of the relational path. The ending item class of a relational path as a *terminal item class* of the relational path. The path explains the relation of the terminal item class from the *perspective* of the base item class. Hence, a base item class is also called a *perspective*.

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., a prefix of P) or to the end of the path (i.e., a suffix of P), i.e., $P^{:m} = P^{1:m}$ and $P^{\ell:} = P^{\ell:n}$. Further, $P^{a:b:-1} = \tilde{P}^{n-b+1:n-a+1} = \widetilde{P^{a:b}}$, backward slicing from ath element to bth element. A relational path is *canonical* if it is of unit length.

Multiple relational paths can be combined to yield new relational paths. The basic operation is '+', a concatenation operator. We will use a binary *join* operator '$\bowtie$' for extend and denote $Q^{1:|Q|-i} + R^{i:}$ by $Q \bowtie_i R$ for a pivot i where pivots of two relational paths is defined as

$$\mathsf{pivots}(S, T) := \{i | S^{:i} = T^{:i}\}.$$

Pivot is later used to describe extend devised by (Maier et al., 2013a). Capital letters $P,\ Q,\ R$ are often assigned to represent relational paths.

A relational path, restrictions imposed in a space of relational schema, can be translated into relational skeletons. Terminal set relates a base item (an item whose associating item class is the base item class of the relational path) to a set of items.

Definition 4 (Terminal Set)**.** For a relational skeleton $\sigma \in \Sigma_{\mathcal{S}}$, the terminal set, denoted by $P|_i^{\sigma}$, for relational path P and an item $i \in \sigma(I)$ where I is the base item class of P is a set of items reachable from the item i via interpreting P given the choice of semantics.

We might omit σ from $P|_i^{\sigma}$ for readability when it is not necessary to disambiguate different relational skeletons. We will describe how terminal sets are defined in Section 2.3 in detail. Under the choice of semantics, a relational path P is

valid under the chosen semantics if $\exists_{\sigma \in \Sigma_{\mathcal{S}}} \exists_{i \in \sigma(I)} P|_i^\sigma \neq \emptyset$. We will denote all valid relational paths of the given relational schema as $\mathbf{P}_{\mathcal{S}}$.

2.2.2 Relational Variable, Dependency, and Model

In a relational setting, item attributes interact with other item attributes when they satisfy certain relationship, which RCM uses relational paths and terminal sets. Maier et al. (2013a) defined *Relational Variable* as follows.

Definition 5 (Relational Variable)**.** A relational variable $P.X$ consists of a relational path P and an attribute class X where X is an attribute class of the terminal item class of P, i.e., $X \in \mathbf{A}(P^{|P|})$.

Capital letters $U,\ V,\ W$ are often assigned to represent relational variables. An (infinite) set of valid relational variables is denoted by $\mathbf{V}_{\mathcal{S}}$, which is

$$\{P.X \mid P \in \mathbf{P}_{\mathcal{S}},\ X \in \mathbf{A}(P^{|P|})\}.$$

A relational variable is canonical if its relational path is canonical. A *canonical relational variable* is denoted by V_X, which is simply $[I_X].X$. Combined with a terminal set of its relational path, a relational variable is instantiated into a set of item attributes, called relational variable instances,

$$P.X|_i^\sigma := \{j.X \mid j \in P|_i^\sigma\}.$$

For a set of relational paths $\mathbf{P}$, $\mathbf{P}.X|_i^\sigma := \{j.X \mid j \in P|_i^\sigma, P \in \mathbf{P}\}$ and for a set of relational variables $\mathbf{U}$, $\mathbf{U}|_i^\sigma := \bigcup_{U \in \mathbf{U}} U|_i^\sigma$. Now, we define a relational dependency, which is a basic building block of RCM.

Definition 6 (Relational Dependency)**.** A relational dependency $P.X \rightarrow Q.Y$ is a pair of two relational variables $P.X$ and $Q.Y$ where P and Q share the same base item class and Q is a canonical relational variable.

A capital letter D is often used as a symbol for a relational dependency. Restricting Q to be a canonical relational variable significantly reduces expressive power of RCMs. However, this lets us define models a simple and more interpretable way.[1]

[1] A primary reason will be intersectability among relational variables which will be described in detail in Chapter 3.

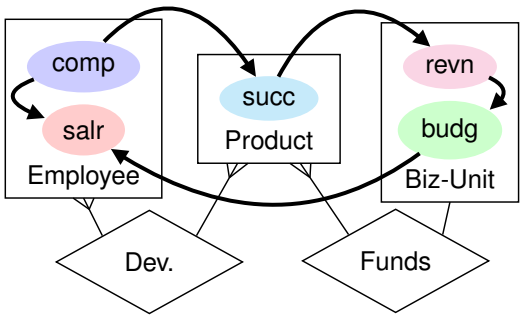


Figure 2.3: An example of an RCM $\mathcal{M}$ (Maier, 2014) defined on the company schema $\mathcal{S}$. Actual five relational dependencies are shown on the right side with initial letters of item classes since a relational dependency cannot be represented with an arrow between two attribute classes.

Two useful operations on relational dependencies are reversing and undirecting them. The reverse of a relational dependency $D = P.X \rightarrow V_Y$ is

$$\tilde{D} := \tilde{P}.Y \rightarrow V_X.$$

An undirected relational dependency $\bar{D}$ of D is defined as $\bar{D} := P.X - V_Y$ (or $\bar{D} := \tilde{P}.Y - V_X$). We treat them as identical, $P.X - V_Y = \tilde{P}.Y - V_X$. Now, we proceed to define Relational Causal Model.

Definition 7 (Relational Causal Model)**.** Given a relational schema $\mathcal{S}$, a Relational Causal Model $\mathcal{M}$ is defined with a set of relational dependencies $\mathbf{D}$ and corresponding parameters $\boldsymbol{\Theta}$,

$$\mathcal{M} := \langle \mathcal{S}, \mathbf{D}, \boldsymbol{\Theta} \rangle$$

where $\boldsymbol{\Theta}$ defines a set of conditional probability distributions

$$\{P(V_X \mid pa(V_X; \mathcal{M}))\}_{X \in \mathbf{A}}.$$

We define adjacencies, parents, and children of canonical variables of an RCM $\mathcal{M}$: $adj(V_X; \mathcal{M}) := \{P.Y \mid P.Y - V_X \in \bar{\mathbf{D}}\}$, $pa(V_X; \mathcal{M}) := \{P.Y \mid P.Y \rightarrow V_X \in \mathbf{D}\}$, and $ch(V_X; \mathcal{M}) := \{\tilde{P}.Y \mid P.X \rightarrow V_Y \in \mathbf{D}\}$. We later extend adj, pa, and ch to take a non-canonical relational variable as an argument (see Section 3.2.2).

The parameters $\boldsymbol{\Theta}$ define conditional distributions of a canonical relational variable given 'values' of parent relational variables, where each value can be a

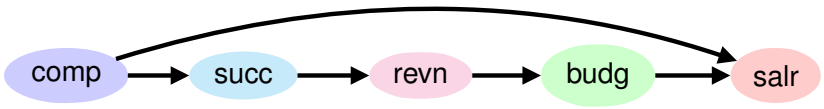


Figure 2.4: The class dependency graph $\mathcal{G}^{\mathcal{M}}_{\mathbf{A}}$ of the company RCM $\mathcal{M}$

set of values. The definition will be best understood within the context of *Ground Graph* in the next section. When parameters are not necessary, we omit $\boldsymbol{\Theta}$ in the specification of an RCM. We now project an RCM into the space of attribute classes.

Definition 8 (Class Dependency Graph)**.** Given a relational causal model $\mathcal{M} = \langle \mathcal{S}, \mathbf{D}, \Theta \rangle$, the class dependency graph $\mathcal{G}^{\mathcal{M}}_{\mathbf{A}}$ is a directed acyclic graph of attribute classes $\mathbf{A}$ where a set of edges of $\mathcal{G}^{\mathcal{M}}_{\mathbf{A}}$ is defined as

$$\{X \to Y \mid P.X \to V_Y \in \mathbf{D}\}$$

Throughout this dissertation, we assume that there exists a partial order of attribute classes with respect to the given RCM where the partial order $\pi_{\mathbf{A}}$ (or π for short) is induced by $\mathbf{D}$. We denote by $X \prec_\pi Y$ if $X \in an(Y; \mathcal{G}^{\mathcal{M}}_{\mathbf{A}})$. 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 $Q.X \to \mathcal{V}_Y$). This acyclicity assumption limits the expressive power of RCM. However, it makes us to prove the characteristics of RCM with tools developed for causal Bayesian network.

2.2.3 Ground Graph

As a relational schema is a template for relational skeletons. A relational causal model, which elaborates causal dependencies through the language of underlying relational schema, is a template for causal relationships arising in a given relational skeleton. Such an instantiation of a relational causal model given a relational skeleton is called a ground graph.

Definition 9 (Ground Graph)**.** Given the choice of semantics, RCM $\mathcal{M} = \langle \mathcal{S}, \mathbf{D}, \boldsymbol{\Theta} \rangle$, and $\mathcal{S} = \langle \mathbf{E}, \mathbf{R}, \mathbf{A}, \mathsf{card} \rangle$, a ground graph $\mathcal{G}^{\mathcal{M}}_{\sigma}$ is a directed graph where vertices are

$$\{i.X \mid X \in \mathbf{A}(I), i \in \sigma(I), I \in \mathbf{I}\}$$

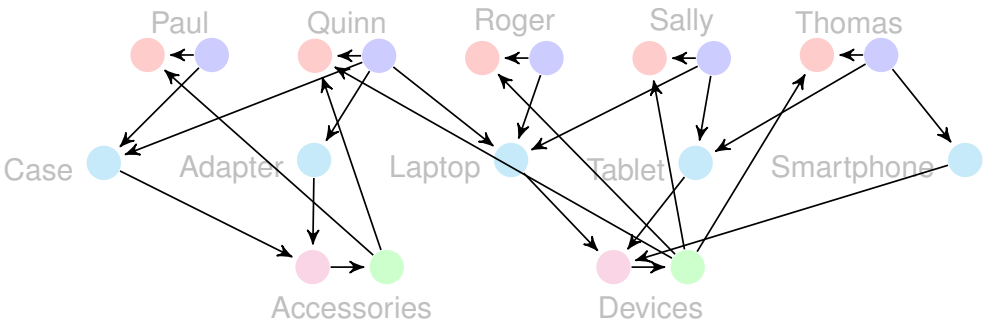


Figure 2.5: The ground graph $\mathcal{G}_\sigma^{\mathcal{M}}$ of the company RCM $\mathcal{M}$

and edges are

$$\{i.X \to j.Y \mid P.X \to V_Y \in \mathbf{D}, i \in P|_j^\sigma, j \in \sigma(J), J \in \mathbf{I}, Y \in \mathbf{A}(J)\}$$

The resulting ground graph can be viewed as a causal Bayesian network. A set of parameters $\boldsymbol{\Theta}$ is also translated into conditional probability distribution for every item attribute,

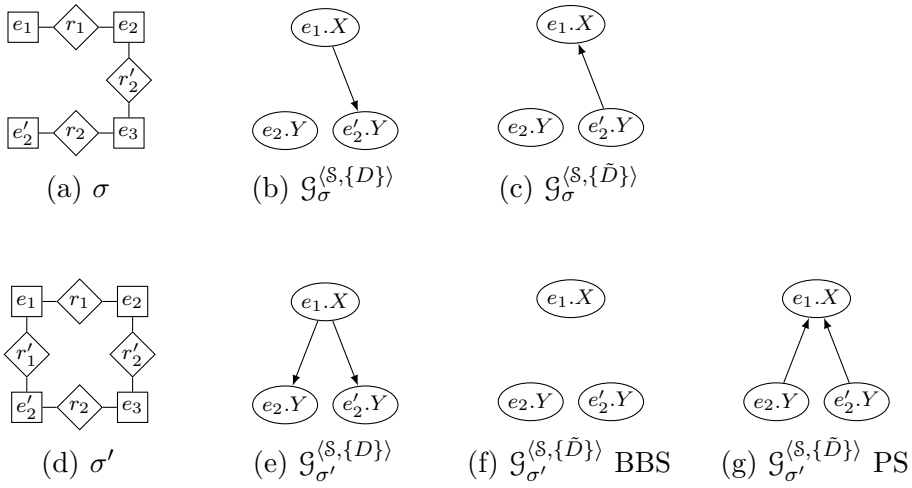
$$\{P(i.X \mid pa(i.X; \mathcal{G}_\sigma^{\mathcal{M}})) \mid i \in \sigma(I),\ X \in \mathbf{A}(I),\ I \in \mathbf{I}\}.$$

Under the assumption that the relational data represented by a relational skeleton is generated based on some ground graph, identifying the structure of a ground graph-yielding relational causal model is called Relational Causal Discovery, which is also the name of first relational causal discovery algorithm by Maier et al. (2013a).

2.3 Two Semantics for Relational Path

We proceed to describe two semantics for interpreting relational paths, and hence translating relational dependencies of an RCM into causal relationships on attributes of items of a relational skeleton.

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 relational skeleton σ, where the item class of ℓth item of $\mathbf{p}$ is the ℓth item class of P for $1 \leq \ell \leq |P|$. Then,



(a) σ (b) $\mathcal{G}_{\sigma}^{\langle \mathcal{S}, \{D\} \rangle}$ (c) $\mathcal{G}_{\sigma}^{\langle \mathcal{S}, \{\tilde{D}\} \rangle}$

(d) σ' (e) $\mathcal{G}_{\sigma'}^{\langle \mathcal{S}, \{D\} \rangle}$ (f) $\mathcal{G}_{\sigma'}^{\langle \mathcal{S}, \{\tilde{D}\} \rangle}$ BBS (g) $\mathcal{G}_{\sigma'}^{\langle \mathcal{S}, \{\tilde{D}\} \rangle}$ PS

Figure 2.6: Comparison of ground graphs under bridge burning semantics and path semantics

under path semantics, a terminal set $P|_i^{\sigma}$ is simply defined as,

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

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 relational skeleton σ along P in breadth-first order starting at i. Formally, BBS defines $P|_i^{\sigma}$ iteratively as $P^{:1}|_i^{\sigma} := \{i\}$ and

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

The choice of BBS has following implications, which are not fully examined in previous RCM literature. We illustrate differences between BBS and path semantics in Figure 2.6. Let $\mathcal{S}$ be a relational schema with $\mathbf{E} = \{E_1, E_2, E_3\}$, $X \in \mathbf{A}(E_1)$, $Y \in \mathbf{A}(E_2)$, and $\mathbf{R} = \{R_1, R_2\}$ where $E_1, E_2 \in R_1$ and $E_2, E_3 \in R_2$ with cardinality greater than 1. $D = [E_2, R_2, E_3, R_2, E_2, R_1, E_1].X \to [E_2].Y$. Both semantics yield the same ground graphs (b), (c), and (e) for relational skeletons σ and σ'. However, the two semantics yield different ground graphs (f) and (g) for $\mathcal{M} = \langle \mathcal{S}, \{\tilde{D}\} \rangle$ for relational skeleton σ'.

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 2.6(f) with 2.6(c). The addition of two edges e_1–r'_1–e'_2 in σ' compared to σ makes $e'_2 \in [E_1, R_1, E_2]|^{\sigma'}_{e_1}$ and, hence, $e'_2 \notin [E_1, R_1, E_2, R_2, E_3, R_2, E_2]|^{\sigma'}_{e_1}$.

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 2.6(f) to 2.6(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 dissertation, we consider RCMs under the 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 is easier to analyze as we will see in the rest of dissertation.

Chapter 3
Reasoning about Conditional Independence among Relational Variables

This chapter focuses on the idea of generalizing d-separation, a graphical criterion to test conditional independence among variables under a Markov condition, to a relational setting. We explore previous attempts and offer new sound methods. This chapter reproduces some of the results from (Lee and Honavar, 2015, 2016b).

3.1 Relational d-Separation

D-separation (Verma and Pearl, 1988; Geiger et al., 1989) is a sound graphical criterion to obtain conditional independence of two variables in a (causal) Bayesian network given a set of conditionals.

Definition 10 (d-separation (Pearl, 1988))**.** A path between two variables, A and B, is said to be *d-separated* (blocked or closed) if:

1. The path contains a noncollider that has been conditioned on, for example, $A \rightarrow C \rightarrow B$ or $A \leftarrow C \rightarrow B$; or

2. The path contains a collider that has not been conditioned on, $A \rightarrow C \leftarrow B$, and no descendant of any collider on the path has been conditioned on either.

We often said that two variables are d-separated if there is no d-connection path exists. We provide an example of d-separation in Figure 3.1. The definition is similarly extended to a relational setting. We revisit the definition originally proposed by Maier et al. (2013b); Maier (2014).

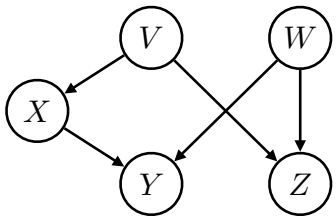


Figure 3.1: A Bayesian network with five variables. V and W are d-separated. However, V and W are d-connected if Y is given since Y is a collider. If X is additionally given, V and W are d-separated again.

Definition 11 (Relational d-separation)**.** Let $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ be three disjoint sets of relational variables of the same perspective $B \in \mathbf{I}$ defined over relational schema $\mathcal{S}$. Then, for relational model structure $\mathcal{M}$, $\mathbf{U}$, and $\mathbf{V}$ are *relationally d-separated* by $\mathbf{W}$ if and only if, for all relational skeletons $\sigma \in \Sigma_{\mathcal{S}}$, $\mathbf{U}|_b^\sigma$ and $\mathbf{V}|_b^\sigma$ are d-separated by $\mathbf{Z}|_b^\sigma$ in ground graph $\mathcal{G}_\sigma^{\mathcal{M}}$ for all $b \in \sigma(B)$:

$$\forall_{\sigma \in \Sigma_{\mathcal{S}}} \forall_{b \in \sigma(B)} \mathbf{U}|_b^\sigma \perp\!\!\!\perp \mathbf{V}|_b^\sigma \mid \mathbf{Z}|_b^\sigma$$

The definition claims that any violation of d-separation in any ground graph will be considered as d-connection. Relational d-separation, as relational causal model is, is defined at a template level.

Unlike traditional d-separation where we have a single given graphical structure $\mathcal{G}$, relational d-separation should cover all possible ground graphs from all relational skeletons $\Sigma_{\mathcal{S}}$. Maier et al. (2013b) proposed a method to "abstract" all possible ground graphs as a single (possibly infinitely large) directed acyclic graph to which traditional d-separation can be applied.

3.2 Abstract Ground Graphs: Revisited

We reproduce the definition of *Abstract Ground Graph* (AGG) and a basic theorem Maier and his colleagues proposed. The lifted representation seemed to play a central role in reasoning with and learning of RCM. The correctness of RCD, an algorithm proposed by Maier et al. (2013a) for learning RCM from data, relies on the *soundness* and *completeness* of AGG for *relational d-separation* to reduce the learning of an RCM to learning of an AGG.

We revisit the definition of AGG and show that AGG, as defined in Maier et al.

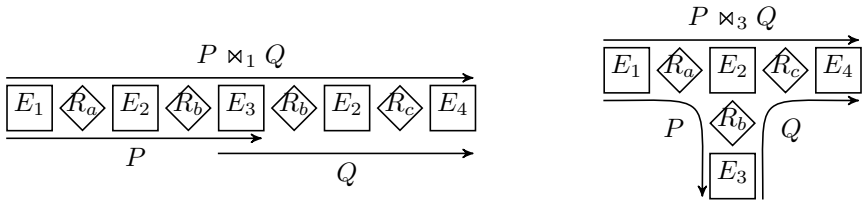


Figure 3.2: A schematic example of extend

(2013b), does *not* correctly abstract all ground graphs. We revise the definition of AGG to ensure that it correctly abstracts[1] all ground graphs. We further show that AGG representation is *not complete* for relational d-separation, that is, there can exist conditional independence relations in an RCM that are not entailed by AGG. As previously stated, we choose to use path semantics although Maier and his colleagues used bridge-burning semantics. This does not change our conclusion about AGGs.

Before we proceed to introduce AGGs, we define intersectability of two relational paths. Two relational paths P and P' of the same perspective B are said to be *intersectable* if and only if:

$$\exists_{\sigma \in \Sigma_{\mathbb{S}}} \exists_{b \in \sigma(B)} P|_b^\sigma \cap P'|_b^\sigma \neq \emptyset. \tag{3.1}$$

Intersectability makes the problem of relational d-separation more difficult in addition to the all-ground-graphs formulation. Note that intersectability can be defined for two relational variables. Two relational variables $P.X$ and $P'.Y$ are intersectable if and only if $X = Y$ and P and P' are intersectable. Trivially, two equal relational paths or variables are intersectable. We denote by $P \cap^* P'$ intersectability of two relational paths (or relational variables).

3.2.1 Abstract Ground Graphs, the Original

An abstract ground graph $AGG_{\mathcal{M}B}$ is defined for a given relational model $\mathcal{M}$ and a perspective $B \in \mathbf{I}$ (Maier et al., 2013a), Since we fix the model, we omit the subscript $\mathcal{M}$ and denote the abstract ground graph for perspective B by AGG_B. The resulting graph consists of two types of vertices: $\mathbf{RV}_B$ and $\mathbf{IV}_B$; and two types of edges: $\mathbf{RVE}_B$ and $\mathbf{IVE}_B$.

[1] We will later investigate the mathematical definition of 'abstracts'.

We denote by $\mathbf{RV}_B$ the set of *all* relational variables (RV) whose paths originate from B. We denote by $\mathbf{RVE}_B$ the set of all edges between the relational variables in $\mathbf{RV}_B$. A relational variable edge (RVE) implies *direct* influence arising from one or more dependencies in $\mathbf{D}$. There is an RVE $P.X \to Q.Y$ if there exists a dependency $R.X \to V_Y \in \mathbf{D}$ that can be interpreted as a direct influence from $P.X$ to $Q.Y$ from perspective B. Such an interpretation is originally implemented by an extend function (Maier, 2014), which takes two relational paths and produces a set of relational paths: If $P \in \mathsf{extend}(Q, R)$, then there exists an RVE $P.X \to Q.Y$ where

$$\mathsf{extend}(Q, R) := \{Q^{1:|Q|-i} + R^{i:} | i \in \mathsf{pivots}(\tilde{Q}, R)\} \cap \mathbf{P}_\mathcal{S}, \tag{3.2}$$

We will use a binary *join* operator '$\bowtie$' for extend and denote $Q^{1:|Q|-i} + R^{i:}$ by $Q \bowtie_i R$ for a pivot i as mentioned earlier. A schematic overview of extend is shown in Figure 3.2. It demonstrates two relational paths in $P \bowtie Q$ where $\mathsf{card}(R_b, E_3) = \mathsf{many}$. If $\mathsf{card}(R_b, E_3)$ is one, then $P \bowtie_1 Q$ is not valid for both BBS and path semantics. A path $P \bowtie_2 Q$ is also invalid, i.e., $[\ldots, E_2, R_b, E_2, \ldots]$, for both semantics.

We denote by $\mathbf{IV}_B$ the set of *intersection variables* (IVs), i.e., unordered pairs of *intersectable* relational variables in $\mathbf{RV}_B$. Given two *different* RVs $P.X$ and $P'.X$ that are intersectable with each other, we denote the resulting intersection variable by $P.X \cap P'.X$ (Here, the intersection symbol '$\cap$' connotes *intersectability* of the two relational variables). By the definition (Maier et al., 2013b), if there exists an RVE $P.X \to Q.Y$, then there exist edges $P.X \cap P'.X \to Q.Y$ and $P.X \to Q.Y \cap Q'.Y$ for every P' and Q' intersectable with P ($\neq P'$) and Q ($\neq Q'$), respectively. The IVs and the edges that connect them with RVs (IVEs) correspond to *indirect influences* (arising from intersectability) as opposed to *direct* influence due to dependencies (which are covered by RVs and RVEs). We denote by $\mathbf{IVE}_B$ the set of all such edges that connect RVs with IVs.

Two AGGs with different perspectives share no vertices nor edges. Hence, we view all AGGs, $\{AGG_B\}_{B \in \mathbf{I}}$, as a collection or a single multi-component graph $\mathbf{AGG} := \bigcup_{B \in \mathbf{I}} AGG_B$. We similarly define $\mathbf{RV}$, $\mathbf{IV}$, $\mathbf{RVE}$, and $\mathbf{IVE}$ as the unions of their perspective-based counterparts.

For any mutually disjoint sets of relational variables $\mathbf{U}, \mathbf{V}$, and $\mathbf{W}$, one can test $\mathbf{U} \perp\!\!\!\perp \mathbf{V} \mid \mathbf{W}$, conditional independence admitted by the underlying probability distribution, by checking $\bar{\mathbf{U}} \perp\!\!\!\perp \bar{\mathbf{V}} \mid \bar{\mathbf{W}}$ (traditional) d-separation on an AGG, where $\bar{\mathbf{V}}$ includes $\mathbf{V}$ and their related IVs, $\bar{\mathbf{V}} := \mathbf{V} \cup \{V \cap T \in \mathbf{IV} \mid V \in \mathbf{V}\}$.

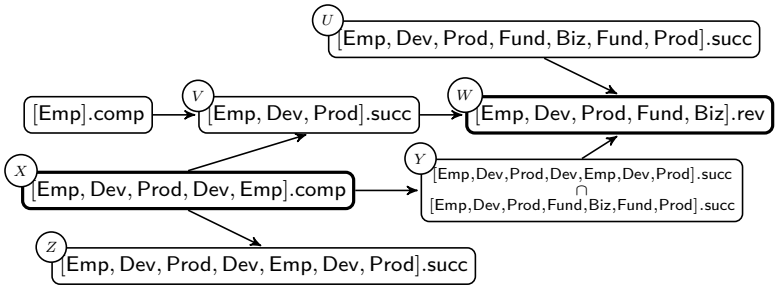


Figure 3.3: An AGG example excerpted from (Maier, 2014)

Figure 3.3 illustrates relational d-separation on an AGG. The Revenue of business units that fund the products developed by an employee (W) is affected by the employee's co-workers' competence (X), i.e., $\bar{W} \not\perp\!\!\!\perp \bar{X}$. Two are conditionally independent by blocking both V and Y. Since IV Y is in $\bar{U}$ and $\bar{Z}$, both $\bar{W} \perp\!\!\!\perp \bar{X}|\overline{\{V,U\}}$ and $\bar{W} \perp\!\!\!\perp \bar{X}|\overline{\{V,Z\}}$ hold, which are equivalent to $(W \perp\!\!\!\perp X|\{V,U\})_{\mathcal{M}}$ and $(W \perp\!\!\!\perp X|\{V,Z\})_{\mathcal{M}}$, respectively.

3.2.2 Extend and RVE

In (Lee and Honavar, 2015), we showed that how the definitions of IVs and IVEs are problematic. However, we found out that the definition of RVEs based on the function extend is also problematic. The purpose of an RVE in AGG, e.g., $R.X \to P.Y$, is to provide the causes of a non-canonical relational variable. One can interpret RVE as a relational dependency observed from different perspective, say from $\tilde{P}$ away.

To consider $R.X \to P.Y$, R must satisfy:

$$\exists_{Q.X \to V_Y \in \mathbf{D}} \exists_{\sigma \in \Sigma_{\mathcal{S}}} \exists_{b \in \sigma(B)} \exists_{j \in P|_b^\sigma} \exists_{k \in Q|_j^\sigma} k \in R|_b^\sigma.$$

However, the above condition is a sufficient condition for R. Understanding the purpose of AGG as a means to perform d-separation, parents of $P.Y$ should not be redundant. Then, a necessary condition is

$$\exists_{Q.X \to V_Y \in \mathbf{D}} \exists_{\sigma \in \Sigma_{\mathcal{S}}} \exists_{b \in \sigma(B)} \exists_{j \in P|_b^\sigma} \exists_{k \in Q|_j^\sigma} \forall_{S \in \mathbf{P}_{\mathcal{S}} \setminus \{R\}} k \notin S|_b^\sigma.$$

We provide an algorithm that generates parents (in the context of AGG) of a non-canonical relational variable where the design of the algorithm has its root

in Canonical Unshielded Triple (CUT) enumeration algorithm (Lee and Honavar, 2016a). The CUT enumeration algorithm generates relational skeletons made of items corresponding to two relational paths to be 'joined', P and Q. The algorithm is complete in a sense that *any* relational skeletons containing i, j, and k such that $j \in P|_i$ and $k \in Q|_j$ embeds (i.e., subgraph isomorphism) one of the generated relational skeleton. CUT enumeration algorithm focuses on R such that $k \in R|_i$. There are three categories of relational paths the CUT enumeration algorithm generates. Among these, the first category, R_r, satisfies both sufficient and necessary conditions.

For the algorithm, we define $\mathsf{LLRSP}(P, Q)$ (*the length* of *the longest required shared path*) for two relational paths P and Q of the common perspective as

$$\mathsf{LLRSP}(P, Q) := \max\{\ell \mid P^{1:\ell} = Q^{1:\ell},\ \forall_{\sigma\in\Sigma_\mathrm{s}}\forall_{b\in\sigma(B)} \left|P^{1:\ell}|^{\sigma}_{b}\right| = 1\}. \tag{3.3}$$

$\mathsf{LLRSP}(P, Q)$ is computed as follows. Initially set $\ell = 1$ since $P^1 = Q^1$. Repeat incrementing ℓ by 1 if $P^{\ell+1} = Q^{\ell+1}$ and either $P^\ell \in \mathbf{R}$ or $P^\ell \in \mathbf{E}$ with $\mathsf{card}(P^\ell, P^{\ell+1}) = \mathsf{one}$.

Algorithm 1 Pseudocode for newextend

procedure NEWEXTEND(P, Q, $\mathcal{M}$)
Prerequisite: $P^{|P|} = Q^1$
  $\mathsf{LL} := \mathsf{LLRSP}$
  $m := |P|$
  $\ell := \mathsf{LL}(\tilde{P}, Q)$
  $\mathbf{J} := \left\{(a, b) \mid P^a = Q^b,\ 1 \le a \le m - \ell + 1,\ \ell \le b \le |Q|\right\}$
  for (a_r, b_r) **in J such that** $\mathsf{LL}(P^{:a_r:-1}, Q^{b_r:}) = \mathsf{LL}(P^{a_r:}, Q^{b_r:}) = 1$ **do**
    $\ell_\alpha := \mathsf{LL}(Q^{\ell:b_r:-1}, P^{:a_r:-1})$
    if $\ell_\alpha = 1$ **and** $P^{a_r:m-\ell+1} \cap^* Q^{\ell:b_r:-1}$ **then**
      yield $R_r := P^{:a_r} \oplus Q^{b_r:}$
    if $1 < \ell_\alpha < b_r - \ell + 1$ and $a_r < m - \ell + 1$ and $\ell < b_r$ **then**
      yield $R_r := P^{:a_r} \oplus Q^{b_r:}$

The newextend algorithm (Algorithm 1) subsumes what extend can yield – extend is a special case where $\ell_\alpha = 1$ and $P^{a_r:m-\ell+1} = Q^{\ell:b_r:-1}$ (hence, simply $P^{a_r:} = Q^{\ell:b_r:-1}$.) Now, RVEs pointing to a relational variable, e.g., $P.Y$, of an AGG

can be constructed as follows:

$$\{R.X \to P.Y \mid R \in \mathsf{newextend}(P, Q),\ Q.X \to V_Y \in \mathbf{D}, P^{|P|} = Q^1\}$$

3.2.3 Intersectability and IV

The declarative characterization of *intersectability* (Equation 3.1) does not offer practical procedural criteria to determine *intersectability*. Based on the criteria (Maier, 2014), two different relational paths P and Q are *intersectable* if and only if 1) they share the same perspective, say $B \in \mathbf{I}$, and 2) they share the common terminal class, and 3) one path is *not* a prefix of the other. The third one is only applied for bridge-burning semantics. We will prove that the preceding criteria are *not sufficient*.

Lemma 12. *Given a relational schema $\mathcal{S}$, let P and Q be two different relational paths satisfying the (necessary) criteria of Maier (2014)*[2] *and $|Q| \leq |P|$. Let m and n be $\mathsf{LLRSP}(P, Q)$ and $\mathsf{LLRSP}(\tilde{P}, \tilde{Q})$, respectively. Then, P and Q are intersectable if and only if $m + n \leq |Q|$.*

Proof. (If part) If $m + n \leq |Q|$, then we can construct a relational skeleton σ such that $P|_b^\sigma \cap Q|_b^\sigma \neq \emptyset$ for some $b \in \sigma(P^1)$ by adding unique items for Q and for $P^{m+1:|P|-n}$ and complete the skeleton in the same manner as shown in Lemma 3.4.1 (Maier, 2014). Note that if $m + n = |Q|$, then $|P| \geq |Q| + 2$ since $P \neq Q$ and a relational path is an alternating sequence. This guarantees that there are at least two items for $P^{m+1:|P|-n}$.

(Only if part) Let c be in $P|_b^\sigma \cap Q|_b^\sigma$ for some arbitrary skeleton $\sigma \in \Sigma_\mathcal{S}$ and $b \in \sigma(P_1)$. Then, there should be two lists of items corresponding to P and Q sharing the first m and the last n. The condition $m + n > |Q|$ implies $Q|_b^\sigma$ is a singleton set. We define

$$\mathbf{p} = \langle p_1, \ldots, p_m, p_{|P|-n+1}, \ldots, p_{|P|} \rangle$$

and

$$\mathbf{q} = \langle q_1, \ldots, q_{|Q|} \rangle,$$

[2]For the path semantics, the third condition is ignored as mentioned earlier.

$$
\begin{array}{lll}
1)\ \exists_{\sigma\in\Sigma_{\mathcal{S}}}\exists_{b\in\sigma(B)}\exists_{i_j\in Q|_b^\sigma} & R|_{i_j}^\sigma \cap P|_b^\sigma & \neq \emptyset \\
2)\ \exists_{\sigma\in\Sigma_{\mathcal{S}}}\exists_{b\in\sigma(B)} & P|_b^\sigma \cap P'|_b^\sigma & \neq \emptyset \\
3)\ \exists_{\sigma\in\Sigma_{\mathcal{S}}}\exists_{b\in\sigma(B)}\exists_{i_j\in Q|_b^\sigma} & R|_{i_j}^\sigma \cap P|_b^\sigma \cap P'|_b^\sigma & \neq \emptyset
\end{array}
$$

Figure 3.4: Comparison of 1) the necessary condition of the existence of an RVE $P \to Q$ through R, the cause path of a dependency (attribute classes are omitted), 2) intersectability between P and P', and 3) co-intersectability of $\langle Q, R, P, P'\rangle$.

where $\{q_\ell\} = Q^{1:\ell}|_b^\sigma$ and $\{p_\ell\} = P^{1:\ell}|_b^\sigma$ for $1 \leq \ell \leq m$, and $p_{|P|-l+1} \in \tilde{P}^{1:l}|_c^\sigma$ for $1 \leq l \leq n$. We can see that $p_1 = q_1 = b$ and $p_{|P|} = q_{|Q|} = c$. Moreover,

$$p_m = q_m = q_{|Q|-(|Q|-m)} = p_{|P|-(|Q|-m)}$$

by the definition of LLRSP. If $|Q| < |P|$, then $m \neq |P| - |Q| + m$ and mth item for P is repeated at $|P| - (|Q| - m)$th, which violates both semantics. Otherwise, it is not the case, since $|P| = |Q|$ implies $\mathbf{p} = \mathbf{q}$ and, hence, $P = Q$ by the definition of LLRSP, which contradicts the assumption that P and Q are different relational paths. □

The lemma demonstrates the criteria by Maier (2014) do not rule out the case of $m + n > |Q|$ where P and Q cannot be intersectable.

3.2.4 Co-intersectability and IVE

Based on the definition (Maier, 2014), an IVE exists between an IV, $U \cap V$, and an RV, W, if and only if there exists an RVE between U and W or V and W. It would indeed be appealing to define IV, $U \cap V$, such that it inherits properties of the corresponding RVs, U and V. However, the abstract ground graph resulting from such a definition turns out to be not a sound representation of the underlying ground graphs. We proceed to prove this result.

Definition 13 (Co-intersectability)**.** Given a relational schema $\mathcal{S}$, let Q, R, P, and P' be valid relational paths of the same perspective B where $P \in \mathsf{newextend}(Q, R)$ and P and P' are intersectable. Then, a tuple $\langle Q, R, P, P'\rangle$ is said to be *co-*

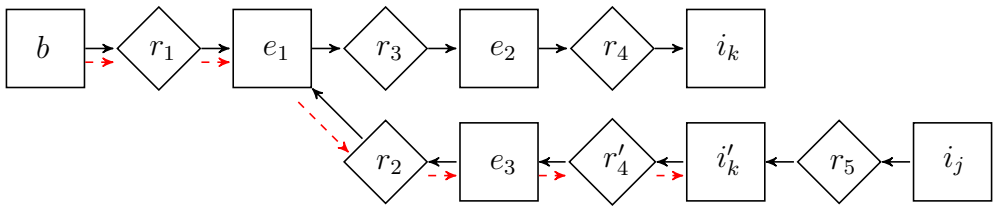


Figure 3.5: A schematic illustration of Example 14 superimposing a relational skeleton and relational paths

intersectable if and only if

$$\exists_{\sigma\in\Sigma_{\mathcal{S}}}\exists_{b\in\sigma(B)}\exists_{i_j\in Q|_b^\sigma} R|_{i_j}^\sigma \cap P|_b^\sigma \cap P'|_b^\sigma \neq \emptyset. \tag{3.4}$$

We relate co-intersectability with the definition of IVE. Let an RVE $P.X \to Q.Y$ be due to some dependency $R.X \to V_Y \in \mathbf{D}$ where $P \in \mathsf{newextend}(Q, R)$. This implies

$$\exists_{\sigma\in\Sigma_{\mathcal{S}}}\exists_{b\in\sigma(B)}\exists_{i_j\in Q|_b^\sigma} R|_{i_j}^\sigma \cap P|_b^\sigma \neq \emptyset, \tag{3.5}$$

and there are edges from X of $R|_{i_j}^\sigma \cap P|_b^\sigma$ to Y of $Q|_b^\sigma$ in $\mathcal{G}_\sigma^{\mathcal{M}}$. In order for the intersectability of P' with P translates into an influence between P and Q, it is necessary that there exists a relational skeleton that admits such influence. However, we can construct a counterexample that satisfies the necessary conditions for the existence of an RVE and the conditions for intersectability but does *not* satisfy the conditions for co-intersectability (see Figure 3.4 for a comparison of Equation 3.5, 3.1, and 3.4). Given the fact that $\mathsf{newextend}(Q, R) \supseteq \mathsf{extend}(Q, R)$ for any valid arguments Q and R, the following example applies not only to newextend but also to extend.

Example 14. Let $\mathcal{S}$ be a relational schema where $\mathbf{E} := \{I_j, I_k, B, E_1, E_2, E_3\}$, $\mathbf{R} := \{R_i\}_{i=1}^5$ such that $R_1 := \langle B, E_1\rangle$, $R_2 := \langle E_1, E_3\rangle$, $R_3 := \langle E_1, E_2\rangle$, $R_4 := \langle E_2, E_3, I_k\rangle$, $R_5 := \langle I_k, I_j\rangle$ with the cardinality of each relationship and each entity in the relationship being one. Let

- $Q = [B, R_1, E_1, R_2, E_3, R_4, I_k, R_5, I_j]$,
- $R = [I_j, R_5, I_k, R_4, E_3, R_2, E_1, R_3, E_2, R_4, I_k]$,
- $P = [B, R_1, E_1, R_3, E_2, R_4, I_k]$, and
- $P' = [B, R_1, E_1, R_2, E_3, R_4, I_k]$.

Observe that

1. $P \in \mathsf{newextend}\,(Q, R)$;
2. P' and P are intersectable; and
3. P' is a subpath of Q.

This example satisfies Equation 3.1 and Equation 3.5. Assume for contradiction that there exists a relational skeleton σ satisfying Equation 3.4. Since, in this example, the cardinality of each relationship and each entity in the relationship is one, for each $b \in \sigma\,(B)$, there exists only one $i_j \in Q|_b^\sigma$ and only one $i_k \in P|_b^\sigma$. By the assumption, $P'|_b^\sigma = \{i_k\}$. Since P' is a subpath of Q, $P'|_b^\sigma$ will end at $i'_k = R^{1:3}|_{i_j}^\sigma$. See Figure 3.5. The items for P' starting with b should follow a dashed red line, and, hence, P' cannot be related to a ground graph edge between i_k and i_j, i.e., an RVE between P and Q (attributes and connections between entities and relationships are omitted). Due to common characteristics of BBS and path semantics, $R|_{i_j}^\sigma \cap R^{1:3}|_{i_j}^\sigma = \emptyset$, that is, $\{i_k\} \cap \{i'_k\} = \emptyset$. This contradicts the assumption that $i_k = i'_k$.

This counterexample clearly represents there is an inter-dependency between intersection variables and RVEs. Therefore, we revise the definition of **IVE** accompanying co-intersectability.

Definition 15 (IVE)**.** There exists an IVE edge, $P.X \cap P'.X \to Q.Y$ (or $P.X \to Q.Y \cap Q'.Y$), if and only if there exists a relational path R such that $R.X \to V_Y \in \mathbf{D}$, $P \in \mathsf{newextend}(Q, R)$, and $\langle Q, R, P, P' \rangle$ (or $\langle P, \tilde{R}, Q, Q' \rangle$) is *co-intersectable*.

To determine IVEs, *co-intersectability* of a tuple can be easily solved by encoding constraints (e.g., path constraints for path semantics and cardinality constraints). There are four relational paths where corresponding items to be unique within each relational path. Constraints are imposed by the definition of co-intersectability so that $p_1 = p'_1 = q_1$, $q_{|Q|} = r_1$ and $r_{|R|} = p_{|P|} = p_{|P'|}$ where $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$, and $\mathbf{p}'$ are items for P, Q, R, and P', respectively. Iteratively merging items to meet cardinality constraints while checking violations of path constraints will either fail or yield a valid relational skeleton made of items for P, Q, R, and P' (see Algorithm 2).

We (i) have investigated original definitions of RVE, IV, and IVE and (ii) revised the definitions based on $\mathsf{newextend}$, intersectability, and co-intersectability so as to (iii) guarantee that AGG correctly abstracts all ground graphs as asserted

Algorithm 2 Pseudocode for co-intersectability

procedure CO-INTERSECTABILITY(Q,R,P,P',$\mathcal{M}$)
  Initialize $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$, and $\mathbf{p}'$ with unique items
  Let $\mathbf{I}(\cdot)$ an item class of a given item
  $p_1 := p'_1 := q_1$
  $p_{|P|} := p'_{|P'|} := r_{|R|}$
  $q_{|Q|} := r_1$
  Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \mathbf{p} \cup \mathbf{q} \cup \mathbf{r} \cup \mathbf{p}'$ and $\mathcal{E}$ is a set of pairs adjacent in $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$, and $\mathbf{p}'$.
  while True **do**
    for $v \in \mathcal{V}$ **do**
      if $\mathbf{I}(v) \in \mathbf{R}$ **then**
        for $E \in \mathbf{I}(v)$ **do**
          if $|\{w \mid w \in adj(v; \mathcal{G}), \mathbf{I}(w) = E\}| > 1$ **then**
            $\mathbf{m} := \{w \mid w \in adj(v; \mathcal{G}), \mathbf{I}(w) = E\}$
            if $\exists_{\mathbf{x} \in \mathbf{p},\mathbf{q},\mathbf{r},\mathbf{p}'} |\mathbf{x} \cap \mathbf{m}| > 1$ **then**
              return False
            replace every $\mathbf{m} \setminus \{m\}$ to $m \in \mathbf{m}$ for $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$, $\mathbf{p}'$, and $\mathcal{G}$.
            continue to **while**
      else if $\mathbf{I}(v) \in \mathbf{E}$ **then**
        for $R \in \mathbf{R}$ **such that** $\mathbf{I}(v) \in R$ **do**
          if card$(R, \mathbf{I}(v)) = 1$ **and** $|\{w \in adj(v; \mathcal{G}) \mid \mathbf{I}(w) = R\}| > 1$ **then**
            $\mathbf{m} := \{w \mid w \in adj(v; \mathcal{G}), \mathbf{I}(w) = R\}$
            if $\exists_{\mathbf{x} \in \mathbf{p},\mathbf{q},\mathbf{r},\mathbf{p}'} |\mathbf{x} \cap \mathbf{m}| > 1$ **then**
              return False
            replace every $\mathbf{m} \setminus \{m\}$ to $m \in \mathbf{m}$ for $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$, $\mathbf{p}'$, and $\mathcal{G}$.
            continue to **while**
    return True

(although incorrectly) by Theorem 4.5.2 (Maier, 2014). Revised AGG abstracts all ground graphs so that (i) there exists an edge in an AGG only if there exists a corresponding edge in a ground graph, and (ii) every edge in every possible ground graph is covered by an edge in an AGG.

The new criterion, called *co-intersectability,* is especially interesting since it describes the interdependency between intersection variables and related relational variable edges. Several key results (e.g., soundness and completeness of AGG for relational d-separation, Theorem 4.5.2) and concepts (e.g., (B,h)-reachability) of Maier (2014) are based on *independence* between intersection variables and related relational variable edges. Hence, the next step is to carefully scrutinize the

relationship between AGGs and relational d-separation.

3.3 Non-completeness of Abstract Ground Graph for Relational d-Separation

We first revisit the definition of relational d-separation. Given three disjoint sets of relational variables $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ of a common perspective $B \in \mathbf{I}$, $\mathbf{U}$ and $\mathbf{V}$ are relational d-separated given $\mathbf{W}$, denoted by $\mathbf{U} \perp\!\!\!\perp \mathbf{V} \mid \mathbf{W}$, if and only if

$$\forall_{\sigma \in \Sigma_{\mathcal{S}}} \forall_{b \in \sigma(B)} \left(\mathbf{U}|_b \perp\!\!\!\perp \mathbf{V}|_b \mid \mathbf{W}|_b\right)_{\mathcal{G}^{\mathcal{M}}_{\sigma}} .$$

From Theorem 4.5.4 of (Maier, 2014), the lifted representation $\mathbf{AGG}_{\mathcal{M}}$ is said to be sound (or complete) for relational d-separation of $\mathcal{M}$ if traditional d-separation holds on the $\mathbf{AGG}_{\mathcal{M}}$ with a modified CI query only when (or whenever) relational d-separation holds true. Then, the completeness of AGG for relational d-separation can be represented as

$$\mathbf{U} \perp\!\!\!\perp \mathbf{V} \mid \mathbf{W} \Rightarrow \left(\bar{\mathbf{U}} \perp\!\!\!\perp \bar{\mathbf{V}} \mid \bar{\mathbf{W}}\right)_{\mathbf{AGG}_{\mathcal{M}}} .$$

The completeness can be proved by the construction of a relational skeleton $\sigma \in \Sigma_{\mathcal{S}}$ demonstrating d-connection $(\mathbf{U}|^{\sigma}_{b} \not\perp\!\!\!\perp \mathbf{V}|^{\sigma}_{b} \mid \mathbf{W}|^{\sigma}_{b})_{\mathcal{G}^{\mathcal{M}}_{\sigma}}$ for some $b \in \sigma(B)$ if $(\bar{\mathbf{U}} \not\perp\!\!\!\perp \bar{\mathbf{V}} \mid \bar{\mathbf{W}})_{\mathbf{AGG}_{\mathcal{M}}}$. In other words, we might disprove the completeness by showing

$$(\bar{\mathbf{U}} \not\perp\!\!\!\perp \bar{\mathbf{V}} \mid \bar{\mathbf{W}})_{\mathbf{AGG}_{\mathcal{M}}} \wedge \forall_{\sigma \in \Sigma_{\mathcal{S}}} \forall_{b \in \sigma(B)} \left(\mathbf{U}|^{\sigma}_{b} \perp\!\!\!\perp \mathbf{V}|^{\sigma}_{b} \mid \mathbf{W}|^{\sigma}_{b}\right)_{\mathcal{G}^{\mathcal{M}}_{\sigma}} .$$

The following counterexample shows that AGG is not complete for relational d-separation.

Example. Let $\mathcal{S} := \langle \mathbf{E}, \mathbf{R}, \mathbf{A}, \mathsf{card} \rangle$ be a relational schema such that: $\mathbf{E} = \{E_i\}_{i=1}^{5}$; $\mathbf{R} = \{R_j\}_{j=1}^{3}$ with $R_1 = \langle E_1, E_2, E_4 \rangle$, $R_2 = \langle E_2, E_3 \rangle$, and $R_3 = \langle E_3, E_4, E_5 \rangle$; $\mathbf{A} = \{E_2 : \{Y\}, E_3 : \{X\}, E_5 : \{Z\}\}$; and $\forall_{R \in \mathbf{R}} \forall_{E \in R} \mathsf{card}(R, E) = \mathsf{one}$. Let $\mathcal{M} := \langle \mathcal{S}, \mathbf{D} \rangle$ be a relational causal model with

$$\mathbf{D} := \{D_1.X \rightarrow V_Y,\ D_2.Z \rightarrow V_Y\}$$

such that $D_1 := [E_2, R_2, E_3, R_3, E_4, R_1, E_2, R_2, E_3]$ and $D_2 := [E_2, R_2, E_3, R_3, E_5]$.

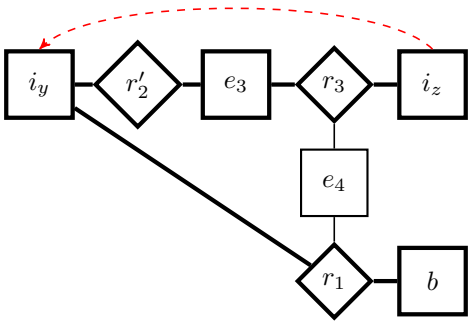


Figure 3.6: Co-intersectability of $\langle Q, D_2, S, S' \rangle$

Let $P.X$, $Q.Y$, $S.Z$, and $S'.Z$ be four relational variables of the same perspective $B = E_1$ where their relational paths are distinct where

- $P := [E_1, R_1, E_2, R_2, E_3]$,
- $Q := [E_1, R_1, E_4, R_3, E_3, R_2, E_2]$,
- $S := [E_1, R_1, E_4, R_3, E_5]$, and
- $S' := [E_1, R_1, E_2, R_2, E_3, R_3, E_5]$.

Given the above example, we can make two claims. Before we proceed, co-intersectability of $\langle Q, D_2, S, S' \rangle$ holds true (see Figure 3.6) where $i_y \in Q|_b^\sigma$, $i_z \in D_2|_{i_y}^\sigma$, $i_z \in S|_b^\sigma$, and $i_z \in S'|_b^\sigma$. Thick lines highlight items for S' from b to i_z. The red dashed line represents the instantiation of an RVE $S.Z \to Q.Y$ as $i_z.Z \to i_y.Y$ in a ground graph (item attributes are omitted).

Claim 16. $\left(\overline{P.X} \not\perp\!\!\!\perp \overline{S'.Z} \mid \overline{Q.Y}\right)_{\mathbf{AGG}_\mathcal{M}}$.

Proof. By the definition of RVE, there are RVEs $P.X \to Q.Y$ and $Q.Y \leftarrow S.Z$ in $\mathbf{AGG}_\mathcal{M}$ since $P = Q \bowtie_6 D_1$ and $S \in Q \bowtie_4 D_2$. Moreover, there is an IVE $Q.Y \leftarrow S.Z \cap S'.Z$ in $\mathbf{AGG}_\mathcal{M}$ since 1) S and S' are *intersectable,* 2) there is an RVE $Q.Y \leftarrow S.Z$, and 3) $\langle Q, D_2, S, S' \rangle$ is *co-intersectable*.[3] Since $P.X \to Q.Y \leftarrow S.Z \cap S'.Z$ and $S.Z \cap S'.Z \in \overline{S'.Z}$, we derive $\left(P.X \not\perp\!\!\!\perp \overline{S'.Z} \mid Q.Y\right)_{\mathbf{AGG}_\mathcal{M}}$, which implies $\left(\overline{P.X} \not\perp\!\!\!\perp \overline{S'.Z} \mid Q.Y\right)_{\mathbf{AGG}_\mathcal{M}}$. Furthermore, conditioning on $\overline{Q.Y}$, compared to $Q.Y$, does not block any possible d-connection paths between $\overline{P.X}$ to $\overline{S'.Z}$ since there are only incoming edges to $\overline{Q.Y}$. Finally, $\left(\overline{P.X} \not\perp\!\!\!\perp \overline{S'.Z} \mid \overline{Q.Y}\right)_{\mathbf{AGG}_\mathcal{M}}$ holds. □

[3]Note that the original definition of $\mathbf{AGG}_\mathcal{M}$ does not check *co-intersectability* and $Q.Y \leftarrow S.Z \cap S'.Z$ is granted.

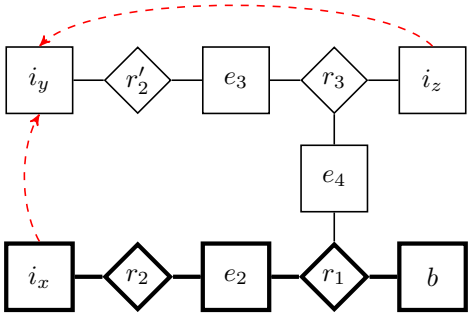


Figure 3.7: A subgraph of ground graphs representing $i_x \to i_y \leftarrow i_z$

Assuming that AGG is complete for relational d-separation, we can infer $(P.X \not\perp\!\!\!\perp S'.Z \mid Q.Y)_{\mathcal{M}}$ and there must exist a pair of a relational skeleton σ and a base item $b \in \sigma(B)$ that satisfies $(P.X|_b^\sigma \not\perp\!\!\!\perp S'.Z|_b^\sigma \mid Q.Y|_b^\sigma)_{\mathcal{G}_\sigma^{\mathcal{M}}}$. However, we claim that such a relational skeleton and base item may not exist.

Claim 17. There is no $\sigma \in \Sigma_{\mathcal{S}}$ and $b \in \sigma(B)$ such that

$$(P.X|_b^\sigma \not\perp\!\!\!\perp S'.Z|_b^\sigma \mid Q.Y|_b^\sigma)_{\mathcal{G}_\sigma^{\mathcal{M}}}.$$

Proof. Suppose that there exist such a relational skeleton σ and base item $b \in \sigma(B)$ satisfying $(P.X|_b^\sigma \not\perp\!\!\!\perp S'.Z|_b^\sigma \mid Q.Y|_b^\sigma)_{\mathcal{G}_\sigma^{\mathcal{M}}}$. Every terminal set for P, Q, and S' given the base item must not be empty because of the definition of d-separation and the fact that attribute classes X and Z are connected only through Y (i.e., Y is a collider). Since every cardinality is one, terminal sets must be singletons. Let $\{i_x\} = P.X|_b^\sigma$, $\{i_y\} = Q.Y|_b^\sigma$, and $\{i_z\} = S'.Y|_b^\sigma$. Furthermore, since i_x and i_z must be d-connected given i_y, $\mathcal{G}_\sigma^{\mathcal{M}}$ must have two edges $i_x \to i_y \leftarrow i_z$, which requires $i_x \in D_1|_{i_y}^\sigma$ and $i_z \in D_2|_{i_y}^\sigma$. However, due to semantics (both BBS and path semantics) and cardinality constraints (i.e., one), there exists only one possible structure (see Figure 3.7) where i_x and i_z are the cause of i_y while satisfying all previously mentioned conditions except $\{i_z\} = S'.Y|_b^\sigma$. In other words, the constraint $\{i_z\} = S'.Y|_b^\sigma$ violates with the set of the rest of conditions. Hence, there exists no such relational skeleton and base item. □

The counterexample demonstrates that a d-connection path captured in an $\mathbf{AGG}_{\mathcal{M}}$ might not have a corresponding d-connection path in *any* ground graph.

Corollary 18. *The revised (as well as the original) abstract ground graph for an RCM is not complete for relational d-separation.*

It is possible that an additional test can be utilized to check whether there *exists* such a ground graph that can represent a d-connection path captured in $\mathbf{AGG}_{\mathcal{M}}$. However, the efficiency of such an additional test is unknown and designing such a test is beyond the scope of this dissertation.

Our counterexamples involves the use of one cardinality. However, it is shown that there are counterexamples even with no one cardinality (where cardinality function always returns many). For example, AGG-based relational d-separation is problematic even with the popular company example given by Maier (2014) – either as is or with changing the cardinality between Funds and Product with many. Readers can check the following example,

$$[\mathsf{D},\mathsf{E}].\mathsf{Competence} \perp\!\!\!\perp [\mathsf{D},\mathsf{P},\mathsf{F},\mathsf{B}].\mathsf{Budget} \mid [\mathsf{D},\mathsf{P},\mathsf{F},\mathsf{B}].\mathsf{Revenue}$$

An AGG-based method identifies a d-connection path of

$$[\mathsf{DE}].\mathsf{Competence} \rightarrow [\mathsf{DEDP}] \cap [\mathsf{DEDPFBFP}].\mathsf{Success} \rightarrow [\mathsf{DEDPFB}].\mathsf{Revenue} \rightarrow [\mathsf{DEDPFB}] \cap [\mathsf{DPFB}].\mathsf{Budget}.$$

Simply, [DPFB].Revenue is the only parent of [DPFB].Budget, and Competence is non-descendant of Budget. This case is covered by relational causal Markov condition. This closes the problem of whether AGG-based methods are sound and complete when all cardinalities are many and relationships are binary.

3.4 Sound Relational d-Separation Methods

We propose several methods that can partially answer whether relational d-separation or relational d-connection holds true. First method utilizes the class dependency graph of an RCM based on the attribute class level acyclicity assumption. The method can soundly tell relational d-separation. Second method uses a randomized relational skeleton. One can construct a ground graph from a randomly generated relational skeleton. Then, any d-connection path in the grounded space provides an evidence for relational d-connection. Third method makes use of RVEs

of (revised) Abstract Ground Graph. The last method is a constructive-approach, which explores possible d-connection paths. The method is sound in relational d-connection and complete for relational d-separation considering a prespecified finite length of d-connection paths.

3.4.1 Class Dependency Graph-Based Approach

Consider a relational conditional independence query $U \perp\!\!\!\perp V \mid \mathbf{W}$. Assume that there exists, in some $\mathcal{G}^{\mathcal{M}}_{\sigma}$, a d-connection path made of item attributes. There exists a sequence of corresponding attribute classes. Unlike the d-connection path, the sequence can be repetitive. Since a d-connection path is based on a set of relational dependencies and the CDG $\mathcal{G}^{\mathcal{M}}_{\mathbf{A}}$ is $\mathcal{M}$ projected to $\mathbf{A}$, the sequence is a valid walk in $\mathcal{G}^{\mathcal{M}}_{\mathbf{A}}$. Let $\mathbf{w} := [w_1,\ w_2,\ ,\cdots,\ w_m]$ be such a walk. The walk must satisfy that if $w_{i-1} \rightarrow w_i \leftarrow w_{i+1}$ in $\mathcal{G}^{\mathcal{M}}_{\mathbf{A}}$, then

$$\Big(de(w_i; \mathcal{G}^{\mathcal{M}}_{\mathbf{A}}) \cup \{w_i\}\Big) \cap \{Z \mid P.Z \in \mathbf{W}\} \neq \emptyset. \tag{3.6}$$

Further, if there is a walk between two attribute classes, we can construct a path $\mathbf{p} := [p_1,\ p_2,\ ,\cdots,\ p_n]$ by iteratively removing any cycles in the walk by the first attribute class of the cycle.

Algorithm 3 walk-to-path

```
Initialize p with an empty sequence
i = 1
while i < |w| do
    Append w_i to p
    Assign i one plus the last index of w_i in w
```

Lemma 19. *Let* $\mathbf{w}$ *be the walk of attribute classes projected from a d-connection path in some ground graph. Let a path* $\mathbf{p}$ *be induced from* $\mathbf{w}$ *following the above Algorithm 3. Then,* $\mathbf{p}$ *satisfies that if* $p_{a-1} \rightarrow p_a \leftarrow p_{a+1}$ *in* $\mathcal{G}^{\mathcal{M}}_{\mathbf{A}}$*, then*

$$\Big(de(p_a; \mathcal{G}^{\mathcal{M}}_{\mathbf{A}}) \cup \{p_a\}\Big) \cap \{Z \mid P.Z \in \mathbf{W}\} \neq \emptyset. \tag{3.7}$$

where $\mathbf{W}$ *is a set of conditionals of the given relational independence query.*

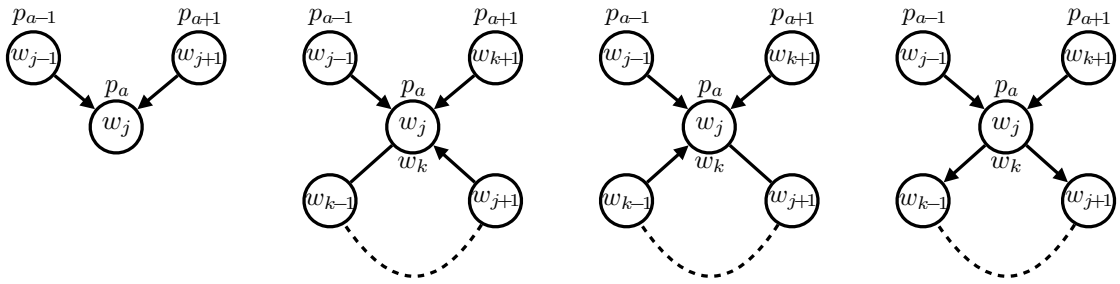


Figure 3.8: Diagrams for a walk and its induced path in $\mathcal{G}^{\mathcal{M}}_{\mathbf{A}}$

Proof. Let jth element of the walk correspond to the ath path element, that is, $p_a := w_j$. Observe that $w_{j-1} = p_{a-1}$. First, if w_j was not involved in any cycle in $\mathbf{w}$, then $w_{j+1} = p_{a+1}$. Hence, Equation 3.7 is satisfied due to $w_{j-1} \to w_j \leftarrow w_{j+1}$.

We now consider where w_j was the start (and end) of a cycle in the walk $\mathbf{w}$. Let the cycle starts from index j to index k, that is, $w_j = w_k$ and $j + 2 \leq k$. Observe that $w_{k+1} = p_{a+1}$ (see Figure 3.8). We examine following cases:

1. (Case $w_j \leftarrow w_{j+1}$) Then, $w_{j-1} \to w_j \leftarrow w_{j+1}$. Done.
2. (Case $w_{k-1} \leftarrow w_k$) Then, $w_{k-1} \to w_k \leftarrow w_{k+1}$. Done.
3. (Case otherwise) Given $w_j \to w_{j+1} \cdots w_{k-1} \leftarrow w_k$ (n.b. $j + 1$ can be $k - 1$), there must be at least one collider since the direction needs to be changed at least once to avoid a directed cycle. Let

$$\ell^* := \min_{\ell \in \{\ell | j < \ell < k \text{ and } w_{\ell-1} \to w_\ell \leftarrow w_{\ell+1}\}} \ell.$$

Then,

$$\Big(de(w_{\ell^*}; \mathcal{G}^{\mathcal{M}}_{\mathbf{A}}) \cup \{w_{\ell^*}\}\Big) \cap \{Z \mid P.Z \in \mathbf{W}\} \neq \emptyset.$$

We observe $de(w_{\ell^*}; \mathcal{G}^{\mathcal{M}}_{\mathbf{A}}) \subsetneq de(w_j; \mathcal{G}^{\mathcal{M}}_{\mathbf{A}})$ since $w_{\ell^*} \in de(w_j; \mathcal{G}^{\mathcal{M}}_{\mathbf{A}})$. Finally,

$$\Big(de(w_j; \mathcal{G}^{\mathcal{M}}_{\mathbf{A}}) \cup \{w_j\}\Big) \cap \{Z \mid P.Z \in \mathbf{W}\} \neq \emptyset.$$

□

Since the existence of such a path is a necessary condition, finding such a path does not tell whether relational d-connection actually holds true. However, the

non-existence of such a path demonstrates the non-existence of such a walk. Hence, this procedure soundly determines relational d-separation.

3.4.2 Randomization Approach

By the definition of relational d-separation, the existence of any relational skeleton σ and a base item $b \in \sigma(B)$ that satisfies

$$U|_b^\sigma \perp\!\!\!\perp V|_b^\sigma \mid \mathbf{W}_b^\sigma$$

suffices relational d-connection. One can trivially randomly construct a large relational skeleton σ which might contain a relational d-connection path in its ground graph $\mathcal{G}_\sigma^\mathcal{M}$ if exists.

In the case of marginal relational d-separation where an empty set is given, making a dense relational skeleton increases the chance of finding a d-connection path if exists. However, there is a trade-off between dense or sparse relational skeleton since conditionals can either 'help' finding a d-connection path through activating colliders or 'deter' through blocking a possible path. Further, due to one cardinality, we cannot just make a denser relational skeleton.

Considering how 'difficult' it is to randomly generate a relational d-connection path implies that how it is unlikely to detect relational conditional dependence from real-world relational data unless the underlying relational skeleton is systematically formed to create such a dependence. Hence, this randomized approach is a valuable means for sound relational d-connection.

3.4.3 Relational Variable-Based d-Separation

Consider testing $U \perp\!\!\!\perp V \mid \mathbf{W}$ where $\{U, V\} \cup \mathbf{W}$ are a set of relational variables of the same perspective $B \in \mathbf{I}$. We only consider cases where $\{U, V\} \cap \mathbf{W} = \emptyset$. Let $U = P.X$ and $V = Q.Y$. If $P.X$ and $Q.Y$ are either equal or intersectable, then, trivially $U \not\perp\!\!\!\perp V \mid \mathbf{W}$. Hence, it is proper to limit $P.X \neq Q.Y$ and $P.X$ and $Q.Y$ are not intersectable.

Assume there exists a *d-connection path* $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_m]$ in a ground graph $\mathcal{G}_\sigma^\mathcal{M}$. We denote all given item attributes by $\boldsymbol{\omega} := \bigcup_{W \in \mathbf{W}} W|_b^\sigma$. A set of relational paths from a base item b to the item of a given item attribute δ is denoted by

$\mathfrak{R}(\delta; \sigma, b) := \{R.Z \mid \delta \in R.Z|_b^\sigma\}$. The d-connection path $\boldsymbol{\delta}$ satisfies:

1. $\forall_{i \neq j} \delta_i \neq \delta_j$,
2. $\delta_i \to \delta_{i+1} \in \mathcal{E}\left(\mathcal{G}_\sigma^{\mathcal{M}}\right)$ or $\delta_{i+1} \to \delta_i \in \mathcal{E}\left(\mathcal{G}_\sigma^{\mathcal{M}}\right)$,
3. $\delta_1 \in P.X|_b^\sigma$,
4. $\delta_m \in Q.Y|_b^\sigma$,
5. If δ_i is a collider (i.e., $\delta_{i-1} \to \delta_i \leftarrow \delta_{i+1}$), then $\boldsymbol{\omega} \cap \left(de\left(\delta_i; \mathcal{G}_\sigma^{\mathcal{M}}\right) \cup \{\delta_i\}\right) \neq \emptyset$.
6. If δ_i is not a collider, then $\delta_i \notin \boldsymbol{\omega}$.

Now we introduce a *d-connection walk*. A d-connection walk $\boldsymbol{\eta} := [\eta_1, \cdots, \eta_\ell]$ satisfies all requirements for $\boldsymbol{\delta}$ except the non-uniqueness (1) and with (5) modified — if η_i is a collider (i.e., $\eta_{i-1} \to \eta_i \leftarrow \eta_{i+1}$), then $\eta_i \in \boldsymbol{\omega}$.

Proposition 20. *The existence of a d-connection path with a base item b and a relational skeleton σ implies the existence of a d-connection walk with the same base item and the same relational skeleton.*

Proof. Let $\boldsymbol{\delta}$ contain $[\cdots \delta_{i-1}, \delta_i, \delta_{i+1}, \cdots]$ where $\delta_{i-1} \to \delta_i \leftarrow \delta_{i+1}$ in $\mathcal{G}_\sigma^{\mathcal{M}}$. Then, there must be a given item attribute $\omega \in \boldsymbol{\omega}$ such that $\omega \in de\left(\delta_i; \mathcal{G}_\sigma^{\mathcal{M}}\right) \cup \{\delta_i\}$ where either $\omega = \delta_i$ or no other $\omega' \in \boldsymbol{\omega}$ exists which appears in a directed path from δ_i to ω in $\mathcal{G}_\sigma^{\mathcal{M}}$. We can explicitly represent the path from δ_i, a collider in the d-connection path, to the given item attribute ω in the d-connection path $\boldsymbol{\delta}$, e.g., $[\cdots \delta_{i-1}, \delta_i, \cdots, \omega, \cdots, \delta_i, \delta_{i+1}, \cdots]$. By replacing every collider, we obtain a valid d-connection walk. □

We move to introduce a sequence of relational variables covering a d-connection walk.

Proposition 21. *If there exist a d-connection walk $\eta = [\eta_1, \ldots, \eta_\ell]$, then there exists a sequence of relational variables $\phi = [\phi_1, \ldots, \phi_\ell]$ where $\eta_i \in \phi_i|_b^\sigma$.*

Proof. First, $\boldsymbol{\eta}$ is formed by traversing edges in $\mathcal{G}_\sigma^{\mathcal{M}}$ where an edge corresponds to a path of items in a relational skeleton while ignoring their attributes. Further, the existence of subsequent paths implies the existence of a path from item b to the item of η_i in σ. □

We further show that

Claim 22. If η_i is a collider, then, $\exists_{W \in \mathbf{W}} \phi_i \cap^* W$.

Proof. Negation of $\exists_{W \in \mathbf{W}} \phi_i \cap^* W$ implies that $\boldsymbol{\omega} \cap \phi_i|_b^\sigma = \emptyset$, which contradicts $\{\eta_i\} \subseteq \boldsymbol{\omega} \cap \phi_i|_b^\sigma$ since $\eta_i \in \boldsymbol{\omega}$ and $\eta_i \in \phi_i|_b^\sigma$. □

Claim 23. $\phi_m \cap^* Q.Y$.

Proof. Similarly, $\neg\left(\phi_m \cap^* Q.Y\right)$ contradicts $\eta_m \in Q.Y|_b^\sigma$ and $\eta_m \in \phi|_b^\sigma$. □

Claim 24. If η_i is not a collider, then $\phi_i \notin \mathbf{W}$.

Proof. Suppose for the sake of contradiction, $\phi_i \in \mathbf{W}$. Then $\eta_i \in \phi_i|_b^\sigma \subseteq \boldsymbol{\omega}$. This violates $\eta_i \notin \boldsymbol{\omega}$. □

We investigate a new method to guarantee relational d-separation by exploring the space of relational variables. The method, which will be described in detail below, examines whether there exists a particular sequence of relational variables as described above. Then, the absence of such sequence implies that *no* d-connection walk exists in *any* relational skeleton and, hence, the relational d-separation, $P.X \perp\!\!\!\perp Q.Y \mid \mathbf{W}$, holds true. The method relies on newextend (see Algorithm 1), which we introduced for the purpose of identifying *parents* (and *children*) of non-canonical relational variables.

Lemma 25. *Given a relational schema $\mathcal{S}$, let $P, Q \in \mathbf{P}_\mathcal{S}$ where the terminal item class of P is the base item class of Q. Given a relational skeleton $\sigma \in \Sigma_\mathcal{S}$ and a base item $b \in \sigma(P_1)$, if $\delta \in P|_b^\sigma$ and $\delta' \in Q|_\delta^\sigma$, then*

$$\exists_{S \in \mathsf{newextend}(P,R)} \delta' \in S|_b^\sigma$$

Proof. It follows from the definition of newextend. □

We present an algorithm (see Algorithm 4) which explores the space of relational variables by applying Lemma 25 to answer whether there exists a sequence of relational variables as mentioned in Proposition 21. We denote the relational path and the attribute class of a relational variable by $\mathsf{rpath}(\cdot)$ and $\mathsf{attr}(\cdot)$, respectively.

Lemma 26. *The algorithm correctly identifies relational d-separation.*

Algorithm 4 Relational variable-based d-separation

```
1: procedure RV-D-SEP(U, V, W, M, limit)
2:   if U = V or U ∩* V then
3:     return False
4:   queue := [(→, U), (←, U)]
5:   while the number of loop is less than limit do
6:     if queue is not empty then
7:       pop (β, S) from queue
8:       if (β =→ and ∃_{W∈W} S ∩* W) or (β =← and S ∉ W) then
9:         for R.Z → V_{attr(S)} ∈ D do
10:          for T ∈ newextend(rpath(S), R) do
11:            if T.Z ∩* Q.Y then
12:              return False
13:            else
14:              Add (←, T.Z) to queue if it has not been queued.
15:      else if S ∉ W then
16:        for R.attr(S) → V_Z ∈ D do
17:          for T ∈ newextend(rpath(S), R̃) do
18:            if T.Z ∩* Q.Y then
19:              return False
20:            else
21:              Add (→, T.Z) to queue if it has not been queued.
22:    else
23:      return True
24:  return None
```

Proof. The algorithm explores whether there exists a sequence of relational variables as defined in Proposition 21. The termination of the algorithm returning False does not imply the existence of a relational d-connection path since the algorithm relies on a sufficient condition for relational d-connection. However, having an empty queue means search for the sequence is exhaustively performed, that is, there exists no relational d-connection path. □

This algorithm is related to the revised abstract ground graph. As its search procedure resembles traversing RVs with RVEs in an AGG. However, this does not require explicit construction of RVEs, IVs, nor IVEs. Intersectability plays a role only for $Q.Y$ and $\mathbf{W}$. This algorithm does not take co-intersectability into account. Although the soundness of the algorithm is irrelevant to the use of co-intersectability, if it is applied to the algorithm, then a number of wrong relational d-connection

results can be avoided. One can incorporate co-intersectability of $\langle \mathsf{rpath}\,(S)\,, R, T, Q\rangle$ at Line 12 or $\left\langle \mathsf{rpath}\,(S)\,, \tilde{R}, T, Q\right\rangle$ at Line 19. However, considering co-intersectability for line 8 will require additional information about how S is returned from newextend.

3.4.4 Constructive Approach

Let there exist a d-connection path $\boldsymbol{\delta}$ for an RCI query $P.X \perp\!\!\!\perp Q.Y \mid \mathbf{W}$. Imagine a $\boldsymbol{\delta}$-embedding *minimal* relational skeleton from which the d-connection path $\boldsymbol{\delta}$ becomes invalid if any of items are removed. A minimal relational skeleton includes only (i) items corresponding to relational paths representing edges in a ground graph, (ii) items corresponding to P and Q from b, and (iii) items from b corresponding to item classes in $W \in \mathbf{W}$ for every collider. Hence, we can build a minimal relational skeleton, which corresponds to some d-connection path if exists, as in Algorithm 5.

It first starts by considering all possible minimal relational skeletons containing items for P and Q from a base item b. It stores a queue of 'states' where a state is an intermediate result — a pair of an intermediate d-connection path and an intermediate relational skeleton (Line 1–8). For every queued state, we examine whether 'the last item attribute' can be a collider. If so, items corresponding to some $W \in \mathbf{W}$ is attached to make $\delta_{-1} \in W|_b^\sigma$ (Line 13–15).[4] Line 16–18 treats a current d-connection path as a d-connection walk and translate it back to a d-connection path. This resolves an issue that a given item attribute is not known in advance. Lines 19–23 investigates relational dependencies that can extend the current d-connection path (either $\delta_{-1} \leftarrow j.A$ or $\delta_{-1} \rightarrow j.A$). Any changes in a relational skeleton can affect its associated d-connection path, for example, some of added items open a new path from b to an existing item attribute so that an unblocked item attribute can be blocked. Hence, Line 27 checks the validity of a state. Finally, if the newly made d-connection path connects $\delta'_1 \in P.X|_b^{\sigma'}$ to $\delta'_{-1} \in Q.Y|_b^{\sigma'}$, then U and V are relationally d-connected by $\mathbf{W}$. Otherwise, a new state is added to the queue. The algorithm is not clever enough to tell when to stop extending a current state. The algorithm returns undetermined (i.e., None) when the queue is not empty but a user-set stopping criterion is met (this can be time limit, the maximum number of loops, or the length of d-connection path, etc). One can generalize the algorithm by equipping two queues for $U \perp\!\!\!\perp V \mid \mathbf{W}$ and

[4] A subscript -1 represents the last element of a given sequence.

Algorithm 5 Constructive relational d-separation

procedure CONSTRUCTIVE-REL-D-SEP($U = P.X$, $V = Q.Y$, $\mathbf{W}$, $\mathcal{M} = \langle \mathcal{S}, \mathbf{D} \rangle$, limit)
  Preconditions: $P.X \neq Q.Y$ and $P.X$ and $Q.Y$ are not intersectable, $\{U, V\} \cap \mathbf{W} = \emptyset$
  add a base item b to an empty relational skeleton σ.
  add items for P to σ from b
  $\{\delta_1\} := P.X|_b^\sigma$.
  $\boldsymbol{\sigma}'$:= all relational skeletons by attaching items for Q to σ from b in all valid ways.
  $\boldsymbol{\delta} := [\delta_1]$
  Add $(\boldsymbol{\delta}, \sigma')$ to queue for every $\sigma' \in \boldsymbol{\sigma}'$
  while the number of loop is less than limit and queue is not empty **do**
    $(\boldsymbol{\delta}, \sigma)$:=pop from queue
    $i.Z := \delta_{-1}$
    given := $\delta_{-1} \in \bigcup_{W \in \mathbf{W}} W|_b^\sigma$, colliderable := given **or** $de(\delta_{-1}; \mathcal{G}_\sigma^\mathcal{M}) \cap \bigcup_{W \in \mathbf{W}} W|_b^\sigma \neq \emptyset$
    if $|\boldsymbol{\delta}| > 1$ **and** $\delta_{-2} \to \delta_{-1} \in \mathcal{G}_\sigma^\mathcal{M}$ **and not** given **then**
      for σ' created by attaching items for $W \in \mathbf{W}$ such that $\mathsf{attr}(W) = Z$ to σ from b to i in all valid ways **do**
        append $(\boldsymbol{\delta}, \sigma')$ to queue.
    if $|\boldsymbol{\delta}| > 1$ **and** $\delta_{-2} \to \delta_{-1} \in \mathcal{G}_\sigma^\mathcal{M}$ **and** given **then**
      for δ_j **such that** $\delta_j \in de(\delta_{-1}; \mathcal{G}_\sigma^\mathcal{M}) \cup \delta_{-1}$ **and** $\delta_{j-1} \to \delta_j \in \mathcal{G}_\sigma^\mathcal{M}$ **do**
        append $(\boldsymbol{\delta}_{1:j}, \sigma)$ to queue with 'collider-only' property.
    $\mathbf{T} := \emptyset$
    if not given **and not** collider-only **then**
      $\mathbf{T} := \mathbf{T} \cup \{\tilde{R}.A \mid R.Z \to V_A \in \mathbf{D}\}$
    if $|\boldsymbol{\delta}| = 1$ **or** $\delta_{-2} \leftarrow \delta_{-1} \in \mathcal{G}_\sigma^\mathcal{M}$ **and not** given **or** $\delta_{-2} \to \delta_{-1} \in \mathcal{G}_\sigma^\mathcal{M}$ **and** colliderable **then**
      $\mathbf{T} := \mathbf{T} \cup \{R.A \mid R.A \to V_Z \in \mathbf{D}\}$
    for $R.A$ **in T do**
      for σ' created by attaching items for R to σ from i in all valid ways **do**
        Let $\boldsymbol{\delta}'$ be $\boldsymbol{\delta}$ with $j.A$ appended where j is the last item for R.
        if $\boldsymbol{\delta}'$ is a valid d-connection path in $\mathcal{G}_{\sigma'}^\mathcal{M}$ from δ_1' to δ_{-1}' **then**
          if $j.A \in Q.Y|_b^\sigma$ **then**
            return False
          else
            append $(\boldsymbol{\delta}', \sigma')$ to queue
  return True if queue is empty else None

$V \perp\!\!\!\perp U \mid \mathbf{W}$, respectively. Then, two queues can be alternatively explored over the loop to check either relational d-separation or relational d-connection, whichever is figured out first.

3.4.4.1 Thoughts on the Completeness of Constructive Approach

We aim to characterize conditions under wchich we can stop constructing a relational d-connection path and claim that U and V are relationally d-separated by $\mathbf{W}$. One condition is related to the number of colliders in a d-connection path. If we focus on the sequence of attribute classes corresponding to its d-connection walk (where a collider needs to be a given item attribute) in a ground graph, attribute classes must move upward or downward (in a CDG), and changing direction from downward to upward is only possible through a given collider. Hence, limiting the number of colliders in a d-connection walk can be used as a stopping criterion. However, the failure to find a d-connection walk based on such a stopping criterion does not imply that U and V are, in fact, relationally d-separated by $\mathbf{W}$.

There is one easy case for relational d-separation: if $\mathbf{W} = \emptyset$, or $\mathsf{card}\,(W) = \mathsf{one}$ for every $W \in \mathbf{W}$. There are a finite number of given item attributes, and visiting any of them more than once is redundant since one can shorten such a d-connection walk by removing elements between two repeated colliders. Hence, the constructive approach is *complete* when $\mathbf{W} = \emptyset$, or $\mathsf{card}\,(W) = \mathsf{one}$ for every $W \in \mathbf{W}$ while allowing at most $|\mathbf{W}|$ colliders.

To check whether an action (extending a d-connection path by a relational dependency or attaching $W \in \mathbf{W}$ for the purpose of making the last item attribute a collider) is redundant, it is essential to examine previous states whether the action results in a state which enables us to do other actions that were not available in any of the previous states.

3.4.5 Empirical Study with Company Example

We tested proposed relational d-separation methods with the company example (Maier, 2014). We only considered relational variables with at most of 4-hop length. There are total 3141 RCI queries of the form $U \perp\!\!\!\perp V \mid \mathbf{W}$ where (i) U and V are not intersectable, (ii) $\{U, V\} \cap \mathbf{W} = \emptyset$. We implemented randomized relational skeleton algorithm so that (i) there are 100 relational skeletons; (ii) each contains 200 entities for each entity class and 100 relationships for each relationship class; and (iii) every relationship randomly chooses participating entities randomly. There are more entities than relationships (per item class) because having a larger number of entities prevents every entity from being exhaustively connected to some

relationship (n.b. there can be unfunded products although it is often not the case in real environments.)

The constructive approach determined all relational d-connection and -separation. There are 199 queries determined as relational d-separation, and 2942 as relational d-connection. Since all attribute classes are connected in $\mathcal{G}_{\mathbf{A}}^{\mathcal{M}}$, CDG-based approach always returned undetermined. Randomized approach yielded 83.85% of recall for relationally d-connected queries, although one can improve the recall by simply use more (larger) relational skeletons. Relational variable-based approach identified 91.96% (183/199) of relational d-separation queries. It is shown to be very effective with the company example. The abstract ground graph-based relational d-separation (which is not sound) results 100% of recall for relational d-connection. However, the method only correctly answers 20.60% of relational d-separation queries. The AGG-based method is worse than the newly proposed relational variable-based approach, which does not require pre-construction of a large graph. We conjecture that the revised abstract ground graph can be turned into a sound relational d-separation method by changing relational d-connection output as undetermined. We confirmed (in the company example) that every relational d-separation query identified by AGG-based method is also correctly identified by the relational variable-based method.

3.5 Concluding Remarks

Relational d-separation generalizes d-separation to a relational setting. Maier et al. (2013b) explored the design of abstract ground graphs in answering relational d-separation through applying traditional d-separation in the lifted representation. Relational Causal Discovery (RCD), a causal discovery algorithm for relational causal model, was proposed by Maier et al. (2013a) with an abstract ground graph as a primary representation. However, Lee and Honavar (2015) identified several problems of abstract ground graphs, and Lee and Honavar (2016b) further figured out the non-completeness of RCD due to the problems of abstract ground graphs. We showed the original definition of RVEs (based on extend), IVs, and IVEs are all problematic. We revised (i) RVEs with newextend, (ii) IVs with a correct sufficient and necessary condition for *intersectability*, and (iii) IVEs with *co-intersectability*. Unfortunately, even the revised abstract ground graph does not provide a means

for relational d-separation. Whether one can devise a new method on top of revised AGG under path semantics is still an open question.

We provide four sound relational d-separation methods: (i) a class dependency graph-based approach investigates a necessary condition for relational d-connection at an attribute class level; (ii) a randomized approach provides a practical means to assess relational d-connection; (iii) a relational variable-based approach also considers a necessary condition for relational d-connection in the space of relational variables; and (iv) a constructive approach enumerates concrete relational skeletons so that it can return a d-connection path if exists. The problem of a sound and complete characterization of relational d-separation still remains open. We view solving this problem as an important direction for future research.

Chapter 4
Characterization of Markov Equivalence Class of RCM

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*. It is an essential step in specifying a provably complete constraint-based algorithm for learning the structure of an RCM under *path semantics*, a more elegant alternative to bridge-burning semantics. 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 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. This chapter is based on the results from (Lee and Honavar, 2016a).

4.1 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.

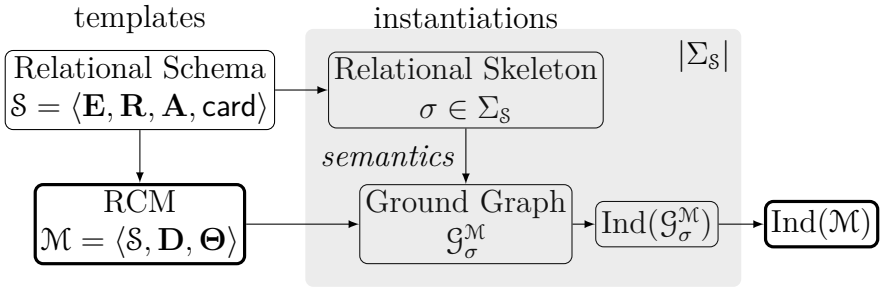


Figure 4.1: Schematic showing a relational schema, an RCM, and their respective instantiations, i.e., relational skeleton, and ground graphs and the independence relations of the RCM entailed from the independence relations admitted by the ground graphs.

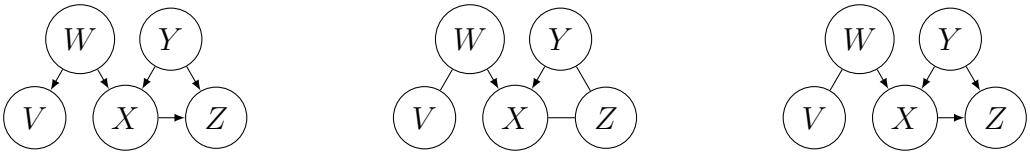
Definition 27 (Markov Equivalence of RCMs)**.** Two RCMs are *Markov equivalent* if they entail the same set of relational d-separation conditions.

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, 2016b). Hence, we proceed to characterize the Markov equivalence of RCMs under path semantics.

Recall that the relational d-separation $U \perp\!\!\!\perp V \mid \mathbf{W}$ in an RCM is equivalent to $U|_i^\sigma \perp\!\!\!\perp V|_i^\sigma \mid \mathbf{W}|_i^\sigma$ for every base item i in *every* ground graph σ 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:

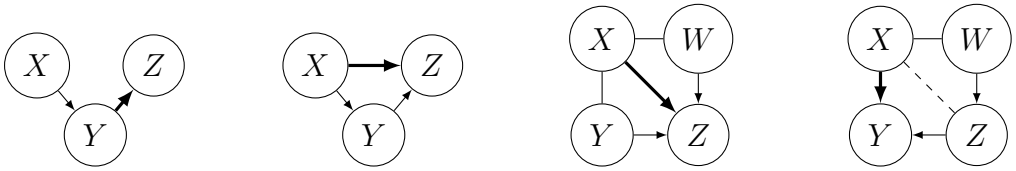
$$\forall_{\sigma \in \Sigma_{\mathcal{S}}} [\mathcal{G}_\sigma^{\mathcal{M}}] = [\mathcal{G}_\sigma^{\mathcal{M}'}] \;\Rightarrow\; [\mathcal{M}] = [\mathcal{M}'] \tag{4.1}$$

where $[\mathcal{M}]$ and $[\mathcal{G}]$ denote the Markov equivalence class of an RCM and a DAG $\mathcal{G}$, respectively. In Section 4.1.1, we will demonstrate that the converse of Equation 4.1 holds as well, thereby establishing a necessary and sufficient condition for two RCMs to be Markov equivalent.



(a) (b) (c)

Figure 4.2: An example of a DAG, its pattern, and its CPDAG



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

Figure 4.3: Meek's rules

Markov equivalence of DAG

First, we recall the characterization of Markov equivalence of DAG (Verma and Pearl, 1990; Andersson et al., 1997). See Figure 4.2 for an example of a DAG, its pattern, and its CPDAG where $\langle W, X, Y \rangle$ is an unshielded collider and $\langle V, W, X \rangle$ and $\langle W, X, Z \rangle$ are unshielded non-colliders.

Let $\mathcal{G}$ be a DAG with random variables $\mathbf{V}$ as vertices. Let X, Y, and Z be in $\mathbf{V}$. A triple $\langle X, Y, Z \rangle$ 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 $X \to Y \leftarrow Z$ 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 $X \to Y$ implies that every DAG in the class shares the edge $X \to Y$ (*compelled edge*) while an undirected 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). See Figure 4.3 – 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. 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.

4.1.1 Pattern of 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 $\mathcal{G}^{\mathcal{M}}_{\sigma}$ of an RCM $\mathcal{M}$ for an arbitrary relational skeleton $\sigma \in \Sigma_{\mathcal{S}}$. Then, $\langle i.X, j.Y, k.Z\rangle$ is an unshielded triple in $\mathcal{G}^{\mathcal{M}}_{\sigma}$ only if $P.Y \in adj\,(V_X; \mathcal{M})$ and $Q.Z \in adj\,(V_Y; \mathcal{M})$ where $j \in P|^{\sigma}_{i}$ and $k \in Q|^{\sigma}_{j}$ for $i.X$ and $k.Z$ to be connected to $j.Y$. Furthermore, $R.Z$ must not be in $adj\,(V_X; \mathcal{M})$ for every path R such that $k \in R|^{\sigma}_{i}$ for $i.X$ and $k.Z$ to be disconnected in $\mathcal{G}^{\mathcal{M}}_{\sigma}$. Then, we define a *canonical unshielded triple* as follows:

Definition 28 (Canonical Unshielded Triple)**.** Let $\mathcal{M}$ be an RCM defined on a relational schema $\mathcal{S}$. Suppose $\langle i.X, j.Y, k.Z\rangle$ is an unshielded triple (UT) in the ground graph $\mathcal{G}^{\mathcal{M}}_{\sigma}$ for some $\sigma \in \Sigma_{\mathcal{S}}$. There must be two (not necessarily distinct) dependencies $P.Y - V_X$ and $Q.Z - V_Y$ of $\mathcal{M}$ (ignoring directions) such that $j \in P|^{\sigma}_{i}$ and $k \in Q|^{\sigma}_{j}$. Then, we say that $\langle V_X, \mathbf{P}.Y, R.Z\rangle$ is a *canonical unshielded triple* (CUT) of $\mathcal{M}$ for every $R \in \{T \mid k \in T|^{\sigma}_{i}\}$ where $\mathbf{P} = \{T \mid j \in T|^{\sigma}_{i}\}$.

Since whenever $\langle i.X, j.Y, k.Z\rangle$ is an unshielded triple in $\mathcal{G}^{\mathcal{M}}_{\sigma}$, so is $(k.Z, j.Y, i.X)$, it follows that whenever $\langle V_X, \mathbf{P}.Y, R.Z\rangle$ is a CUT of $\mathcal{M}$, there exists a CUT $\langle V_Z, \mathbf{Q}.Y, \tilde{R}.X\rangle$ for some relational paths $\mathbf{Q}$.

Theorem 29. *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 $P.Y \rightarrow V_X$ in $\mathcal{M}$ while both $P.Y \rightarrow V_X$ and $\tilde{P}.X \rightarrow V_Y$ are not in $\mathcal{M}'$ (or vice versa). Then, either $P.Y \perp\!\!\!\perp V_X \mid pa(V_X; \mathcal{M}')$ or $\tilde{P}.X \perp\!\!\!\perp V_Y \mid pa(V_Y; \mathcal{M}')$ 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 $i.X \rightarrow j.Y$ in $\mathcal{G}_\sigma^{\mathcal{M}}$ where $\{P\} = \{T \mid i \in T|_j^\sigma\}$ while $P.Y \notin pa(V_X; \mathcal{M}')$ and $\tilde{P}.X \notin pa(V_Y; \mathcal{M}')$.

Case (ii): There must exist a CUT $\langle V_X, \mathbf{P}.Y, R.Z \rangle$ corresponding to an unshielded triple $\langle i.X, j.Y, k.Z \rangle$, 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 $R.Z \notin adj\,(V_X; \mathcal{M})$ for every $R \in \{T \mid k \in T|_i^\sigma\}$, there must exist a separating set $\mathbf{S} \subseteq adj(V_X; \mathcal{M})$ such that $V_X \perp\!\!\!\perp R.Z \mid \mathbf{S}$ in $\mathcal{M}$ assuming $X \not\prec_\pi Z$ without loss of generality.[1] By the definition of relational d-separation, $\mathbf{S}$ must be disjoint with $\mathbf{P}.Y$. However, in $\mathcal{M}'$, $V_X \not\perp\!\!\!\perp R.Z \mid \mathbf{S}$ 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.

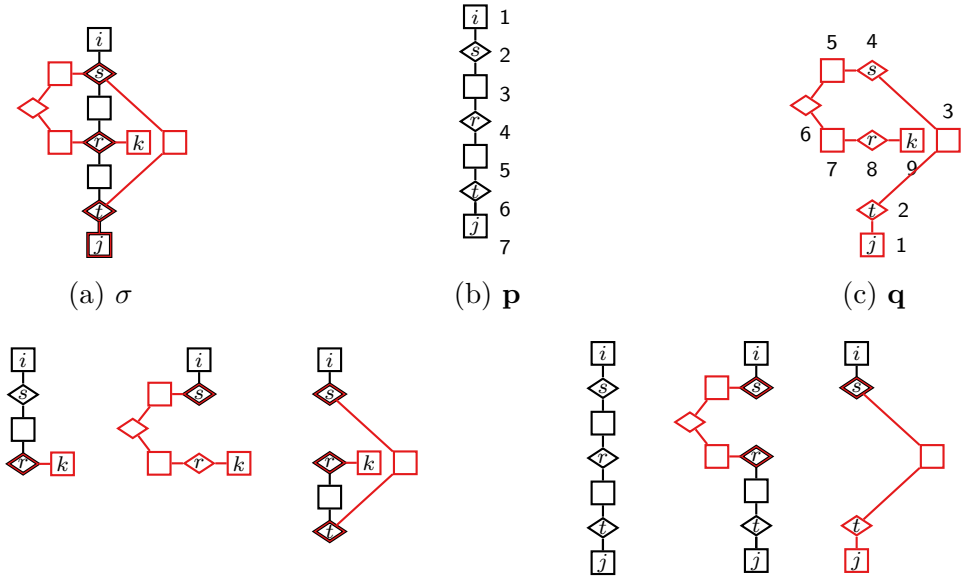
Definition 30 (Pattern of RCM)**.** Let $\mathcal{M} = \langle \mathcal{S}, \mathbf{D} \rangle$ be an RCM and $\mathfrak{C}^{\mathcal{M}}$ be all canonical unshielded colliders of $\mathcal{M}$. We define the set of *attribute class level colliders* as

$$\mathfrak{C}_{\mathbf{A}}^{\mathcal{M}} := \{\langle X, Y, Z \rangle \mid \langle V_X, \mathbf{P}.Y, R.Z \rangle \in \mathfrak{C}^{\mathcal{M}}\}.$$

Then, the *pattern* of $\mathcal{M}$, *pattern* $(\mathcal{M})$, is a partially-directed RCM $\langle \mathcal{S}, \mathbf{D}' \cup \mathbf{D}'' \rangle$ where

$$\mathbf{D}' := \{Q.X \rightarrow V_Y \in \mathbf{D} \mid \langle X, Y, Z \rangle \in \mathfrak{C}_{\mathbf{A}}^{\mathcal{M}}\}$$

[1] Otherwise, the proof can be obtained using $\langle V_Z, \tilde{\mathbf{Q}}.Y, \tilde{R}.X \rangle$.



(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 4.4: Illustration of key concepts used to characterize CUTs for a hypothetical RCM $\mathcal{M}$ with $P.Y \in adj\,(V_X; \mathcal{M})$ and $Q.Z \in adj\,(V_Y; \mathcal{M})$ yielding an unshielded triple $\langle i.X, j.Y, k.Z \rangle$ in $\mathcal{G}_\sigma^{\mathcal{M}}$ where P and Q correspond to item classes of $\mathbf{p}$ and $\mathbf{q}$, respectively.

and

$$\mathbf{D}'' := \{P.Y - V_X \mid P.Y \to V_X \in \mathbf{D} \setminus \mathbf{D}'\}.$$

Lemma 31. $[\mathcal{M}] = [\mathcal{M}'] \Leftrightarrow pattern(\mathcal{M}) = pattern(\mathcal{M}')$.

Proof. The proof follows from Theorem 29. □

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.

4.1.1.1 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 unshielded triples in the corresponding ground graphs, and the corresponding

CUTs. Because such an approach is not computationally tractable, we consider the following alternative: enumerate the relational skeletons that are just large enough to include an unshielded triple in the corresponding ground graph, and the corresponding CUT. We first investigate conditions under which a relational skeleton includes an unshielded triple in the corresponding ground graph, and then provide a characterization of CUTs in terms of such unshielded triples, which leads to an efficient CUT enumeration algorithm whose time complexity is polynomial in the number of dependencies, $|\mathbf{D}|$, and $\max\{|P| \mid P.X \to V_Y \in \mathbf{D}\}$ (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 $\langle i.X, j.Y, k.Z\rangle$ in some relational skeleton $\sigma \in \Sigma_{\mathcal{S}}$. Let $P.Y \in adj\,(V_X; \mathcal{M})$ and $Q.Z \in adj\,(V_Y; \mathcal{M})$ (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 $\mathbf{q}$:

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

See Figure 4.4 for an example, illustrating key concepts used to characterize CUTs for a hypothetical RCM $\mathcal{M}$ with $P.Y \in adj\,(V_X; \mathcal{M})$ and $Q.Z \in adj\,(V_Y; \mathcal{M})$ yielding an unshielded triple $\langle i.X, j.Y, k.Z\rangle$ in $\mathcal{G}_\sigma^{\mathcal{M}}$ where P and Q correspond to item classes of $\mathbf{p}$ and $\mathbf{q}$, respectively. In this example, $\mathbf{J}_{\mathbf{p},\mathbf{q}} = \{(2, 4), (4, 8), (6, 2), (7, 1)\}$ in Figure 4.4(e). 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:

$$(a_r, b_r) := \underset{(a,b)\in\mathbf{J}_{\mathbf{p},\mathbf{q}}}{\arg\max}\, b.$$

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

$$R_r := P^{:a_r} \oplus Q^{b_r:}$$

where the symbol "$\oplus$" is a path concatenation operator (e.g., $[E_1, R_1, E_2] \oplus [E_2, R_2] = [E_1, R_1, E_2, R_2]$). We can infer that $R_r.Z \notin adj\,(V_X; \mathcal{M})$ since $i.X$ and $k.Z$ are disconnected. Next, we define an anchor for the first shared item of $\mathbf{p}$:

$$(a_s, b_s) := \underset{(a,b)\in\mathbf{J}_{\mathbf{p},\mathbf{q}}}{\arg\min}\, a.$$

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

$$R_s := P^{:a_s} \oplus Q^{b_s:},$$

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, $\mathbf{p}_{:a_s} \oplus \mathbf{q}_{b:b_s:-1} \oplus \mathbf{p}_{a_r:a:-1} \oplus \mathbf{q}_{b_r:}$, where the subpath with ": -1" represents the reverse of the subpath. There do exist such anchors:

$$(a_t, b_t) := \underset{(a,b)\in\mathbf{J}_{\mathbf{p},\mathbf{q}}, a_r<a, b<b_s}{\arg\max} b,$$

and we likewise define

$$R_t := P^{:a_s} \oplus Q^{b_t:b_s:-1} \oplus P^{a_r:a_t:-1} \oplus Q^{b_r:}.$$

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 4.4(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 an unshielded triple. First, since all triples characterized by a given characteristic anchor share common relational path(s) from i to k, the existence of a dependency $R_r.Z - V_X$ (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 $adj(V_X; \mathcal{M}) \cap \{R_r, R_s, R_t\}.Z$ 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 Algorithm 6 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 an unshielded triple. Then, it outputs a CUT $\langle V_X, \{P.Y, (P^{:a_r} \oplus Q^{:b_r:-1}).Y\}, R_r.Z\rangle$ where the (virtually constructed) relational skeleton σ' satisfies $\{R_r\} = \{T \mid k \in T|_i^{\sigma'}\}$ and $\{P, P^{:a_r} \oplus Q^{:b_r:-1}\} = \{T \mid j \in T|_i^{\sigma'}\}$. If (a_r, b_r) must differ from (a_s, b_s), then the

Algorithm 6 An algorithm to return a canonical unshielded triple of an RCM given a pair of (undirected) dependencies. R_r, R_s, R_t, and $\mathbf{P}$ are as defined in the main text.

procedure GET_ONE_CUT($P.Y - V_X$, $Q.Z - V_Y$, $\mathcal{M}$)
  $\mathsf{LL} := \mathsf{LLRSP}$
  $m := |P|$, $\ell := \mathsf{LL}(\tilde{P}, Q)$, $\mathbf{J} := \left\{(a,b) \mid P^a = Q^b,\ 1 \le a \le m - \ell + 1,\ \ell \le b \le |Q|\right\}$
  for (a_r, b_r) **in J such that** $\mathsf{LL}(P^{:a_r:-1}, Q^{b_r:}) = \mathsf{LL}(P^{a_r:}, Q^{b_r:}) = 1$ **and** $R_r.Z \notin adj(V_X; \mathcal{M})$ **do**
    $\ell_\alpha := \mathsf{LL}(Q^{\ell:b_r:-1}, P^{:a_r:-1})$
    if $\ell_\alpha = 1$ **then**
      if $P^{a_r:m-\ell+1} \cap^* Q^{\ell:b_r:-1}$ **then**
        return $\langle V_X,\ \{P.Y, (P^{:a_r} \oplus Q^{:b_r:-1}).Y\},\ R_r.Z\rangle$
    else if $\ell_\alpha < b_r - \ell + 1$ **and** $a_r < m - \ell + 1$ **and** $\ell < b_r$ **then**
      for (a_s, b_s) **in** $\{(a,b) \in \mathbf{J} \mid a \le a_r - \ell_\alpha + 1,\ \ell < b \le b_r - \ell_\alpha + 1\}$ **such that** $R_s.Z \notin adj(V_X; \mathcal{M})$ **do**
        $P_A,\ P_B,\ Q_A,\ Q_B := P^{:a_s:-1},\ P^{a_s:a_r-\ell_\alpha+1},\ Q^{b_s:b_r-\ell_\alpha+1},\ Q^{\ell:b_s:-1}$
        if $\mathsf{LL}(P_A, Q_A) > 1$ **or** $\mathsf{LL}(P_A, Q_B) > 1$ **or not** $P_B \cap^* Q_A$ **or** $1 < \mathsf{LL}(P_B, Q_B) = \min(|P_B|, |Q_B|))$ **then**
          continue
        for (a_t, b_t) **in** $\{(a,b) \in \mathbf{J} \mid a_r < a \le m - \ell + 1,\ \ell \le b < b_s - \mathsf{LL}(P_B, Q_B) + 1\}$ **such that** $R_t.Z \notin adj(V_X; \mathcal{M})$ **do**
          $P_C,\ P_D,\ Q_C,\ Q_D = P^{a_r:a_t:-1},\ P^{a_t:m-\ell+1},\ Q^{b_t:b_s-\mathsf{LL}(P_B,Q_B)+1},\ Q^{\ell:b_t:-1}$
          if $\mathsf{LL}(P_C, Q_C) > 1$ **or** $\mathsf{LL}(P_D, Q_C) > 1$ **then**
            continue
          if $\mathsf{LL}(P_C, Q_D) = 1$ **and** $P_D \cap^* Q_D$ **then**
            return any of $\langle V_X,\ \mathbf{P}.Y,\ R_r.Z\rangle$, $\langle V_X,\ \mathbf{P}.Y,\ R_s.Z\rangle$, $\langle V_X,\ \mathbf{P}.Y,\ R_t.Z\rangle$
          else if $1 < \mathsf{LL}(P_C, Q_D) < \min(|P_C|, |Q_D|)$ **and** $m - \ell + 1 < a_t$ **and** $\ell < b_t$ **then**
            return any of $\langle V_X,\ \mathbf{P}.Y,\ R_r.Z\rangle$, $\langle V_X,\ \mathbf{P}.Y,\ R_s.Z\rangle$, $\langle V_X,\ \mathbf{P}.Y,\ R_t.Z\rangle$
  return None

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: $\langle V_X, \mathbf{P}.Y, R_r.Z\rangle$, $\langle V_X, \mathbf{P}.Y, R_s.Z\rangle$, and $\langle V_X, \mathbf{P}.Y, R_t.Z\rangle$ 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:

1. P,
2. $P^{:a_w} \oplus Q^{:b_w:-1}$,
3. $P^{:a_s} \oplus Q^{:b_s:-1}$,
4. $P^{:a_s} \oplus Q^{b_t:b_s:-1} \oplus P^{a_t:}$,

5. $P^{:a_s} \oplus Q^{b_s:b_r} \oplus P^{a_r:}$, and

6. $P^{:a_s} \oplus Q^{b_s:b_r} \oplus P^{a_r:a_w} \oplus Q^{:b_w:-1}$

with $a_w := a_t - \gamma + 1$ and $b_w := b_t - \gamma + 1$ where $\gamma := \mathsf{LLRSP}(P^{a_r:a_t:-1}, Q^{:b_t:-1})$.

For example, paths of items in Figure 4.4(e) correspond to three distinct relational paths of $\mathbf{P}$.

4.1.2 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 $\mathcal{G}_{\mathbf{A}}$ (initialized with $\mathcal{G}_{\mathbf{A}}^{pattern(\mathcal{M})}$) with a set of attribute class level non-colliders, denoted by $\mathfrak{N}_{\mathbf{A}}^{\mathcal{M}}$ ($\mathfrak{N}$ 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 $\langle V_X, \mathbf{P}.Y, R.Z \rangle$, corresponding attribute class level non-collider is $\langle X, Y, Z \rangle$. It is the case that $X = Z$, that is, $\langle X, Y, X \rangle \in \mathfrak{N}$. 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 $\mathcal{G}_{\mathbf{A}}$ that can be oriented using RBO have been oriented, and we exclude them (e.g., $\langle X, Y, X \rangle$) from $\mathfrak{N}$. Otherwise if $X \neq Z$, then X and Z may be connected making $\langle X, Y, Z \rangle$ *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 $\mathcal{G}_{\mathbf{A}}^{pattern(\mathcal{M})}$ and $\mathfrak{N}_{\mathbf{A}}^{\mathcal{M}}$:

Proposition 32. *Let $\langle X, Y, Z \rangle$ be an unshielded collider in $\mathcal{G}_{\mathbf{A}}^{\mathcal{M}}$, then $X \to Y \leftarrow Z$ in $\mathcal{G}_{\mathbf{A}}^{pattern(\mathcal{M})}$.*

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 33. *For every unshielded non-collider* $\langle X, Y, Z\rangle \in \mathcal{G}_{\mathbf{A}}^{\mathcal{M}}$, $\langle X, Y, Z\rangle \in \mathfrak{N}_{\mathbf{A}}^{\mathcal{M}}$.

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

4.1.2.1 Sound Rules

The four rules in Figure 4.3 can be used to correctly orient the edges in a partially-directed CDG $\mathcal{G}_{\mathbf{A}}$ (Corollary 33). We provide three additional rules that make use of $\mathfrak{N}$. First, if $\langle X, Y, Z\rangle \in \mathfrak{N}$ and $X \to Y$, then $Y \to Z$. This can be viewed as a generalization of R1 that avoids checking unshieldedness. Second, if $\langle X, Y, Z\rangle \in \mathfrak{N}$ and $X \to Z$, then $Y \to Z$. This is similar to R4 in the sense that $Y \to Z$ 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.

4.1.2.2 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

Algorithm 7 Completing a PDAG given non-colliders

procedure *completes*(PDAG $\mathcal{G}$, non-colliders $\mathfrak{N}$)
  $\mathbf{U} := \{X \to Y, Y \to X\}_{X-Y \in \mathcal{G}}$
  for $X \to Y$ in $\mathbf{U}$ **do**
    $\mathcal{G}' := (\mathcal{G} \setminus \{X - Y\}) \cup \{X \to Y\}$
    if $\forall_{V \in pa(Y;\mathcal{G})} \langle X, Y, V \rangle \notin \mathfrak{N}$ **and** $ext\,(\mathcal{G}', \mathfrak{N})$ **then**
      remove edges of $\mathcal{G}'$ from $\mathbf{U}$
    else orient $Y \to X$ in $\mathcal{G}$, remove $Y \to X$ from $\mathbf{U}$

procedure *ext*($\mathcal{G}$, $\mathfrak{N}$)
  $\mathcal{H} := copy\,(\mathcal{G})$
  repeat
    for X in $\mathcal{V}\,(\mathcal{H})$ **such that** $ch\,(X; \mathcal{H}) = \emptyset$ **do**
      if $\langle V_1, X, V_2 \rangle \notin \mathfrak{N}$ for every $V_1, V_2 \in adj\,(X; \mathcal{H})$
        orient $Y \to X$ in $\mathcal{G}$ for every $Y \in ne\,(X; \mathcal{H})$
        $\mathcal{H} := \mathcal{H} \setminus \{X\}$
        break
    else return False
  until $\mathcal{H}$ is empty
  return True

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 7).

Theorem 34. *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 $\mathfrak{N}\,(\mathcal{G}) = \{\langle X, Y, Z \rangle \in \mathfrak{N} \mid \{X, Y, Z\} \subseteq \mathcal{V}(\mathcal{G})\}$ 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 Algorithm 7 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 $\mathcal{G}'' \cup \{X\} \cup \{Y \to X \mid Y \in ne\,(X; \mathcal{G})\}$. 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 $Y \to X \leftarrow Z$, $\langle Y, X, Z\rangle \notin \mathfrak{N}$ (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. □

4.2 RpCD 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.

Algorithm 8 PC algorithm

initialize a complete undirected graph $\mathcal{G}$.
$\ell := 0$
repeat
  for every ordered pair (Y, X) **such that** $Y - X \in \mathcal{G}$ **do**
    for every $\mathbf{S} \subseteq ne(X; \mathcal{G}) \setminus \{Y\}$ **such that** $|\mathbf{S}| = \ell$ **do**
      if $X \perp\!\!\!\perp Y \mid \mathbf{S}$ **then**
        remove $Y - X$ from $\mathcal{G}$.
        break
  $\ell := \ell + 1$
until $|ne(X; \mathcal{G})| - 1 < \ell$ for every $X \in \mathbf{G}$

initialize $\mathfrak{U}$ with unshielded triples from $\mathcal{G}$.
$\mathfrak{N} := \emptyset$, $\mathcal{H} := \langle \mathbf{A}, \{X - Y \mid P.Y - V_X \in \mathcal{G}\}\rangle$
for every $(X, Y, Z) \in \mathfrak{U}$ **do**
  continue if $\langle X, Y, Z\rangle \in \mathfrak{N}$
  if exists $\mathbf{S} \subseteq adj(X; \mathcal{G}) \cup adj(Z; \mathcal{G})$ **such that** $Z \perp\!\!\!\perp X \mid \mathbf{S}$ **then**
    if $\mathbf{S} \cap Y = \emptyset$ **then** orient $X \to Y \leftarrow Z$ in $\mathcal{G}$
    else add $\langle X, Y, Z\rangle$ to $\mathfrak{N}$
  orient edges in $\mathcal{G}$ with sound rules with $\mathfrak{N}$.

RpCD (see Algorithm 9) extends the key ideas of the PC algorithm (Spirtes

Algorithm 9 RpCD

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

1: initialize $\mathbf{D}$ with candidate relational 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 $(P.Y, V_X)$ **such that** $P.Y - V_X \in \mathcal{G}$ **do**
6: **for** every $\mathbf{S} \subseteq ne(V_X; \mathcal{G}) \setminus \{P.Y\}$ **such that** $|\mathbf{S}| = \ell$ **do**
7: **if** $V_X \perp\!\!\!\perp P.Y \mid \mathbf{S}$ **then**
8: remove $\{P.Y - V_X, \tilde{P}.X - V_Y\}$ from $\mathcal{G}$.
9: **break**
10: $\ell := \ell + 1$
11: **until** $|ne(V_X; \mathcal{G})| - 1 < \ell$ for every $X \in \mathbf{A}$

12: initialize $\mathfrak{U}$ with canonical unshielded triples from $\mathcal{G}$.
13: $\mathfrak{N} := \emptyset$, $\mathcal{H} := \langle \mathbf{A}, \{X - Y \mid P.Y - V_X \in \mathcal{G}\} \rangle$
14: **for** every $\langle V_X, \mathbf{P}.Y, R.Z \rangle \in \mathfrak{U}$ **do**
15: **if** $\langle X, Y, Z \rangle \in \mathfrak{N}$ **or** $\{X, Z\} \cap ne(Y; \mathcal{H}) = \emptyset$ **or** $\{X, Z\} \cap ch(Y; \mathcal{H}) \neq \emptyset$ **then**
16: **continue**
17: **if** exists $\mathbf{S} \subseteq adj(V_X; \mathcal{G})$ **such that** $R.Z \perp\!\!\!\perp V_X \mid \mathbf{S}$ **then**
18: **if** $\mathbf{S} \cap \mathbf{P}.Y = \emptyset$ **then**
19: orient $X \to Y \leftarrow Z$ in $\mathcal{H}$
20: **else if** $X = Z$ **then**
21: orient $Y \to X$ in $\mathcal{H}$
22: **else**
23: add $\langle X, Y, Z \rangle$ to $\mathfrak{N}$
24: orient edges in $\mathcal{H}$ with sound rules with $\mathfrak{N}$.
25: *completes* $(\mathcal{H}, \mathfrak{N})$
26: **return** $\bigcup_{P.Y - V_X \in \mathcal{G}} \begin{cases} P.Y \to V_X & Y \to X \in \mathcal{H} \\ P.Y - V_X & Y - X \in \mathcal{H} \end{cases}$

et al., 2000) (Algorithm 8) to the relational domain. Phase I of RpCD identifies adjacencies (Lines 1–11) and phase II orients the dependencies (Lines 12–25). 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 12–25, it orients undirected dependencies through conditional independence tests on CUTs. Redundant tests are avoided by skipping (i) already known non-colliders (first term in Line 15), (ii) already oriented edges (second term in Line

15), and (iii) inactive non-colliders (third term in Line 15). At an attribute class level, edges are oriented if forming a collider (Line 18–19) or forming a non-collider having the same attribute classes on its flanking elements (Line 20–21, RBO). All orientations that can be inferred from the sound orientation rules (see Section 4.1.2) are enforced (Line 24). Finally, Line 25 maximally-orients partially-directed class dependency graph with a complete 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 26).

Theorem 35 (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 29) with the completeness of (i) CUTs for unshielded triples, (ii) the CUT-enumerating algorithm for (non-)colliders ($\mathfrak{C}_{\mathbf{A}}^{\mathcal{M}}$ and $\mathfrak{N}_{\mathbf{A}}^{\mathcal{M}}$), and (iii) generalized extensibility (Theorem 34). □

We include a road map (see Figure 4.5) for Markov equivalence class of RCM. RCM and CDG (gray colored) are not available in the problem of causal discovery. Dashed boxes represent partially-known structures — canonical unshielded triples (CUT) are sufficiently (not fully) enumerated to retrieve Markov equivalence class of an RCM. For causal discovery, candidate dependencies are enumerated and refined through independence tests to obtain an "Undirected RCM". Whether a canonical unshielded triple $\langle V_X, \mathbf{P}.Y, R.Z\rangle$ is a canonical unshielded collider or a non-collider is determined by checking orientation at attribute class level (e.g., a collider if $X \to Y \leftarrow Z$ in the CDG) or by checking the separating set $\mathbf{S}$ between V_X and $R.Z$ in the case of causal discovery (e.g., a collider if $\mathbf{S} \cap \mathbf{P}.Y$ is empty).

4.3 Concluding Remarks

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

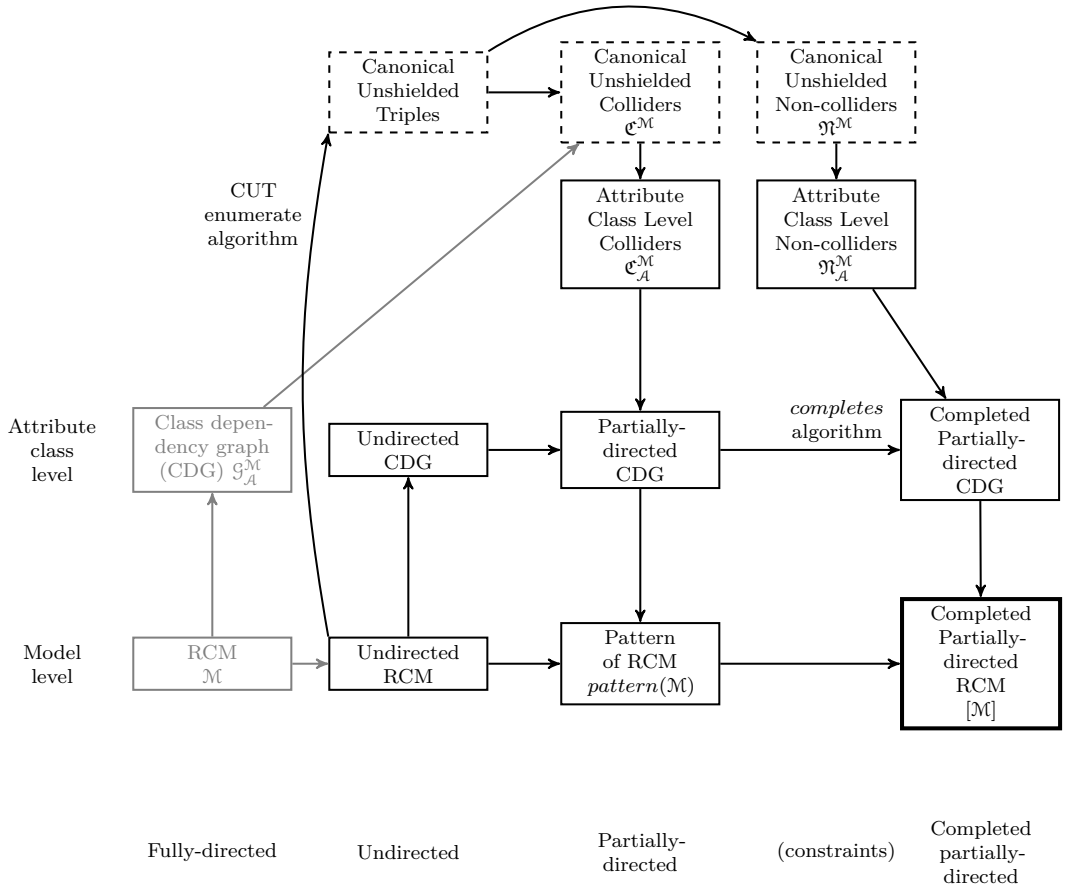


Figure 4.5: Overview of concepts

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.

Chapter 5
Conditional Independence Test for Relational Data

Conditional independence (CI) tests play a central role in statistical inference, machine learning, and causal discovery. Most existing CI tests assume that the samples are independently and identically distributed (iid). However, this assumption often does not hold in the case of *relational data*. We define more general *Relational Conditional Independence* (RCI), a generalization of CI to the relational setting without relying on the current path-like definition of relational variable. We show how, under a set of structural assumptions, we can test for RCI by reducing the task of testing for RCI on non-iid data to the problem of testing for CI on several data sets each of which consists of iid samples. We develop *Kernel Relational CI Test* (KRCIT), a nonparametric test as a principled approach to testing for RCI by relaxing the structural assumptions used in our analysis of RCI. We describe results of experiments with synthetic relational data that show the benefits of KRCIT relative to traditional CI tests that don't account for the non-iid nature of relational data. Note that this chapter explores possibility of more general definition of relational variable. This chapter is based on (Lee and Honavar, 2017a).

5.1 Introduction

Observational and experimental data represent systematic interactions among a set of random variables of interest. Conditional independence (CI) tests constitute essential tools for understanding such interactions. Random variables X and Y are said to be conditionally independent given Z, denoted by $X \perp\!\!\!\perp Y \mid Z$, if and only if the joint distribution P_{xyz} can be factorized as $\mathrm{P}_{x|z}\mathrm{P}_{y|z}\mathrm{P}_{z}$. The notion of CI plays

a central role in statistical inference (Dawid, 1979), probabilistic graphical models (Koller and Friedman, 2009), and causal discovery (Pearl, 2000; Spirtes et al., 2000). A variety of methods including, in particular, nonparametric methods (Fukumizu et al., 2008; Zhang et al., 2011; Doran et al., 2014; Lee and Honavar, 2017b) have been developed to test for CI in settings where the parametric form of the underlying distribution is unknown but a measure of closeness between data samples can be defined, e.g., using a kernel function. However, these methods implicitly or explicitly assume that the data samples are independently and identically distributed (iid).

Many sources of real-world data, e.g., the WWW, citation networks, social networks, biomolecular networks, exhibit a *relational* structure, wherein the data are naturally represented as collections of interlinked entities. In the resulting relational data, e.g., a citation network, the entities, e.g., authors, articles, and institutions, clearly do not constitute iid observations. Methods for learning causal models from relational data rely on oracles that can answer CI queries from such data (Maier et al., 2013a; Lee and Honavar, 2016a) (see Chapter 4). Practical realizations of such algorithms will need to replace such oracles by CI tests against relational data. However, in the relational setting, with the exception of autocorrelated data, e.g., time series (Chwialkowski et al., 2014), where 'closeness' in time, space, or network is well-defined (Flaxman et al., 2016), effective ways to define and test for CI have been lacking. Any attempt to generalize the notion of CI to the relational setting needs to overcome several challenges.

Against this background, inspired by the notion of relational d-separation (Maier et al., 2013a) (see Chapter 3), which generalizes a graphical criterion for CI to a specific model of relational data, we (i) Formalize Relational Conditional Independence (RCI), the relational counterpart of CI. (ii) Examine the dependence and heterogeneity of relational variables in terms of the underlying relational structure. (iii) Based on the preceding analyses, devise a Kernel Relational CI Test (KRCIT) that, to the best of our knowledge, offers the first principled method for testing for RCI. (iv) Describe results of experiments with synthetic relational data that show the benefits of KRCIT relative to traditional CI tests that don't account for the non-iid nature of relational data. RCI and KRCIT offer new ways to understand dependencies in relational data.

5.2 Preliminaries

We follow the notational conventions from statistics, graph theory, and relational models. We use a capital letter X to denote a random variable; $\mathcal{X}$ to denote the range of X; and a lowercase letter x to denote the value of X. Calligraphic letters are also used to represent mathematical objects, e.g., graphs.

We define a labeled (directed or undirected) *graph* $\mathcal{G} := \langle \mathcal{V}, \mathcal{E}, \mathcal{L} \rangle$ where $\mathcal{V}$ denotes a set of vertices (or nodes) and $\mathcal{E}$ a set of edges. Each vertex is assigned a discrete label by a labeling function $\mathcal{L} : \mathcal{V} \mapsto \Sigma$ where Σ is a set of labels. We disallow self-loops. Given an undirected graph $\mathcal{G}$, a *connected component* $\mathcal{G}'$ is a vertex-induced subgraph of $\mathcal{G}$ where there exists a path between every pair of vertices in $\mathcal{G}'$. We denote all connected components in $\mathcal{G}$ by $\mathrm{CC}^{\mathcal{G}}$ and a connected component containing $v \in \mathcal{V}$ by $\mathrm{CC}^{\mathcal{G}}_v$. Two labeled graphs $\mathcal{G}$ and $\mathcal{G}'$ are said to be *isomorphic,* denoted by $\mathcal{G} \cong \mathcal{G}'$, if there exists a bijective function $f : \mathcal{V} \mapsto \mathcal{V}'$ such that

$$\forall_{v \in \mathcal{V}} \mathcal{L}(v) = \mathcal{L}'(f(v))$$

and

$$\forall_{u,v \in \mathcal{V}} (u, v) \in \mathcal{E} \Leftrightarrow (f(u), f(v)) \in \mathcal{E}'.$$

We use a simplified version of Entity-Relationship (ER) model (Chen, 1976) to describe relational data (Friedman et al., 1999; Heckerman et al., 2007; Maier et al., 2013a). The model is represented as a relational schema $\mathcal{S} := \langle \mathbf{E}, \mathbf{R}, \mathbf{A} \rangle$. We omit card as it is not of interest in this chapter. We represent a relational skeleton, whose relationship classes are binary, as an undirected graph of entities. We use "relational structure" to highlight the the graphical structure of a relational skeleton, and "relational data" to mean a set of data that conform to a given relational schema.

5.3 CI Test with Relational Data

We define the notion of relational variables followed by the notion of Relational Conditional Independence (RCI). We provide both a theoretical characterization of RCI as well as a practical approach to testing for RCI.

Consider attribute classes (attributes for short) X, Y, and Z of a relational schema. In the absence of any relational structure, the "data" corresponding to

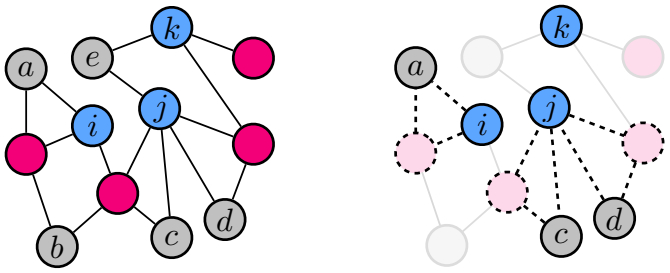


Figure 5.1: A relational skeleton and visualization of a relational variable

instantiations of these random variables in a dataset can be naturally indexed so that (x_i, y_i, z_i) denotes the ith instance drawn from P_{xyz}. However, in the relational setting, there is no such natural index. Hence, we can use a set of items of each item class to serve the role of an index. This indexing scheme generalizes the notion of the 'ith instance' in an iid setting to the notion of 'instantiated by item i' where $i \in \sigma(I)$ for some $I \in \mathbf{I}$. Note that different item classes provide different ways to index relational data.

Definition (Relational Variable)**.** Let $\mathcal{S} := \langle \mathbf{E}, \mathbf{R}, \mathbf{A} \rangle$ be a relational schema and $\sigma \in \Sigma_{\mathcal{S}}$ be an arbitrary relational skeleton. A *relational variable* V is a function from $\sigma(I)$ for some item class $I \in \mathbf{I}$ to a subset of $\{j.X \mid j \in \sigma(J)\}$ for some attribute class X and item class J such that $X \in \mathbf{A}(J)$:

$$V : \sigma(I) \mapsto 2^{\{j.X \mid j \in \sigma(J)\}}$$

where every $i \in \sigma(I)$ is connected to $j \in \sigma(J)$ such that $j.X \in V(i)$.

As a simple but concrete example, a relational variable V, 'smoking status of one's neighbors', is defined with I, J both being the 'Person' item class, X corresponding to the attribute class 'smoking', where $j \in \sigma(J)$ where $j.X \in V(i)$ is i's neighbor for every person $i \in \sigma(I)$. Given a person Arthur who has two neighbors Sally and Sean, $V(\text{arthur}) = \{\text{sally}.Smoke\, \text{sean}.Smoke\}$ results a 2-dimensional random variable. In Figure 5.1, we illustrate another example of relational variables involving three entity classes — (Left) a small relational skeleton σ as an undirected graph of entities of three classes, Blue, Magenta, and Gray; and (Right) an example of a relational variable V where $V(\ell)$ refers to the multiset of attribute X of a set of Gray items forming a triangle with the given Blue item ℓ and a Magenta

item. Hence, $V(i) = \{a.X\}$; $V(j) = \{c.X, d.X\}$; and $V(k) = \emptyset$. The definition of relational variable in this chapter extends the notion of relational variable in RCM.

We describe *isomorphism-invariance*, a desired property of a relational variable. This property ensures that the interpretation of a relational variable is consistent across graph-isomorphic connected components of any relational skeleton $\sigma \in \Sigma_{\mathcal{S}}$. Let $\mathcal{S} := \langle \mathbf{E}, \mathbf{R}, \mathbf{A} \rangle$ be a relational schema and $\sigma \in \Sigma_{\mathcal{S}}$ be an arbitrary instantiation of the schema. Let $\mathrm{CC}_a^\sigma := \langle \mathcal{V}_a, \mathcal{E}_a, \mathcal{L}_a \rangle$ and $\mathrm{CC}_b^\sigma := \langle \mathcal{V}_b, \mathcal{E}_b, \mathcal{L}_b \rangle$ where each item is labeled with its item class. Let $\mathbf{f}_{a,b}^\sigma$ be a set of mapping functions demonstrating $\mathrm{CC}_a^\sigma \cong_{a,b} \mathrm{CC}_b^\sigma$, that is, $\mathbf{f}_{a,b}^\sigma := \{f \mid \forall_{v \in \mathcal{V}_a} \mathcal{L}_a(v) = \mathcal{L}_b(f(v)) \wedge \forall_{u,v \in \mathcal{V}_a} (u, v) \in \mathcal{E}_a \Leftrightarrow (f(u), f(v)) \in \mathcal{E}_b \wedge f(a) = b\}$. Let U be a relational variable with domain $\sigma(I)$ where $I \in \mathbf{E} \cup \mathbf{R}$. If

$$\forall_{f \in \mathbf{f}_{a,b}^\sigma} \{f(i) \mid i.X \in U_a\} = \left\{j \mid j.X \in U_{f(a)}\right\}$$

for any $a, b \in \sigma(I)$ for any $\sigma \in \Sigma_{\mathcal{S}}$, then, U is said to be *isomorphism-invariant.*

Understanding an item attribute, e.g., $j.X$, as a random variable, $V(i)$ (or V_i for short) is a set of random variables. The 'value' of V_i is denoted by $v_i := (j.x)_{j.X \in V_i} \in \mathcal{X}^{|V_i|}$, indexed with item attributes in V_i. In the preceding example, $v_{\text{arthur}} = \{\text{sally} : \mathsf{False}, \text{sean} : \mathsf{True}\}$. Noting that a relational variable itself is *not* a random variable, we proceed to carefully define *relational conditional independence.* Without loss of generality, we consider the case where the conditioning is on a single relational variable although, in general, the conditioning can be on a set of relational variables.

Definition (Relational Conditional Independence)**.** Let $\{U, V, W\}$ be relational variables with a common domain, $\sigma(I)$, defined on a relational schema $\mathcal{S}$ where $\sigma \in \Sigma_{\mathcal{S}}$. Let item attributes $\bigcup_{i \in \sigma(I)} U_i \cup V_i \cup W_i$ be random variables. Then, U and V are said to be independent to each other given W, denoted by $(U \perp\!\!\!\perp V \mid W)_\sigma$, if and only if

$$\forall_{i \in \sigma(I)} \mathrm{P}_{u_i v_i w_i} = \mathrm{P}_{u_i|w_i} \mathrm{P}_{v_i|w_i} \mathrm{P}_{w_i}.$$

It is easy to see that this definition of relational CI (RCI) generalizes traditional CI where iid samples are drawn from P_{xyz}. Let U be "smoking status of oneself" and W be "smoking status of one's parents", e.g., $U(\text{arthur}) = \{\text{arthur}.Smoke\}$. We might ask $(U \perp\!\!\!\perp V \mid W)_\sigma$: "Is one's smoking status independent of one's neighbors' smoking status given one's parents' smoking status?" While, in the real world,

answering such question can be quite difficult because of the complexity and partial observability of interactions among people, the notion of relational CI, once operationalized, can help extract useful insights from relational data.

5.3.1 Relational Variables and Parameter Tying

In the relational setting, assuming that each item attribute of the same attribute class is identically distributed (e.g., sally.$Smoke \stackrel{d}{=}$ arthur.$Smoke$), would be tantamount to ignoring the relational structure of the data. On the other hand, if we were to let each item attribute share no commonality whatsoever with any other item attribute, checking for RCI becomes nearly impossible. Hence, we restrict our attention to the practically relevant setting where the item attributes of each attribute class share some commonality, e.g., the joint probability distribution of *all* item attributes can be modeled as a directed acyclic graph $\mathcal{G}$ of item attributes (Friedman et al., 1999),

$$\mathrm{P}(\mathbf{v}) = \prod_{X \in \mathbf{A}} \prod_{i \in \sigma(\mathbf{A}^{-1}(X))} \mathrm{P}\left(i.x \mid pa\left(i.X; \mathcal{G}\right)\right)$$

where $\mathbf{v}$ represents values of all item attributes in σ. For every item of the same item class, e.g., $i, j \in \sigma(I)$, $\mathrm{P}(i.X \mid pa^{\mathcal{G}}(i.X)) = \mathrm{P}(j.X \mid pa(j.X; \mathcal{G}))$ is often assumed if $pa(i.X; \mathcal{G})$ and $pa(j.X; \mathcal{G})$ are matched under model-specific assumption, e.g., their averages are the same (Friedman et al., 1999) (i.e., the ratio of one's smoking neighbors to one's neighbors). This is called *parameter-tying* or *templating* (Koller, 1999) and is widely used in relational or temporal domains to capture time-invariant aspects of the domain. We relate parameter tying and item attributes to the homogeneity (in the sense of being identically distributed) and independence of random variables. Let $\mathcal{G} \cong_{i,j} \mathcal{G}'$ denote graph isomorphism between $\mathcal{G}$ and $\mathcal{G}'$ subject to the constraint that vertex $i \in \mathcal{G}$ must be matched to $j \in \mathcal{G}'$.

Proposition 36 (Identically Distributed Random Variables)**.** *Let $\mathcal{G}$ be a directed acyclic graph representing a conditional independence structure of item attributes where each item attribute, e.g., $k.X$, is labeled with its attribute class, e.g., X. Given that the parameters are tied, random variables $i.X$ and $j.X$ in $\mathcal{G}$ are identically distributed if $\mathcal{G}[an(i.X; \mathcal{G})] \cong_{i.X, j.X} \mathcal{G}[an(j.X; \mathcal{G})]$.*

Proposition 37 (Independent Random Variables)**.** *Two random variables $i.X$ and $j.X$ are independent if $\mathcal{G}\left[an\left(i.X;\mathcal{G}\right)\right] \cap \mathcal{G}\left[an\left(j.X;\mathcal{G}\right)\right] = \emptyset$.*

Proofs for both propositions directly follow from Markov condition that a random variable is independent of its non-descendants given its parents in a DAG $\mathcal{G}$. For an undirected graph $\mathcal{G}$ of item attributes (e.g, Markov random field) labeled as above, $\mathrm{CC}^{\mathcal{G}}_{i.X} \cong_{i.X,j.X} \mathrm{CC}^{\mathcal{G}}_{j.X}$ and $\mathrm{CC}^{\mathcal{G}}_{i.X} \neq \mathrm{CC}^{\mathcal{G}}_{j.X}$ would be a sufficient condition for $i.X, j.X \overset{\text{iid}}{\sim} \mathrm{P}$ for some distribution P under the parameter-tying assumption where graph isomorphic maximal cliques share the same parameters.

However, we have no access to the underlying CI structure $\mathcal{G}$ of *item attributes*. Hence, we deduce an iid condition through *items* on an observed skeleton σ:

Assumption 38. *Let $\mathcal{S} := \langle \mathbf{E}, \mathbf{R}, \mathbf{A} \rangle$ be a relational schema and σ be a relational skeleton of $\mathcal{S}$. Let i and j be items in σ and $X \in \mathbf{A}\left(I\right)$ of $I \in \mathbf{I}$. Then, random variables $i.X$ and $j.X$ are independent and identically distributed if*

$$\left(\mathrm{CC}^{\sigma}_{i} \cong_{i,j} \mathrm{CC}^{\sigma}_{j}\right) \wedge \left(\mathrm{CC}^{\sigma}_{i} \neq \mathrm{CC}^{\sigma}_{j}\right)$$

Note that the condition is sufficient but not necessary for the random variables corresponding to the item attributes to be iid. This is based on our understanding of how parameter-tying assumption is realized in a given relational structure and determines the qualitative aspects (i.e., homogeneity and independence) of the random variables corresponding to the item attributes.

5.3.2 Handling Non-iid Variables

It is possible that a relational structure induces dependent and non-identically distributed (i.e., heterogeneous) item attributes even when parameter tying is assumed. Hence, we cannot simply apply a traditional CI test to test $\left(U \perp\!\!\!\perp V \mid W\right)_{\sigma}$ on the flattened version of relational data $\left\{\left(u_i, v_i, w_i\right)\right\}_{i \in \sigma(I)}$ where $\sigma(I)$ is the common domain of U, V, and W.

Our solution to this problem is to perform CI tests by decomposing, with respect to a given RCI query, the set of items $\sigma\left(I\right)$ into subsets of items such that each subset yields a set of iid observations under the above assumption. Consider a function $id : \sigma\left(I\right) \mapsto \mathbb{Z}$ such that for $id\left(i\right) = id\left(j\right)$ only if $\left(u_i, v_i, w_i\right) \overset{d}{=} \left(u_j, v_j, w_j\right)$ and $\left(u_i, v_i, w_i\right) \perp\!\!\!\perp \left(u_j, v_j, w_j\right)$ (i.e., $\mathrm{CC}^{\sigma}_{i} \cong_{i,j} \mathrm{CC}^{\sigma}_{j}$ and $\mathrm{CC}^{\sigma}_{i} \neq \mathrm{CC}^{\sigma}_{j}$ given that U, V,

and W are *isomorphism-invariant*. Then, a traditional CI test, treating U, V, and W as random variables, $U \perp\!\!\!\perp V \mid W, id$ will remove bias introduced by the relational structure provided we have large enough samples per condition (i.e., a large number of CCs per isomorphic class). Such a naive solution, however, has severe limitations in practice: i) It is possible, in the worst case, that all items in relational data are connected and ii) Each connected component might be non-isomorphic to others.

5.4 A Kernel Relational CI Test

To address the limitations noted above, we will relax the requirements that the connected components be isomorphic and that items be partitioned into non-overlapping connected components. Recent progress in kernel-based nonparametric tests (e.g., two-sample tests or CI) allows us to utilize the notion of *closeness* between samples to test for homogeneity or conditional independence. We proceed to show how kernel-based conditional independence tests, originally introduced for testing independence of iid random variables from data, can be adapted to the relational setting, by defining a novel kernel function for relational variables.

5.4.1 A Kernel for Relational Variables

We provide kernels for relational variables that reflect our understanding of relational structure as in Section 5.3.1. Consider a relational variable U, associated with the attribute class X. A kernel function for U, k_U, measures *similarity* between two instantiations of U where each instantiation, e.g., $u_i \in \mathcal{X}^{|U_i|}$, consists of a set of item attributes. We illustrate our approach using the R-convolution kernel (Haussler, 1999), which computes the kernel over two sets as the sum of kernel values for every pair of elements from two sets. Thus, we define k_U as

$$k_U\left(u_i, u_j; k_{\mathrm{IA}}^X\right) := \sum_{a.X \in U_i} \sum_{b.X \in U_j} k_{\mathrm{IA}}^X\left(a.X, b.X\right) \tag{5.1}$$

where the base kernel $k_{\mathrm{IA}}^X\left(a.X, b.X\right)$ measures the similarity between two item attributes. Based on the analysis in Section 5.3.1, U_i and U_j do not necessarily yield identically-distributed item attributes. Subgraphs as contexts can be combined with item attributes in an observation, e.g., (u_i, v_i, w_i). By augmenting contexts to each

value represented by relational variables, each observation is mapped to a higher dimensional space where the proximity of the two augmented observations, say $(u_i, v_i, w_i)'$ and $(u_j, v_j, w_j)'$, implies that they are not only similar in their values but also values are from similarly distributed random variables. Hence, we design $k^X_{\mathrm{IA}}(a.X, b.X)$ by taking both homogeneity and attribute values into consideration: k^X_{IA} is defined as a product kernel of the kernel for homogeneity k_σ and a kernel for attribute values k_x:

$$k^X_{\mathrm{IA}}(a.X, b.X) := k_\sigma(a, b)\, k_x(a.x, b.x)$$

where the kernel for attribute values is typically defined using a standard kernel for the data type, for example, a Gaussian RBF kernel if $\mathcal{X} \subseteq \mathbb{R}^d$.

We would like to note that the R-convolution kernel based kernel definition is one possible definition. However, there can be cases where multi-sets of different sizes should have zero kernel value. Hence, domain-related background knowledge should guide the design of kernel for multi-sets. We now elaborate the kernel for homogeneity k_σ below.

5.4.1.1 Kernel for Homogeneity among Item Attributes

We postulate that two random variables $a.X$ and $b.X$ will be *similarly distributed* if the corresponding items appear in *similar contexts* in σ, i.e., similarly interconnected neighbors in σ. Therefore, the degree to which two item attributes $a.X$ and $b.X$ are identically distributed can be approximated by the similarity of the context of a and the context of b in σ.

We use h-hop neighbors of an item in σ to induce a context of the item for practicality since a connected component, e.g., CC^σ_a, can be as large as the given relational skeleton σ. Thus, we design the kernel for homogeneity $k_\sigma(a, b)$ as a graph kernel between two labeled graphs $\sigma[ne^\sigma_h(a)]$ and $\sigma[ne^\sigma_h(b)]$ where $ne^\sigma_h(a)$ is a set of items in σ that are reachable in no more than h hops from item a in σ. Each item in a context, e.g., $\sigma[ne^\sigma_h(a)]$, is labeled with its item class except the item a, which is assigned to a special label allowing a graph kernel between $\sigma[ne^\sigma_h(a)]$ and $\sigma[ne^\sigma_h(b)]$ to match a and b. This reflects '$\cong_{i,j}$', graph isomorphism with an additional constraint (Assumption 38).

We choose to exploit an existing graph kernel for labeled graphs. For example a

shortest-path kernel (Borgwardt and Kriegel, 2005) is given by

$$k_{\mathrm{SP}}\left(\mathcal{G},\mathcal{G}'\right) := \sum_{c,d\in\mathcal{V}}\sum_{c',d'\in\mathcal{V}'} k_n\left(c,c'\right)\cdot k_n\left(d,d'\right)\cdot k_l\left(d_{\mathcal{G}}\left(c,d\right),d_{\mathcal{G}'}\left(c',d'\right)\right)$$

with the choice of kernels on node (i.e., item) k_n and on shortest path length k_l where $d_{\mathcal{G}}\left(c,d\right)$ is a shortest path length between c and d in $\mathcal{G}$. We use the Dirac kernel for both k_n and k_l, that is, k_n is 1 if two items have the same label and 0, otherwise, and k_l is 1 if two lengths are the same and 0, otherwise.

5.4.1.2 Kernel for Homogeneity among Observations

The use of contexts does not supplant the role of the indicator *id*. Hence, we introduce a new variable G to play the role of *id* without dealing with dependent observations. With G, the question of RCI, $(U \perp\!\!\!\perp V \mid W)_\sigma$, becomes that of traditional CI, $U \perp\!\!\!\perp V \mid W, G$ and, similarly, an unconditional query $(U \perp\!\!\!\perp V)_\sigma$ becomes $U \perp\!\!\!\perp V \mid G$.

We have already seen the kernel for relational variable which considered both contexts and values. Taking the value part out from the definition of the kernel, we can get a kernel for homogeneity among observations. Since an observation consists of three (two if unconditional) relational variables, we use a product kernel. We define

$$k_G(i,j) := k_U\left(u_i,u_j;k_\sigma\right)k_V\left(v_i,v_j;k_\sigma\right)k_W\left(w_i,w_j;k_\sigma\right) \tag{5.2}$$

Note that while we have used the R-convolution kernel for relational variables and the shortest-path kernel as our graph kernel to illustrate our idea, the approach can accommodate other kernels, e.g., the optimal assignment kernel (Kriege et al., 2016) for relational variables and a Weisfeiler-Lehman kernel (Shervashidze et al., 2011) for graphs. In practice, the choice of kernel can be guided by the knowledge of the domain.

5.4.2 Treating Dependent Observations

Now we briefly discuss how we can handle dependent observations. In relational data, the dependencies among observations can arise for different reasons. In previous section, we showed that two item attributes become dependent if they share the same ancestors (Proposition 37). However, in some settings, we can ignore some

types of dependence among observations. For instance, consider a hidden Markov model with hidden variables X and observed variables Y where $X_{t-1} \perp\!\!\!\perp Y_t \mid X_t$ and $\mathrm{P}(Y_t \mid X_t)$ and $\mathrm{P}(X_t \mid X_{t-1})$ are time-invariant. Simply running a traditional CI test on a sample $\{(x_{t-1}, y_t, x_t)\}_{t=2}^{n}$ would likely result in the null hypothesis not being rejected in spite of correlations among observations, e.g., (X_{t-1}, Y_t, X_t) and $(X_{t-2}, Y_{t-1}, X_{t-1})$. Variables like X_t and X_{t-1} are naturally represented as (temporally) related variables, the dependencies among which can be broken by conditioning on an appropriate set of variables. In this regard, we treat 'dependent observations' to be resolved explicitly through conditionals instead of being implicitly removed, e.g., non-iid CI test for *autocorrelated* data (Flaxman et al., 2016).

5.4.3 Validity of CI Test on Flattened Representation

We provide a sufficient condition under which a flattened sample of relational data conditioned on G correctly transforms the question of an RCI query $(U \perp\!\!\!\perp V \mid W)_\sigma$ to a traditional CI query $U \perp\!\!\!\perp V \mid W, G$. We consider the alternative hypothesis $(U \not\perp\!\!\!\perp V \mid W)_\sigma$ to satisfy $\forall_{i \in \sigma(I)} U_i \not\perp\!\!\!\perp V_i \mid W_i$ instead of just $\exists_{i \in \sigma(I)} U_i \not\perp\!\!\!\perp V_i \mid W_i$.

Condition 39. $U_i, V_i \perp\!\!\!\perp U_j, V_j \mid W_i$ and $(w_i, g_i = w_j, g_j) \to (U_i, V_i) \stackrel{d}{=} (U_j, V_j)$ for every $i \neq j \in \sigma(I)$.

This condition simply makes a set of (U, V) samples into a set of iid samples where either $U \perp\!\!\!\perp V$ or $U \not\perp\!\!\!\perp V$ holds for each condition. In addition to the first condition, we provide a relaxed sufficient condition only for the null hypothesis.

Condition 40. $V_i \perp\!\!\!\perp V_j \mid W_i$ and $(w_i, g_i = w_j, g_j) \to V_i \stackrel{d}{=} V_j$ for every $i \neq j \in \sigma(I)$.

This condition only makes V iid for each condition. However, it is sufficient to observe $U \perp\!\!\!\perp V$ if $\forall_{i \in \sigma(I)} U_i \perp\!\!\!\perp V_i \mid W_i$. Otherwise, $U \not\perp\!\!\!\perp V \mid W, G$ will hold in *most cases* unless the aggregation of (U_i, V_i) per condition makes such dependence vanish.

5.4.4 Algorithm

By supplying customized kernels for U, V, W, and G (Equation 5.1 and 5.2), a kernel CI test in an iid setting will decide whether to accept or reject the null

Algorithm 10 KRCIT

Input: σ: relational data; U, V, W: relational variables of base item class I; k_U, k_V, k_W: kernels for U, V, W; k_σ: a kernel for subgraphs of items; CI: the base kernel-based CI test

1: $\Omega \leftarrow \{(u_i, v_i, w_i)\}_{i \in \sigma(I)}$, which is $(\mathbf{u}, \mathbf{v}, \mathbf{w})$
2: $k_G(\cdot, \cdot) \leftarrow k_U(\cdot, \cdot; k_\sigma) k_V(\cdot, \cdot; k_\sigma) k_W(\cdot, \cdot; k_\sigma)$
3: $\mathbf{K}_u, \mathbf{K}_v, \mathbf{K}_w \leftarrow$kernel matrices for $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$.
4: $\mathbf{K}_g \leftarrow$kernel matrix for $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ with k_G
5: $\mathbf{K}_{wg} \leftarrow \mathbf{K}_w \odot \mathbf{K}_g$
6: **return** CI $(\mathbf{K}_u, \mathbf{K}_v, \mathbf{K}_{wg})$

hypothesis given a flattened sample Ω. We illustrate the pseudocode of the kernel relational conditional independence test, KRCIT, in Algorithm 10. If the given query is unconditional such that W is undefined, then $\mathbf{K}_w = 1$.

We considered the following two kernel-based conditional independence tests as a base CI test for KRCIT: Kernel CI Test (KCIT, Zhang et al., 2011) which uses the norm of conditional cross-covariance operator in RKHS (reproducing kernel Hilbert space); and Self-Discrepancy CI Test (SDCIT, Lee and Honavar, 2017b) which uses RKHS distance between the given sample representing P_{xyz} and the sample modified to mimic $\mathrm{P}_{x|z}\mathrm{P}_{y|z}\mathrm{P}_z$. The time complexity of KRCIT will depend not only on the size of flattened sample but also the cardinality of relational variables, and the size of subgraphs (i.e., hops), etc. If the cardinality of relational variables and the number of hops used to specify contexts are fixed to small constants, computing kernel matrices requires $O(n^2)$ time where n is the size of flattened sample. Then, the time complexity of KRCIT with KCIT as a base CI test is $O(n^3)$ since that of KCIT is $O(n^3)$.

5.5 Empirical Evaluation

We report results of experiments that examine the performance of KRCIT in testing RCI on relational data using synthetic relational data where we know the ground truth RCI. We used RCM (Maier et al., 2013a; Lee and Honavar, 2016a), which is a generative model for relational data where a set of assumed causal relationships among relational variables specifies how values of item attributes are generated given a relational skeleton.

5.5.1 Methods

We compare the performance of KRCIT, traditional CI tests that do not account for the relational structure of the data, and an alternative RCI test that makes use of context using residualization (Flaxman et al., 2016) where regression is used to remove dependence of a variable on a set of conditionals. For example, assume that one wants to test $X \perp\!\!\!\perp Y$ where X and Y are two time series. By regressing each X and Y on time T, one can obtain $\epsilon_{x|t} := X - \hat{E}[X|T]$ and $\epsilon_{y|t} := Y - \hat{E}[Y|T]$. Then, the test becomes $\epsilon_{x|t} \perp\!\!\!\perp \epsilon_{y|t}$ under a set of assumptions. In the case of RCI, one can residualize values in the given relational skeleton to remove dependence on the 'context' of each attribute value. Formally, let $X, Z \in \mathbb{R}^m$ be two random variables where we seek for $\epsilon_{x|z}$ such that $X := f(Z) + \epsilon_{x|z}$. We can train a Gaussian process regression, i.e., $f \sim \mathcal{GP}(0, k)$, by maximizing total marginal likelihood. Then, $\epsilon_{x|z} := X - \hat{X} := (I + \sigma^{-2}\mathbf{K}_z^*)^{-1} X$. Both $\mathbf{K}_z^*$ and σ^2 are learned through Gaussian process regression employing, e.g., a Gaussian RBF kernel and a white noise kernel.

Assume that we only have access to kernel matrices $\mathbf{K}_x$ and $\mathbf{K}_z$. Following Zhang et al. (2011), we can use the empirical kernel map for $\mathbf{x}$, $\boldsymbol{\psi}_x := \mathrm{V}\Lambda^{\frac{1}{2}}$, where V and Λ are obtained through the eigendecomposition of the kernel matrix $\mathbf{K}_x = \mathrm{V}^\top \Lambda \mathrm{V}$. Similarly, we can obtain $\boldsymbol{\psi}_z$. Then, a Gaussian process regression with a linear kernel and a white noise kernel can be used to learn $\mathbf{K}_z^*$. In this case, we focus on the kernel matrix for residuals given by $\mathbf{K}_{x|z} := \mathbf{R}\mathbf{K}_x\mathbf{R}$ where $\mathbf{R} := (I + \sigma^{-2}\mathbf{K}_z^*)^{-1}$. Following Flaxman et al. (2016), we use the expectation of $\mathbf{K}_{x|z}$, which is given by $\mathbf{K}_z^*\mathbf{R} + \mathbf{R}\mathbf{K}_x\mathbf{R}$. We list methods to be compared in our experiments.

- **(Naive)**: We use Hilbert-Schmidt Independence Criterion (HSIC, Gretton et al., 2005) for unconditional cases which uses the eigenspectrum of covariance operators in RKHS. Otherwise, either KCIT or SDCIT is used.

- **(Residualized)**: We residualize values of a given relational skeleton based on contexts (i.e., replace values to its residuals). Then, aforementioned naive tests are used. We append a prefix 'R-' to denote 'residualized'. For example, R-HSIC is an unconditional test based on HSIC with residualization.

- **(Residual Kernel)**: Residuals are computed in RKHSs and the kernel for residuals is obtained as described above. Then, naive tests are used where

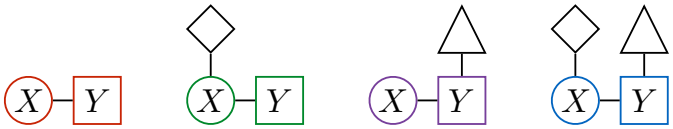


Figure 5.2: Four connected components composing a relational skeleton

computing kernel for two values (e.g., an RBF kernel) is replaced by looking up the residual kernel matrix. We append a prefix 'RK-'.

- **(KRCIT)**: We use a postfix '-K' or '-SD' to denote KRCIT with KCIT or with SDCIT, respectively.

We implemented KRCIT and other kernel CI tests in Python. Throughout experiments, we considered real-valued attributes and an RBF kernel, e.g., $k_x\left(x', x''\right) = \exp\left(-\gamma_x \|x' - x''\|^2\right)$, is used for each attribute where $\gamma_.$ is chosen as $\left(2\sigma^2\right)^{-1}$ where $\sigma^2 = 0.1^2$ is the variance of Gaussian noise we used in data generating processes. Additionally, we normalized both R-convolution kernel and shortest-path kernel, i.e., $k'\left(a, b\right) := k\left(a, b\right) / \sqrt{k\left(a, a\right) k\left(b, b\right)}$. We report how correctly null hypothesis is rejected by a given test (power) measured by Area Under Power Curve (AUPC), which is area under cumulative density function of p-values, and how incorrectly null hypothesis is rejected measured by type-I error rates given $\alpha := 0.05$.

5.5.2 A Simple Example for Visualization

We start with a simple example that explains how and why KRCIT is better for relational CI testing than standard CI tests that assume iid data. Briefly, we considered a relational schema where attribute class X and Y associates with entity class 'circle' and 'square', respectively. There are other item classes and attribute classes in the schema. A relational skeleton σ consists of CCs of four different structures as shown in Figure 5.2. We designed a generative model such that X is a function of the value in an adjacent 'rhombus' item and Y is a function of X in an adjacent circle and the value in an adjacent 'triangle' item. That is, we make sure that $i.X$ and $j.X$ are not identically-distributed if i is adjacent to a rhombus item while j is not. The same idea also applies to Ys.

We controlled *randomness* of relationships between circles and squares: non-random relationship represents that a resulting relational skeleton consists of only circles that are adjacent to a rhombus are connected to squares that are adjacent

to a triangle (1st and 4th components in Figure 5.2), while randomized relationship exhibits a relational skeleton where all four components are balanced. We also controlled *heterogeneity*: the extent to which distributions of (X, Y) of different components diverge from each other. Description of procedure of the test is detailed as follows.

Relational Schema There are four entity classes A, B, C, and D where each associates with attribute class X, Y, S, and T, respectively. There are three relationship classes between A and B, A and C, and B and D. We used circle, square, rhombus, and triangle to refer A, B, C, and D.

Relational Skeleton We generate relational skeletons varying degrees of randomness from 0 to 1 where randomness of 0 is referred to as 'biased'. Given randomness $0 \leq p \leq 1$, we initially generate a fully biased relational skeleton σ, then randomize some of edges between $\sigma(A)$ and $\sigma(B)$ to acquire a relational skeleton of desired randomness p.

Let n be the number of entities for each of A and B (we set $n = 200$). There are $n/2$ entities for C and D, respectively. Hence, let $\sigma(A) = \{a_i\}_{i=1}^{n}$, $\sigma(B) = \{b_i\}_{i=1}^{n}$, $\sigma(C) = \{c_i\}_{i=1}^{n/2}$, and $\sigma(D) = \{d_i\}_{i=1}^{n/2}$. We connect C items to A items and D items to B items, $\{(a_i, c_i)\}_{i=1}^{n/2} \subset \sigma$ and $\{(b_i, d_i)\}_{i=1}^{n/2} \subset \sigma$. Then, we "initially" connect a_i and b_i, $\{(a_i, b_i)\}_{i=1}^{n} \subset \sigma$. That is, by design, an A item having (or not having) a C neighbor is connected to a B item having (or not having) a D neighbor. Given the randomness p, we randomly pick np B items and shuffle their A neighbors.

Relational Causal Model We use a linear Gaussian noise model with sum aggregators as follows:

$$
\begin{aligned}
\forall_{c \in \sigma(C)} \quad & c.S := \mu + \epsilon_c \\
\forall_{d \in \sigma(D)} \quad & d.T := \mu + \epsilon_d \\
\forall_{a \in \sigma(A)} \quad & a.X := \sum_{c \in ne(a;\sigma) \cap \sigma(C)} c.S + \mu + \epsilon_a \\
\forall_{b \in \sigma(B)} \quad & b.Y := \sum_{a \in ne(b;\sigma) \cap \sigma(A)} \beta \cdot a.X + \sum_{d \in ne(b;\sigma) \cap \sigma(D)} d.T + \mu + \epsilon_b
\end{aligned}
$$

where every ϵ is an independent Gaussian noise with zero mean and variance 0.1^2. We control the correlation between connected X and Y by adjusting β where $\beta = 0$ implies that X and Y are independently generated, or more precisely, $[A]\,.X$ and $[A, R_{AB}, B]\,.Y$ are independent. The pair of X and Y values generated from this model can be understood as a mixture of four bivariate normal distributions and μ controls the distance between distributions. When $\mu = 0$, all four distributions are centered at $(0, 0)$. Although, we described four distributions having the same mean as 'homogeneous', they have different variances. We test unconditional independence between $[A]\,.X$ and $[A, R_{AB}, B]\,.Y$.

Results

In the left column of Figure 5.3, we visualize four relational data based on the combinations of underlying hypothesis and heterogeneity where color codes correspond to the types of structure which associates with the value (x, y) in Figure 5.2. Utilizing *permuted* samples, we visualize how KRCIT (with SDCIT) and SDCIT, which is a permutation-based test, produce data consistent with the null hypothesis. KRCIT permutes Ys conditioning on G, which corresponds to 'color', while SDCIT simply shuffles Ys. The center and right columns correspond to the permuted sample under KRCIT and a naive test, respectively. When contexts more strongly correlate with values, we can more clearly observe the difference between the permuted samples by KRCIT and by SDCIT.

For each row, if the center plot is significantly different from its corresponding left plot, KRCIT would reject the null hypothesis. For example, KRCIT correctly rejects the null hypothesis for the samples from the alternative hypothesis (row 2nd and 4th). Interestingly, SDCIT also correctly rejects samples from the alternative hypothesis. In Figure 5.4, we plot null samples when relational skeletons exhibit biased relationships. We can similarly observe difference between two tests. Figure 5.5 illustrates type-I errors (given $\alpha := 0.05$) of a naive test (HSIC) and KRCIT-SD based on relational skeletons generated with various degrees of heterogeneity and non-randomness of relationship. KRCIT-SD is robust to heterogeneity and non-random relationships while HSIC is not. Note that residualization approaches utilizing contexts perform similar to KRCIT.

Finally, we illustrate the changes of power of different tests as the strength of dependency between X and Y is increased (Figure 5.6). Tests that use contexts

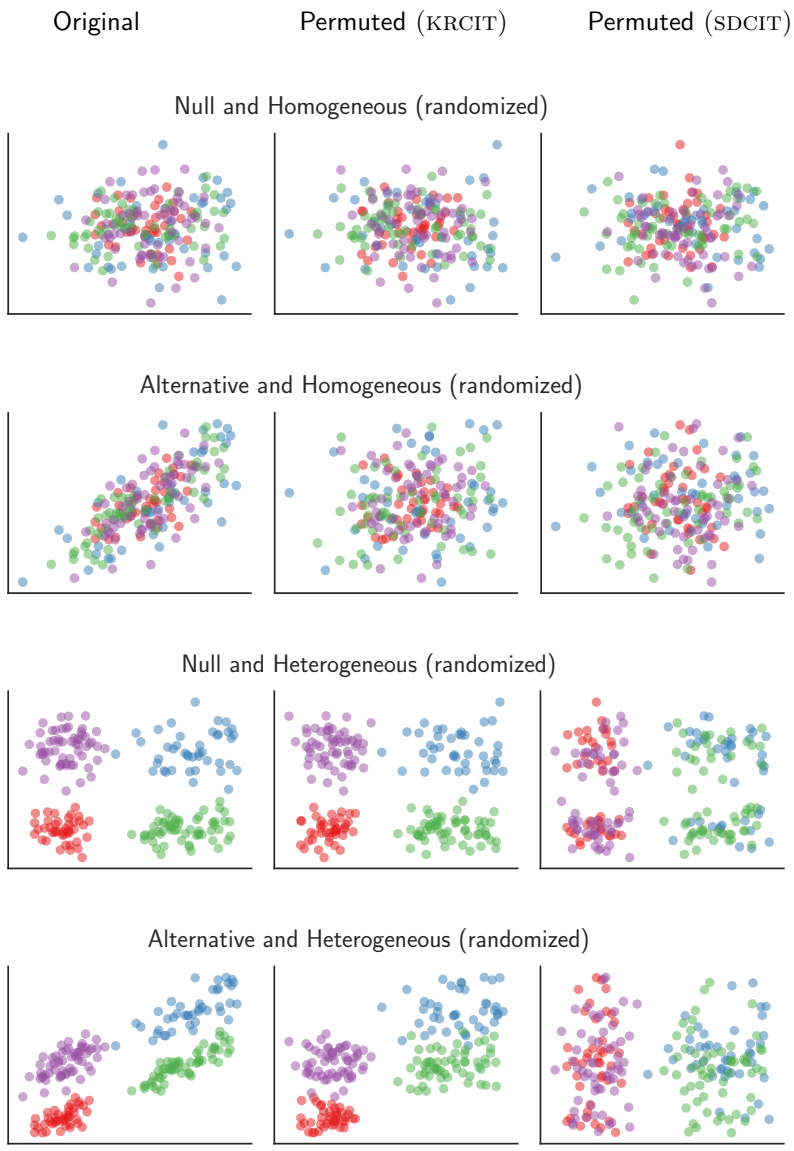


Figure 5.3: Comparisons of the given data (left) and two samples under the null hypothesis using KRCIT with SDCIT (center) and by SDCIT (right), respectively, with randomized relationships

consistently estimates dependency without regard to the underlying conditions while HSIC over-rejects samples from weak dependence in certain conditions where rejection comes partially from other than linear dependence (4th row in Figure 5.3). In summary, whenever contexts provide sufficient information to infer (non-)identically distributed observations, KRCIT is able to eliminate suspicious dependencies due to such heterogeneity.

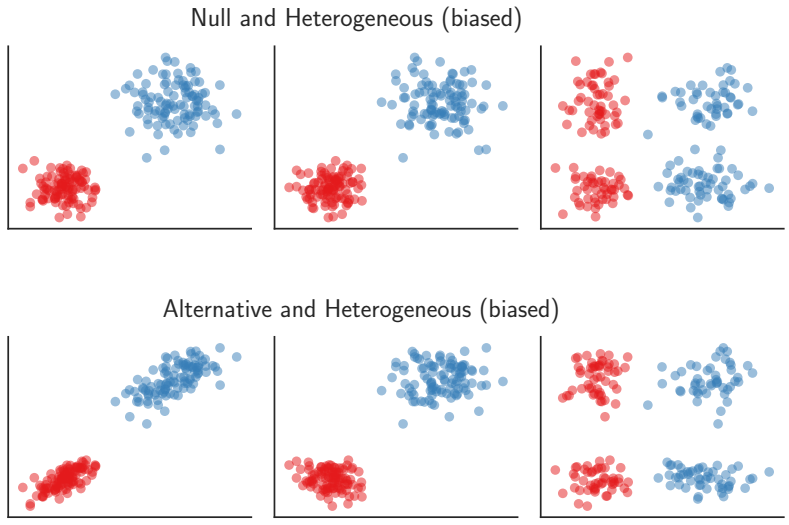


Figure 5.4: Comparisons with fully biased relationship

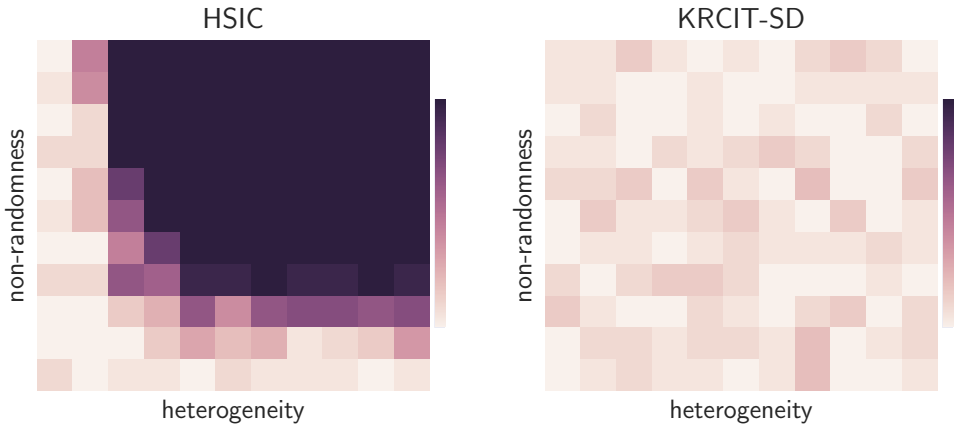


Figure 5.5: Type-I errors varying both non-randomness and heterogeneity

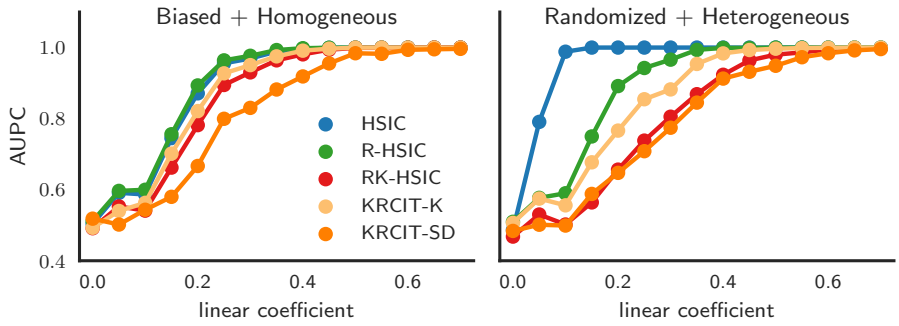


Figure 5.6: AUPCs with varying dependency where relational skeletons are generated with homogeneous and randomized relationships (left) and with heterogeneous and biased relationships (right).

5.5.3 More Complicated Structures

We conduct a similar experiment but with more complicated and larger structures where the cardinality of relational variables is not one and observations are dependent. Such dependence among observations, e.g., two circles, is due to their sharing a *common cause,* e.g., a rhombus. In this experiment, we additionally investigate how different sizes of subgraphs as contexts affect the performance of RCI tests. Contexts based on 1-hop subgraphs contain sufficient information while 2-hop subgraphs will include information about other variable, e.g., a 2-hop subgraph of a circle includes triangle and other circles connected to common rhombuses.

We only change how relational skeletons are generated. We use the same relational schema and relational causal model as shown above.

Relational Skeleton Similarly, we generate $n = 400$ items for A and B and $n/2$ items for C and D. Then, each $a_i \in \sigma(A)$ randomly chooses C neighbor(s) so that a_i has one C neighbor if $1 \leq i \leq \frac{n}{3}$, two neighbors if $\frac{n}{3} < i \leq \frac{2n}{3}$, and three neighbors if $\frac{2n}{3} < i \leq n$. Similarly, each $b_i \in \sigma(B)$ randomly chooses D neighbor(s). As shown in the previous setup, we initialize a relational skeleton with biased relationships between A and B, that is, $\{(a_i, b_i)\}_{i=1}^{n} \subset \sigma$. Then, we randomize the connection based on randomness parameter. We further add n random connections between $\sigma(A)$ and $\sigma(B)$.

This setup yields more complicated structures than the previous setup since each of $\{a.X\}_{a \in \sigma(A)}$ and $\{b.Y\}_{b \in \sigma(B)}$ is made of dependent observations and A and B are in many-to-many relationships.

Results

For tests for the null hypothesis, we obtained similar results as in the previous section. However, the kernel-based residualization approach (RK-HSIC) shows higher type-I errors than expected when larger contexts are employed. KRCIT performed as desired even with larger contexts. For tests for alternative hypothesis, we report AUPC in Figure 5.7. With properly-sized contexts (hop=1), both residualization-based methods perform well. However, they are sensitive to the choice of contexts — the power of both R-HSIC and RK-HSIC drops. Both these KRCIT methods are relatively weaker than any other tests when dependence between relational variables

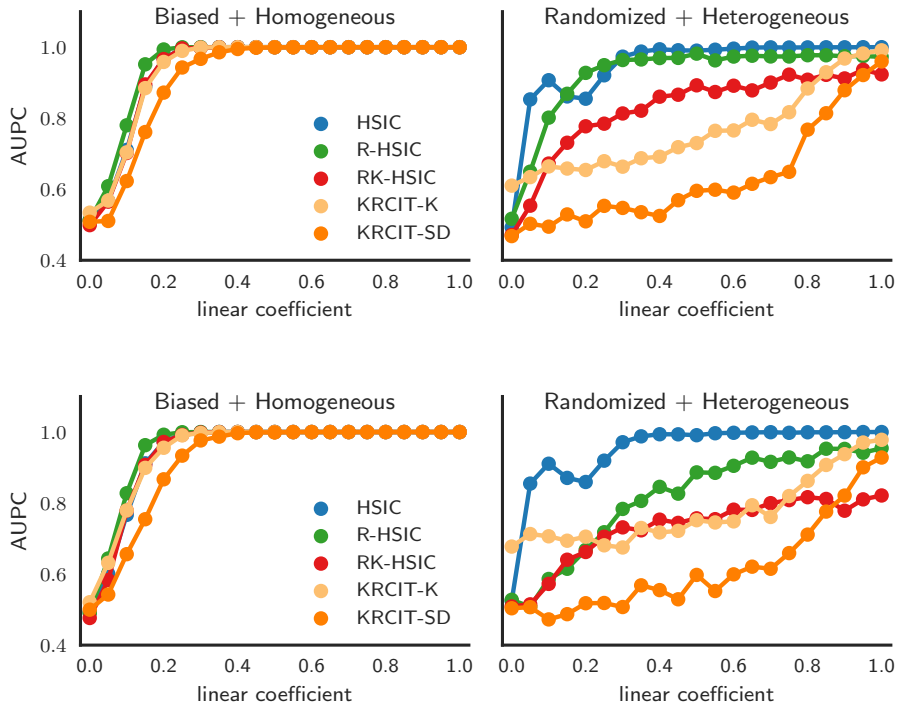


Figure 5.7: AUPCs with hop=1 (top) and hop=2 (bottom) under two different settings for relational skeletons

is not strong enough. However, KRCIT with SDCIT seems robust to the choice of contexts and achieves high AUPC as the dependence increases.

5.5.4 Conditional Tests

We investigated whether KRCIT would be able to discover the causal structure of synthetic relational data generated from an RCM. Thus, we focus on testing 1) relational version of Markov condition, which is essential to learn the undirected causal structure, and 2) conditional dependence, which is critical to infer the orientation of undirected causal relationships. We constructed a set of relational skeletons of 3 entity classes and 3 relationship classes between every pair of entity classes. We controlled for the maximum number of neighbors of the same item class of an item (e.g., a circle item can have at most three square neighbors) and the number of entities per entity class. We generate relational skeletons to exhibit correlation among different relationships. Then, values are generated based on two hypotheses. For the null hypothesis, we generate values, roughly, in a causal order

$X \to Z \to Y$ or $X \leftarrow Z \to Y$. For the alternative hypothesis, a relational data is generated based on a causal order $X \to Z \leftarrow Y$. We test $(U \perp\!\!\!\perp V \mid W)_\sigma$ where U, V, and W associates with X, Y, and Z, respectively.

Relational Schema We have three entity classes A, B, and C, which associates with X, Y, and Z, respectively. There are binary relationship classes for each pair of entity classes, i.e., R_{AB}, R_{AC}, and R_{BC}. All cardinalities are many, hence an entity can have many neighbors of the other entity class.

Relational Skeleton We control the maximum number of neighbors of the same kind. For example, an item of A can have at most k neighbors of B. In other words, $\forall_{a_i \in \sigma(A)} |ne(a_i; \sigma) \cap \sigma(B)| \leq k$. We similarly put restrictions between B and C and between A and C, as well.

We construct relational skeletons where relationships of all three classes (R_{AB}, R_{BC}, and R_{AC}) are correlated. To do so, we adopt the idea of latent space modeling (Hoff et al., 2002). Given n, the number of entities per entity class, we generate n points in $[0,1]^2 \subset \mathbb{R}^2$ for each entity class. Let $\phi(\cdot)$ be the coordinate of an item. Let $\mathbf{D}^{AB}$ be a squared Euclidean distance matrix where $\left(\mathbf{D}^{AB}\right)_{i,j} := \|\phi(a_i) - \phi(b_i)\|_2^2$. Then, a kernel matrix $\mathbf{K}^{AB}$ is $\left(\mathbf{K}^{AB}\right)_{i,j} := \exp\left(-\gamma \cdot \left(\mathbf{D}^{AB}\right)_{i,j}\right)$ where we chose $\gamma := 50$. By normalization, we get a probability matrix $\mathbf{P}^{AB} := \frac{\mathbf{K}^{AB}}{\left(\mathbf{1}^\top \cdot \mathbf{K}^{AB} \cdot \mathbf{1}\right)}$ to (approximately) model $\mathrm{P}((a_i, b_j) \in \sigma) \propto 2\left(\mathbf{P}^{AB}\right)_{i,j}$. With this probability, we sample $nk/2$ edges to form a relational skeleton while satisfying the maximum number of neighbors k. For example, if we limit an item of A can have three neighbors of B, then, there are, on average, 1.5 B neighbors for an item of A. Edges between A and C and between B and C are similarly obtained.

Relational Causal Model We consider three different models: two for conditional independence and one for conditional dependence. For testing null hypothesis, we randomly choose one of the following two models:

$$\begin{aligned} \forall_{a \in \sigma(A)} \quad & a.X := \mu + \epsilon_a \\ \forall_{c \in \sigma(C)} \quad & c.Z := \sum_{a \in ne(c;\sigma) \cap \sigma(A)} a.X + \mu + \epsilon_c \end{aligned}$$

$$\forall_{b \in \sigma(B)} \quad b.Y := \sum_{c \in ne(b;\sigma) \cap \sigma(C)} c.Z + \mu + \epsilon_b$$

and

$$\begin{aligned}
\forall_{c \in \sigma(C)} \quad & c.Z := \mu + \epsilon_c \\
\forall_{a \in \sigma(A)} \quad & a.X := \sum_{c \in ne(a;\sigma) \cap \sigma(C)} c.Z + \mu + \epsilon_a \\
\forall_{b \in \sigma(B)} \quad & b.Y := \sum_{c \in ne(b;\sigma) \cap \sigma(C)} c.Z + \mu + \epsilon_b.
\end{aligned}$$

For testing alternative hypothesis, we use the following model where (roughly speaking) Z is a common effect of X and Y,

$$\begin{aligned}
\forall_{a \in \sigma(A)} \quad & a.X := \mu + \epsilon_a \\
\forall_{b \in \sigma(B)} \quad & b.Y := \mu + \epsilon_b \\
\forall_{c \in \sigma(C)} \quad & c.Z := \sum_{a \in ne(c;\sigma) \cap \sigma(A)} a.X + \sum_{b \in ne(c;\sigma) \cap \sigma(B)} b.Y + \mu + \epsilon_c.
\end{aligned}$$

In all experiments, we set $\mu = 0.3$. We test

$$[B].Y \perp\!\!\!\perp [B, R_{AB}, A].X \mid [B, R_{BC}, C].Z$$

for the null hypothesis and test

$$[C, R_{AC}, A].X \perp\!\!\!\perp [C, R_{BC}, B].Y \mid [C].Z$$

for the alternative hypothesis.

Results

In Figure 5.8, we plot AUPCs and type-I error rates under different sizes of flattened sample, maximum cardinalities of relationships, and the sizes of contexts in terms of hops. The left plots demonstrate that the power of all tests increases as larger sample is used (with max 3 relationships and hop 1). However, KRCIT with KCIT as a base CI test suffers high type-I error rates. The center plots depict the negative effect of more complex structure on both power and type-I errors. This implies the general approach to handling multiple values (i.e., using an R-convolution

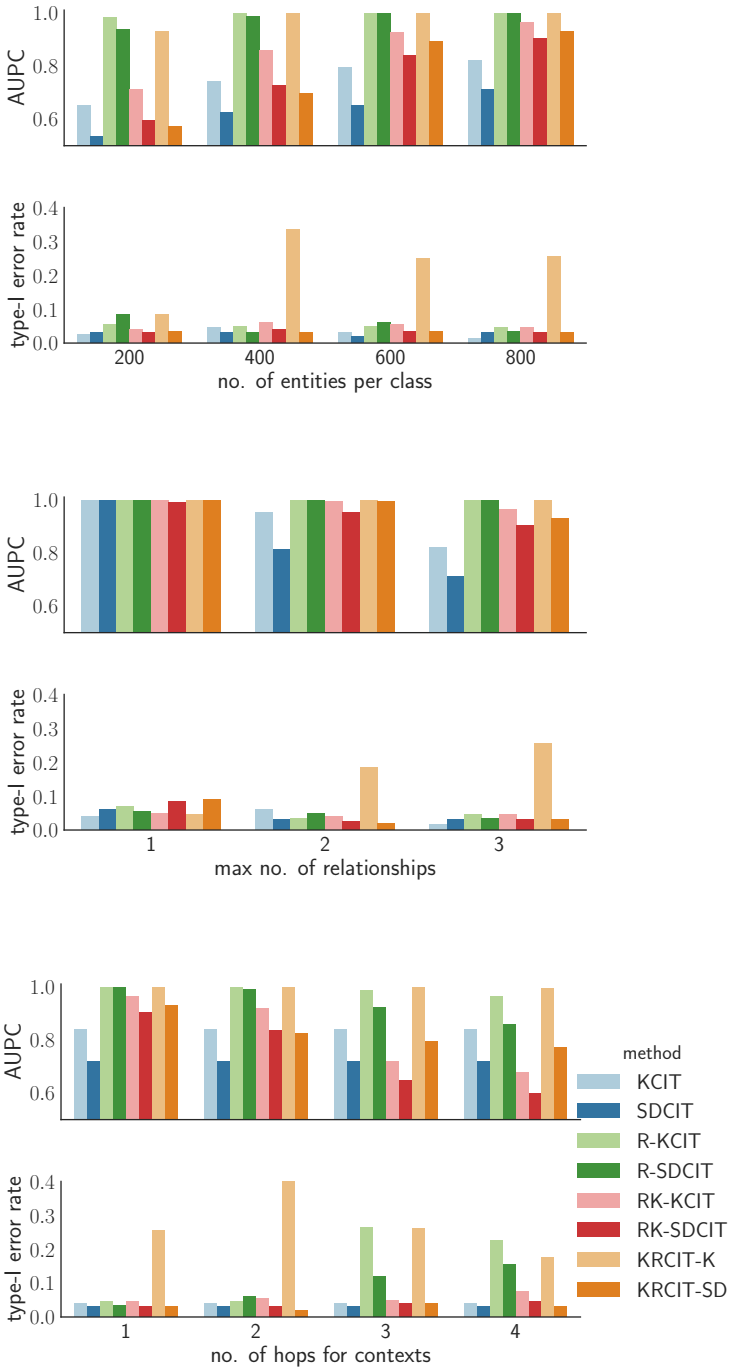


Figure 5.8: AUPCs and type-I error rates of various tests with relational skeletons generated with different settings and tests employed different size of subgraphs as contexts

kernel) should properly modified when a large number of relationships is involved (e.g., friends). Unlike previous experiments with unconditional RCI tests, it is surprising to observe that the power of naive tests decreases as underlying structure is more complex. Finally, contexts larger than *necessary* makes some tests weaker and less-calibrated (right plots). However, we observed that contexts with hop set to 4 are similar to each other. That is, the kernel matrix obtained by applying shortest-path kernel on contexts contains similar values and can not clearly inform heterogeneity among random variables.

Overall, the naive tests that do not account for relational structure performed well for the null hypothesis yielding type-I error rate around 0.05 since, in our generative model, Z 'values' provide all the necessary information to infer Y. However, they showed weak power compared to others in general. Both residualization approaches perform very well when proper contexts are employed. KRCIT performs very differently depending on the choice of base CI test. KCIT, which also uses residualization as an internal mechanism to handle conditionals, seems to have problems dealing with G, a conditioning variable playing a role of *id*. KRCIT with SDCIT is, in general, a good choice since it provides a reasonable power with a precise control of type-I error rates.

5.6 Discussion

Maier et al. (2013a) considered relational d-separation, a problem closely related to RCI. However, they relied on a traditional CI test by simply flattening and aggregating relational data (i.e., average) without incorporating structural information. As we have shown, such an approach biases the results of independence tests (Section 5.5.2). An independence test on two non-iid data sets is addressed by Zhang et al. (2009), who considered generalizing HSIC to the setting of structured and interdependent observations. However, their work focused on explicitly known CI structures which can be represented as undirected graphical models utilizing factorization provided by the exponential family.

In this chapter, we have defined CI in the relational setting and provided an effective approach to testing for RCI. Our definition makes use of a definition of a relational variable that subsumes the notions of *slot chains* in probabilistic relational models (Friedman et al., 1999), *relational path* in relational causal models

with *bridge burning semantics* (Maier et al., 2013a) and *path semantics* (Lee and Honavar, 2016a), and *first order expression* in DAPER (Heckerman et al., 2007).

Structure formation depending on attributes of entities is usually assumed in many statistical network models (e.g., latent space model, (Hoff et al., 2002)). However, existing (statistical) relational models which focus on modeling a joint distribution of values have ignored such phenomenon. For instance, recent work on estimating a causal effect on a relational domain (Arbour et al., 2016a) and finding directions in relational dependency (Arbour et al., 2016b) did not consider effects on network, that is, their simulation studies are based on generated values given a relational structure. It becomes more important to address effects on structure especially when "causal" questions are concerned.

5.7 Concluding Remarks

In this chapter, we defined relational conditional independence (RCI), the generalization of CI to a relational setting with the language of Entity-Relationship model. We proposed kernel RCI test (KRCIT), a principled and general design of RCI test which reduces bias caused by an underlying relational structure. We empirically demonstrated benefits of KRCIT compared to naive CI tests on simulated relational data.

Some directions for future work include: improving KRCIT by employing appropriately designed graph kernels and optimizing the kernel parameters; a more comprehensive experimental study of KRCIT using real-world relational data; investigating a way to incorporate network analysis before performing RCI test to guide the design of kernels; and applying RCI to discover causal relationships in relational domains (Maier et al., 2013a; Lee and Honavar, 2016a).

Chapter 6
Robust Relational Causal Discovery from Relational Data

We presented in Chapter 4 how we can learn the structure of a relational causal model soundly and completely in a sense that there is no other constraint-based methods that can correctly orient more edges than the given algorithm in the presence of a relational conditional independence oracle for the given RCM. In addition, Chapter 5 discusses difficulties of testing relational conditional independence among relational variables due to non-iidness (arose by an underlying relational structure). This chapter focuses on learning the structure of an underlying RCM from relational data (a snapshot of item attributes of a relational skeleton). Instead of naively plugging in KRCIT (Algorithm 10) on RpCD (Algorithm 9) as a choice of relational CI test, we propose a more principled approach towards a robust and practical relational causal discovery algorithm.

6.1 Introduction

Development of a learning algorithm with an independence oracle is important in a theoretical aspect. Learning algorithms designed for a CBN based on the presence of an independence oracle generally work well when the given data is large enough. However, there is a notable difference between CBNs and RCMs in terms of their independence oracles — the oracle for learning an RCM is defined to cover aspects of the model (i.e., $\Sigma_{\mathbb{S}}$ and $\left\{\mathcal{G}_{\sigma}^{\mathcal{M}}\right\}_{\sigma\in\Sigma_{\mathbb{S}}}$ as a whole) not some relational data (σ and $\mathcal{G}_{\sigma}^{\mathcal{M}}$) generated by the RCM. Furthermore, the way relational conditional independence is defined as is cannot easily be answered by an RCI test method given that a relational data is a single sample generated from a ground graph

corresponding to an RCM given a relational skeleton.

Other than the problem (the lack of existing *practical* RCI tests), the given relational data can be noisy (e.g., the small size of relational skeleton lacking structural diversity, or incorrect assumptions). Thus, testing RCI can be error-prone. An error at an early stage of learning algorithm may have negative consequences in its resulting structure, which is called *instability* property (Spirtes et al., 2000). Therefore, we propose a practical learning algorithm that is robust to such violations of assumptions and errors in RCI tests. This necessitates mechanisms for (i) identifying potentially wrong or contradictory relational conditional independence statements and (ii) correcting such errors.

Typically, learning algorithms for CBN (and RCM) consist of following steps:

1. Enumerating candidate dependencies,
2. Performing (R)CI tests to remove spurious dependencies,
3. Performing (R)CI tests to orient dependencies,
4. Orienting based on known constraints (e.g., acyclicity, known non-colliders, etc)

For the step 1, relational models, in general, may have a huge (or infinitely many) number of candidate dependencies. For learning structure for relational models, greedy-iterative deepening search or exhaustive search within fixed distance neighborhood were applied. We will follow RCM literature, which assumes that the maximum hop length of relational dependency is known *a priori*. For a practical purpose, this can be viewed as a user-set parameter.

There are several studies for CBN, which address issues arose during steps 2, 3, and 4. Dash and Druzdzel (1999) proposed an algorithm that tries CI tests based on different orders of variables, and combine results based on a Bayesian metric. Spirtes et al. (2000) provides a mechanism that would remove 'weakest' dependencies at an earlier stage. Abellán et al. (2006) proposed a Bayesian test to replace Chi-square test, and took *minimum cut set* into account. Ramsey et al. (2006) relaxed assumptions about *faithfulness* and proposed a method to check whether 'orientation' (step 3) among variables can be *faithfully* retrieved from CI tests. Cano et al. (2008) scored edges of a Bayesian network according to a Bayesian metric, and took edge scores into account as a means to ordering CI tests, which

is similar to K2 greedy algorithm (Cooper and Herskovits, 1992). Bromberg and Margaritis (2009) resolved errors in conditional independence tests and inconsistent knowledge through *argumentation* framework, an instance of defeasible logic. They interpreted properties of the CI relations (Dawid, 1979; Pearl, 2000) as *integrity constraints* that can fix certain type of errors in test outcomes. Recently, causal discovery algorithms take advantage of general-purpose Boolean satisfiability solvers (Triantafillou et al., 2010; Hyttinen et al., 2013; Triantafillou and Tsamardinos, 2015; Borboudakis and Tsamardinos, 2016; Magliacane et al., 2016; Zhalama et al., 2017).

We are especially interested in identifying relational conditional independence queries that can be reliably answered from a given relational data. Further, we want the learning algorithm *accurate* and *efficient* in terms of the number of RCI queries needs to be asked to orient as many as possible discovered undirected dependencies.

6.2 Generalization of Markov Condition and Faithfulness to Relational Setting

We start by making a connection between a causal structure and a probability distribution modeled by the causal structure. These are conditions (or the variants of) on which the validity of causal discovery algorithms relies.

6.2.1 Causal Markov Condition

(Causal) Markov condition asserts that independence implied in a graphical model holds true in the probability distribution to be modeled. We reproduce the definition given by Ramsey et al. (2006).

Definition 41 (Causal Markov Condition)**.** Given a set of variables whose causal structure can be represented by a DAG $\mathcal{G}$, every variable is probabilistically independent of its non-effects (non-descendants in $\mathcal{G}$) conditional on its direct causes (parents in $\mathcal{G}$).

The condition is about the class of distributions admitted by such a graphical structure. The condition is irrelevant to the choice of conditional independence tests nor the given finite sample.

6.2.1.1 Generalization of Markov Condition to Relational Setting

Next, we put the relational version of Markov condition given by (Maier, 2014) adjusting some notations.

Definition 42 (Relational Markov Condition (Maier, 2014))**.** Let U be a relational variable for perspective $B \in \mathbf{E} \cup \mathbf{R}$ defined over relational schema $\mathcal{S}$. Let $nd(U)$ be the non-descendant variables of U, and let $pa(U)$ be the set of parent variables of U. Then, for relational model $\mathcal{M}_{\boldsymbol{\Theta}}$, $P(U \mid nd(U), pa(U)) = P(U \mid pa(U))$ if and only if $\forall_{u \in U|_b^\sigma} P(u \mid nd(u), pa(u)) = P(u \mid pa(u))$ in parametrized ground graph $\mathcal{G}_\sigma^{\mathcal{M}\Theta}$ for all skeletons $\sigma \in \Sigma_\mathcal{S}$ and for all $b \in \sigma(B)$.

The definition given by Maier (2014) is problematic for several reasons: i) $nd(U)$, non-descendant variables of U, is not explicitly defined; ii) the probability distribution of a relational variable is not clearly defined except for the case of conditional distribution, $P(U \mid pa(U))$, where U is a canonical relational variable; and iii) the condition is given as a necessary and sufficient condition rather than a sufficient condition. We first define *non-descendants* of a canonical relational variable.

Definition 43. Given a relational schema, a relational variable $Q.Z$ is *non-descendant* of V_X *where* $Q.Z \neq V_X$ *if and only if*

$$\forall_{\sigma \in \Sigma_\mathcal{S}} \forall_{b \in \sigma(I_X)} Q.Z|_b^\sigma \cap de\left(b.X; \mathcal{G}_\sigma^\mathcal{M}\right) = \emptyset.$$

The above definition provides a declarative characterization of non-descendants. Apparently, any relational variables of the form $Q.Z$ where $X \not\prec_\pi Z$ (i.e., Z is a non-descendant of X in the class dependency graph $\mathcal{G}_\mathbf{A}^\mathcal{M}$ where $Z \neq X$) will be non-descendants of V_X. Further, for any $Z \in de\left(X; \mathcal{G}_\mathbf{A}^\mathcal{M}\right)$, one can observe that there exists at least one relational variable of the form $Q.Z$ which is not a non-descendant of V_X where Q can be constructed by sequentially applying newextend operations on relational dependencies defined along with attribute classes from X to Z.

Given the above definition, we rewrite relational Markov condition with causality flavor (where parents are considered as direct causes) while avoiding representation of probability for "relational variables":

Definition 44 (Relational Causal Markov Condition)**.** Let $U = V_X$ be a canonical relational variable for perspective $B \in \mathbf{E} \cup \mathbf{R}$ defined over relational schema $\mathcal{S}$. An

RCM $\mathcal{M} = \langle \mathcal{S}, \mathbf{D}, \mathbf{\Theta} \rangle$ satisfies

$$\forall_{\sigma \in \Sigma_{\mathcal{S}}} \forall_{b \in \sigma(B)} \mathrm{P}\left(b.X \mid \mathbf{W}|_b^{\sigma}, pa\left(b.X; \mathcal{G}_{\sigma}^{\mathcal{M}}\right)\right) = \mathrm{P}\left(b.X \mid pa\left(b.X; \mathcal{G}_{\sigma}^{\mathcal{M}}\right)\right)$$

if $\mathbf{W}$ is a set of non-descendant relational variables of V_X.

The definition corresponds to the case of relational conditional independence $\mathbf{W} \perp\!\!\!\perp V_X \mid pa\left(V_X; \mathcal{M}\right)$.

6.2.2 Faithfulness

Faithfulness condition imposes (conditional) "dependency" among variables that can be inferred from d-connection among corresponding vertices in a causal Bayesian network.

Definition 45 (Causal Faithfulness Condition (Ramsey et al., 2006))**.** Given a set of variables whose causal structure can be represented by a DAG $\mathcal{G}$, no conditional independence holds unless entailed by the Causal Markov Condition.

There are several weaker faithfulness conditions. Among those, there are two important faithfulness conditions — adjacency-faithfulness and orientation-faithfulness — in identifying the existence of a dependency (hence, identification of undirected dependencies) and the orientation of such undirected dependencies.

Definition 46 (Adjacency-Faithfulness (Ramsey et al., 2006))**.** Given a set of variables $\mathbf{V}$ whose causal structure can be represented by a DAG G, if two variables X, Y are adjacent in G, then they are dependent conditional on any subset of $\mathbf{V} \setminus \{X, Y\}$.

Definition 47 (Orientation-Faithfulness (Ramsey et al., 2006))**.** Given a set of variables $\mathbf{V}$ whose causal structure can be represented by a DAG $\mathcal{G}$, let $\langle X, Y, Z \rangle$ be any unshielded triple in $\mathcal{G}$.

- (O1) if $X \rightarrow Y \leftarrow Z$, then X and Z are dependent given any subset of $\mathbf{V} \setminus \{X, Z\}$ that contains Y;
- (O2) otherwise, X and Z are dependent conditional on any subset of $\mathbf{V} \setminus \{X, Z\}$ that does not contain Y.

6.2.2.1 Generalization of Faithfulness to Relational Setting

We consider generalizing adjacency-faithfulness and orientation-faithfulness to a relational setting. Both faithfulness conditions for CBNs guarantee that CI tests will correctly capture the dependence, in the limit. Relational data we focus on is a snapshot of relational skeleton σ, which looks like a single sample generated from one big causal Bayesian network called a ground graph $\mathcal{G}_\sigma^{\mathcal{M}}$. In general, it is not obvious what properties an infinitely large relational skeleton would possess.

Since a relational causal model is a template for (often infinitely) many CBNs, we define multiple relational faithfulness conditions as follows.

Definition 48 (Relational Faithfulness)**.** Given an RCM $\mathcal{M} := \langle \mathcal{S}, \mathbf{D}, \mathbf{\Theta} \rangle$, for every ground graph $\mathcal{G}_\sigma^{\mathcal{M}}$, probability distribution of item attributes induced by $\mathbf{\Theta}$ satisfies faithfulness.

Definition 49 (Relational Adjacency-Faithfulness)**.** Given an RCM $\mathcal{M} := \langle \mathcal{S}, \mathbf{D}, \mathbf{\Theta} \rangle$, for every ground graph $\mathcal{G}_\sigma^{\mathcal{M}}$, probability distribution of item attributes induced by $\mathbf{\Theta}$ satisfies adjacency-faithfulness.

Definition 50 (Relational Orientation-Faithfulness)**.** Given an RCM $\mathcal{M} = \langle \mathcal{S}, \mathbf{D}, \mathbf{\Theta} \rangle$, for every ground graph $\mathcal{G}_\sigma^{\mathcal{M}}$, probability distribution of item attributes induced by $\mathbf{\Theta}$ satisfies orientation-faithfulness.

This set of definitions claims that if one can get infinite samples from the resulting ground graph, the Markov equivalence class of the ground graph can be reconstructed. There are apparent limitations applying relational version of faithfulness in an actual learning task since we will not, in general, access to a lot of samples with a fixed relational structure. If it is the case, the learning of an RCM is simply learning a big Bayesian network with additional constraints.

Even when we reconstruct the Markov equivalence class for a ground graph, learning the unique structure of an RCM depends on the underlying relational skeleton. Given a finite set of candidate undirected relational dependencies, there can be multiple subsets of undirected relational dependencies that yield the same undirected structure (i.e., *skeleton* in CBN literature) of a ground graph. Given a sufficiently large randomly generated relational skeleton and a small number of maximum hops for candidate relational dependencies, we would be usually able to uniquely determine a set of (undirected) relational dependencies. One

can measure whether two different relational variables are well distinguished, e.g., $|\{i \in \sigma(B) \mid P|_i^\sigma = Q|_i^\sigma\}| / |\sigma(B)|$. In this dissertation, we presume that such measure is not close enough to 1 for any pair of relational variables so that every relational variables is sufficiently distinguished from others.

6.3 Traditional CI Test for RCM Structure Learning

In the previous chapter, KRCIT is proposed to reduce possible *false positives* in testing RCI among relational variables by taking advantage of relational structure relevant to the aspects of how values of item attributes are determined. However, many practical questions are remained unanswered: How can we determine the relational structure relevant to an item attribute?; How can we choose a graph kernel (or the combination of multiple graphical kernels) and its parameters?; What are sufficient conditions to skip the use of a graph kernel?

Fortunately, there is a class of relational conditional independence queries, which can be *reliably* answered through a flattened representation of relational data without adjusting relational contexts. Consider, for an example, three sequences of (without guaranteeing iid among Xs, Ys, and Zs) random variables $\{X_i\}_{i=1}^n$, $\{Y_i\}_{i=1}^n$ and $\{Z_i\}_{i=1}^n$ where we want to examine $\forall_{1\leq i\leq n} X_i \perp\!\!\!\perp Y_i \mid Z_i$. Assume the following condition:

$$\forall_{1\leq i\neq j\leq n} (Y_i \perp\!\!\!\perp Y_j \mid Z_i) \wedge (Y_i|Z_i \stackrel{d}{=} Y_j|Z_j)$$

This demonstrates that Y is iid per conditionals. Then, $\forall_{1\leq i\leq n} X_i \perp\!\!\!\perp Y_i \mid Z_i$ can be reliably tested treating X, Y, and Z as random variables generated from some P_{xyz} (with large enough n and non-unique Z values). In general, $\forall_{1\leq i\leq n} X_i \not\perp\!\!\!\perp Y_i \mid Z_i$ will yield $X \not\perp\!\!\!\perp Y \mid Z$ with a traditional CI test unless if the set of random variables are adversarially composed in a way to cancel out individual dependence for each z value.

We adopt the idea for relational CI tests. We denote a CI query (or a CI statement) on a flattened representation corresponding to an RCI query $U \perp\!\!\!\perp V \mid \mathbf{W}$ by $(U \perp\!\!\!\perp V \mid \mathbf{W})_\sigma$, which is based on a data with $|\mathbf{W}| + 2$ columns corresponding to relational variables in the query, and with rows corresponding to base items

$\sigma(B)$ so that row-i is

$$\left\langle U|_i^\sigma,\ V|_i^\sigma,\ W^{(1)}|_i^\sigma,\ \ldots,\ W^{(m)}|_i^\sigma \right\rangle$$

where $\mathbf{W} = \left\{W^{(j)}\right\}_{1\leq j\leq m}$. We will develop *faithfulness-like* conditions while illustrating robust structure learning algorithm for an RCM with RCI tests (and variants of) on flattened representations.

6.4 Robust Structure Learning Algorithm for RCM

We introduce a structure learning algorithm for an RCM given a relational data. The algorithm is divided into three phases where i) Phase-I determines the existence of relational dependencies; ii) Phase-II performs RCI tests to collect information about the orientation of relational dependencies; and iii) Phase-III synthesizes given information to finalize the orientation of relational dependencies. Note that Phase-III is often considered as the last step of Phase-II.

6.4.1 Phase I: Testing for Adjacencies

As mentioned, we assume that h, the maximum hop length of relational dependencies, is given *a priori*. Hence, we can enumerate a set of *candidate* relational dependencies as shown in Algorithm 9. Removing spurious relational dependencies while retaining actual relational dependencies involves following two conditions:

$$\forall_{P.X\in(nd(V_Y;\mathcal{M})\setminus pa(V_Y;\mathcal{M}))\cap\mathbf{V}_{\mathbb{S},I_Y,h}} \left(P.X \perp\!\!\!\perp V_Y \mid pa\left(V_Y;\mathcal{M}\right)\right)_\sigma$$

and

$$\forall_{P.X\in adj(V_Y;\mathcal{M})}\forall_{\mathbf{W}\subseteq\mathbf{V}_{\mathbb{S},I_Y,h}\setminus\{P.X,V_Y\}} \left(P.X \not\perp\!\!\!\perp V_Y \mid \mathbf{W}\right)_\sigma \tag{6.1}$$

where $\mathbf{V}_{\mathbb{S},B,h}$ is the set of relational variables whose perspective is B and the hop length of its relational path is less than or equal to h. The Phase-I of practical learning algorithm will be similar to that of RpCD in many aspects. However, we will introduce additional features to make the algorithm robust. These include (i) order-independence, (ii) additional aggregation-based test, (iii) recovering removed relational dependencies based on the violation of relational causal Markov condition (RCMC), and (iv) further refining recovered relational dependencies.

6.4.1.1 Order-Independence

Removal of spurious dependencies at one point of Phase-I affects the set of RCI queries to be performed later. For example, if a given genuine relational dependency is *weak*, it is likely that such a dependency might be removed. In such cases, the removed adjacent relational variables of a canonical relational variable (i.e., the effect) will not be used as an element of possible separating set between some relational variable and the canonical relational variable in later steps. Such *false negatives* might lead to *false positives*.

Colombo and Maathuis (2014) studied the phenomenon from both theoretical and practical aspects. They found out that the order dependence in PC algorithm (and other causal structure learning algorithms) can be avoided by delayed removal, and showed that the performance is comparable to averaging models learned with different orderings, which requires higher computational resource. The principle by Colombo and Maathuis (2014) can be directly applicable to our learning algorithm.

6.4.1.2 Aggregation for Weak Dependence

There exists a notable difference between an edge in a CBN and a relational dependency in an RCM with respect to the test for adjacencies — there can be two different tests for a single adjacency. Note that one can split the condition (Equation 6.1) into two different conditions,

$$\forall_{P.X \in pa(V_Y;\mathcal{M})} \forall_{\mathbf{W} \subseteq \mathbf{V}_{\mathrm{S},I_Y,h} \setminus \{P.X, V_Y\}} \left(P.X \not\perp\!\!\!\perp V_Y \mid \mathbf{W}\right)_\sigma$$

and

$$\forall_{P.X \in ch(V_Y;\mathcal{M})} \forall_{\mathbf{W} \subseteq \mathbf{V}_{\mathrm{S},I_Y,h} \setminus \{P.X, V_Y\}} \left(P.X \not\perp\!\!\!\perp V_Y \mid \mathbf{W}\right)_\sigma$$

where each can be viewed as 'parental' and 'non-parental' (pseudo-)faithfulness.

Unlike traditional adjacency-faithfulness, relational variables as a parent or as a child are quite different. Consider, for an example, a relational dependency, $[\mathsf{P}, \mathsf{D}, \mathsf{E}]\,.\mathsf{Competence} \rightarrow [\mathsf{P}]\,.\mathsf{Success}$ from the company example. The dependency between $[\mathsf{E}, \mathsf{D}, \mathsf{P}]\,.\mathsf{Success}$ and $[\mathsf{E}]\,.\mathsf{Competence}$ can be substantially weaker than the dependency between $[\mathsf{P}, \mathsf{D}, \mathsf{E}]\,.\mathsf{Competence}$ and $[\mathsf{P}]\,.\mathsf{Success}$ since there is a set of multiple employees' competence that affects $[\mathsf{E}, \mathsf{D}, \mathsf{P}]\,.\mathsf{Success}$ but not considered. Two different tests for the same (undirected) relational dependencies yield different

flattened data sets. Further, the functional relationships between $[\mathsf{E}, \mathsf{D}, \mathsf{P}]$ .Success and $[\mathsf{E}]$ .Competence and between $[\mathsf{P}, \mathsf{D}, \mathsf{E}]$ .Competence and $[\mathsf{P}]$ .Success are different.

One can apply an aggregating function (e.g., mode, average, median, etc), $f : 2^{\mathcal{X}} \to \mathcal{Z}$ where $\mathcal{Z}$ can be $\mathcal{X}$, $\mathbb{R}$, or other domain, on $P.X \in adj\,(V_Y; \mathcal{M})$ when the given RCI test outputs $(P.X \perp\!\!\!\perp V_Y \mid \mathbf{W})_\sigma$. Then, an additional test can be conducted (slightly abusing notation),

$$(f\,(P.X) \perp\!\!\!\perp V_Y \mid \mathbf{W})_\sigma$$

where each $P.X|_i^\sigma$ is replaced by $f\,(P.X|_i^\sigma)$. Such aggregation, which maps the given set of values to a single value, does *not* introduce additional dependence. However, in reality, this procedure may help overcome weak RCI tests as the mapping reduces not only dimensionality but also variances due to noises.

6.4.1.3 Detection of Violation of Relational Causal Markov Condition

Given a finite sample, RCI tests often yield false negatives and false positives. In general, we can control a false positive rate with α (e.g., $\alpha = 0.05$). Further, RCI tests on a flattened representation can be reliably used for RCMC-related RCI queries. However, false negatives due to weak dependence are difficult to be avoided in general.

During Phase-I, some of RCI queries involve conditionals whose subset is not a subset of parents of the canonical relational variable in the query. That is,

$$(P.X \perp\!\!\!\perp V_Y \mid \mathbf{S})_\sigma$$

where $\mathbf{S} \not\subseteq pa\,(V_Y; \mathcal{M}) \subseteq adj\,(V_Y; \mathcal{M})$. There is no reason to dismiss such RCI results. Nevertheless, if $P.X \notin adj\,(V_Y; \mathcal{M})$, then the following condition will hold true under RCMC:

$$\exists_{\mathbf{S} \subseteq pa(V_Y;\mathcal{M})} P.X \perp\!\!\!\perp V_Y \mid \mathbf{S} \text{ or } \exists_{\mathbf{S} \subseteq pa(V_X;\mathcal{M})} \tilde{P}.Y \perp\!\!\!\perp V_X \mid \mathbf{S}. \tag{6.2}$$

This suggests that we might be able to detect and restore some of false negatives. Unfortunately, we have no access to the true model $\mathcal{M}$. But we might consider

performing the following tests given the intermediate undirected RCM $\mathcal{M}'$:

$$\exists_{\mathbf{S}\subseteq adj(V_X;\mathcal{M}')}\left(P.X \perp\!\!\!\perp V_Y \mid \mathbf{S}\right)_\sigma \text{ or } \exists_{\mathbf{S}\subseteq adj(V_Y;\mathcal{M}')}(\tilde{P}.Y \perp\!\!\!\perp V_X \mid \mathbf{S})_\sigma.$$

Such a 'relaxed' test is valid when $pa\left(V_X;\mathcal{M}\right) \subseteq adj\left(V_X;\mathcal{M}'\right)$ for every canonical relational variable V_X. If there exists a pair $P.X$ and V_Y, which cannot be separated by a subset of adjacencies of V_Y (or other way around), then it is probable that, under RCMC, there are false negatives:

1. Removal of $P.X - V_Y$ was false negative; or
2. i) There exists a false negative $Q.Z - V_Y$ where $Q.Z \in pa\left(V_Y;\mathcal{M}\right) \backslash adj\left(V_Y;\mathcal{M}'\right)$ and $Y \not\perp_\pi X$ or ii) there exists a false negative $Q'.Z - V_X$ where $Q'.Z \in pa\left(V_X;\mathcal{M}\right) \setminus adj\left(V_X;\mathcal{M}'\right)$ and $X \not\perp_\pi Y$.

If we restore $P.X - V_Y$ for all such cases, subsequently, it would be possible to restore $Q.Z - V_Y$ as well.

6.4.1.4 Additional Refinement of the Recovered

Once false negative candidates are identified, we can further refine again to discover possible false positives among recovered relational dependencies. With non-RCMC related relational dependencies recovered, some of canonical relational variables will be connected with more relational variables that can be served as a separating set. Consider the following scenario where there are two recovered dependencies $P.X - V_Y$ and $Q.Z - V_Y$. Let $\mathcal{M}'$ be an intermediate RCM after recovery. If

$$\exists_{\mathbf{S}\subseteq adj(V_Y;\mathcal{M}')\setminus\{Q.Z\}}\left(Q.Z \perp\!\!\!\perp V_Y \mid \mathbf{S}\right)_\sigma$$

then, it is possible that the 'recovered' $Q.Z - V_Y$ is a false positive (n.b. it was a true negative before being recovered.) To avoid recovered relational variables removing each other, we can build, for each canonical relational variable, a directed (possibly cyclic) graph of relational variables connected to the canonical relational variable where an edge $P.X \to Q.Z$ in the graph represents whether there exists a separating set $\mathbf{S} \subseteq adj\left(V_Y;\mathcal{M}'\right) \setminus \{Q.Z\}$ containing $P.X$ that separates $Q.Z$ and V_Y. Then, relational dependencies between V_Y and relational variables with at least one arrowhead without a tail (i.e., with a positive indegree and a 0-outdegree)

are considered as false positives (i.e., falsely recovered), and will be removed again from $\mathcal{M}'$. We describe the pseudocode for Phase-I in Algorithm 11 where aggregation-based tests are implicit — an RCI test $(U \perp\!\!\!\perp V_X \mid \mathbf{W})_\sigma$ needs to be read as $(U \perp\!\!\!\perp V_X \mid \mathbf{W})_\sigma \wedge (f(U) \perp\!\!\!\perp V_X \mid \mathbf{W})_\sigma$.

Algorithm 11 Phase-I with implicit additional aggregation-based RCI tests

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

initialize $\mathbf{D}$ with candidate relational dependencies up to h hops.
initialize an undirected RCM $\mathcal{M}'$ with undirected $\mathbf{D}$.
$\ell := 0$
repeat
  $\Delta := \emptyset$
  for every ordered pair $(P.Y, V_X)$ **such that** $P.Y \in adj(V_X; \mathcal{M}')$ **do**
    for every $\mathbf{S} \subseteq adj(V_X; \mathcal{G}) \setminus \{P.Y\}$ **such that** $|\mathbf{S}| = \ell$ **do**
      if $(V_X \perp\!\!\!\perp P.Y \mid \mathbf{S})_\sigma$ **then**
        $\Delta := \Delta \cup \{P.Y - V_X\}$
        break
  remove edges in Δ from $\mathcal{M}'$
  $\ell := \ell + 1$
until $|adj(V_X; \mathcal{M}')| - 1 < \ell$ for every $X \in \mathbf{A}$

$\Delta := \emptyset$
for every removed undirected dependency $P.Y - V_X$ **do**
  if $\forall_{\mathbf{S} \subseteq adj(V_X; \mathcal{M}')}(P.Y \not\perp\!\!\!\perp V_X \mid \mathbf{S})_\sigma$ **and** $\forall_{\mathbf{S} \subseteq adj(V_Y; \mathcal{M}')}(\tilde{P}.X \not\perp\!\!\!\perp V_Y \mid \mathbf{S})_\sigma$ **then**
    $\Delta := \Delta \cup \{P.Y - V_X\}$
add edges in Δ to $\mathcal{M}'$

$\Lambda := \emptyset$
for every canonical relational variable V_Y **do**
  Let $\mathcal{H}$ be an empty directed graph
  for $Q.Z - V_Y \in \Delta$ **do**
    if $\exists_{\mathbf{S} \subseteq adj(V_Y; \mathcal{M}') \setminus \{Q.Z\}} (Q.Z \perp\!\!\!\perp V_Y \mid \mathbf{S})_\sigma$ **then**
      add $\{P.X \to Q.Z \mid P.X \in \mathbf{S}\}$ to $\mathcal{H}$
  Recursively remove a node T with a positive indegree and a zero outdegree from $\mathcal{H}$, and $\Lambda := \Lambda \cup \{T - V_Y\}$
remove Λ from $\mathcal{M}'$

6.4.2 Phase II: Testing for Orientations

With undirected relational dependencies identified by Phase-I, the next phases orient them with RCI tests and known constraints. We describe separately the RCI tests for orientation (Phase-II) and deciding orientations with test results and constraints (Phase-III).

In the case of causal Bayesian networks, the orientation-faithfulness provides a theoretical guarantee to identify unshielded colliders, i.e., $X \rightarrow Y \leftarrow Z$ with X and Z disconnected in $\mathcal{G}$. Under the condition, it is sufficient to find a single separating set $\mathbf{S} \subseteq adj\,(X;\mathcal{G}) \cup adj\,(Z;\mathcal{G})$ given a partially-directed CBN $\mathcal{G}$, and check whether Y is in $\mathbf{S}$ or not. A *conservative* approach is proposed (Ramsey et al., 2006) to detect the violation of orientation-faithfulness. The approach examines all separating sets between X and Z, and can tell the violation of orientation-faithfulness unless $\forall_{\mathbf{S}\in\mathbb{S}_{XZ}} Y \in \mathbf{S}$ or $\forall_{\mathbf{S}\in\mathbb{S}_{XZ}} Y \notin \mathbf{S}$ where $\mathbb{S}_{XZ} = \{\mathbf{S} \mid X \perp\!\!\!\perp Z \mid \mathbf{S},\ \mathbf{S} \subseteq adj\,(X;\mathcal{G}) \cup adj\,(Z;\mathcal{G})\}$ is a set of separating sets between X and Z. In such a case, the directions between X and Y and Y and Z are marked as *undetermined*. A generalized rule is proposed (Colombo and Maathuis, 2014) since the conservative approach not only can be expensive but also leaves many number of edges undirected. The generalized rule sets a threshold $\theta > 0.5$ to decide whether to orient or not, e.g., orient it if $|\{\mathbf{S} \in \mathbb{S}_{XZ} \mid Y \in \mathbf{S}\}| \,/\, |\mathbb{S}_{XZ}| > \theta$ or $|\{\mathbf{S} \in \mathbb{S}_{XZ} \mid Y \notin \mathbf{S}\}| \,/\, |\mathbb{S}_{XZ}| > \theta$ and leave it undetermined, otherwise.

Orientation of relational dependencies have similar problems. We defined a Canonical Unshielded Triple (CUT) as a relational counterpart of an unshielded triple (Section 4.1.1). The techniques for robustly orienting edges in a CBN can effortlessly applicable to orient relational dependencies of an RCM — conservative or threshold approach. However, we highlight the difference between CBN and RCM with respect to orienting dependencies. We first discuss properties of canonical unshielded triples (CUTs) which makes them inappropriate for the task of orientation-related tests with relational data. Then, propose a suite of RCI tests to orient what CUTs would do in theory.

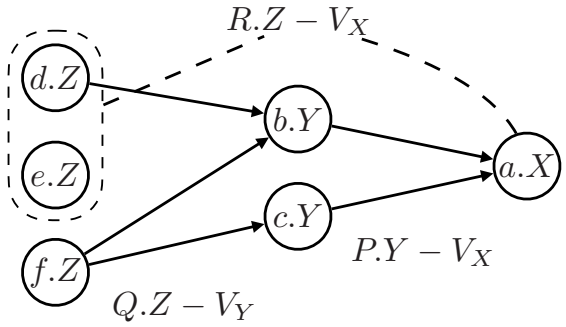


Figure 6.1: An example of a ground graph for $P.Y \rightarrow V_X$, $Q.Z \rightarrow V_Y$, and $R \in \mathsf{newextend}\,(P, Q)$

6.4.2.1 Problems with Orientations with Canonical Unshielded Triples

Every possible unshielded triple in $\left\{\mathcal{G}_\sigma^{\mathcal{M}}\right\}_{\sigma \in \Sigma_{\mathcal{S}}}$ of the form $i.X - j.Y - k.Z$ with $i.X$ and $k.Z$ disconnected corresponds to some canonical unshielded triple (CUT) $\langle V_X, \mathbf{P}.Y, R.Z\rangle$ (or its complementary counterpart $\langle V_Z, \tilde{\mathbf{Q}}.Y, \tilde{R}.X\rangle$) where a separating set $\mathbf{S} \subseteq adj\,(V_X; \mathcal{M})$ between V_X and $R.Z$ exists if $X \not\perp_\pi Z$, and whether $P.Y \in \mathbf{P}.Y$ exists in $\mathbf{S}$ or not can tell whether $X \rightarrow Y \leftarrow Z \in \mathcal{G}_{\mathbf{A}}^{\mathcal{M}}$ or not. There are a few challenges for using CUTs as they are defined for orientation.

1. Given two relational dependencies (without knowing their directions) $P.Y - V_X$ and $Q.Z - V_Y$, $R.Z$ represents one of many possible relationships between $k.Z$ and $i.X$ where $j \in P|_i^\sigma$ and $k \in Q|_j^\sigma$ for *some* $\sigma \in \Sigma_{\mathcal{S}}$. That is, there can be multiple CUTs of the same 3-tuple of attribute classes, e.g., $\langle X, Y, Z\rangle$, even though there exists only a single relational dependency between X and Y and between Y and Z, respectively.

2. $R.Z|_i^\sigma$ can be empty for the given relational skeleton.

3. $R.Z|_i^\sigma$ may contain item attributes that does not form an unshielded triple in a ground graph level (e.g., $k'.Z \in R.Z|_i^\sigma$ does not connect to any item attribute of $P.Y|_i^\sigma$).

4. There may exist some relational dependency $R'.Z - V_X \in \mathcal{M}$ where $R.Z|_i^\sigma = R'.Z|_i^\sigma$ for some base items $i \in \sigma\,(I_X)$.

In sum, the statistical dependency between two relational variables V_X and $R.Z$ can be weak and obtained separating sets between V_X and $R.Z$ can be less reliable to examine whether $P.Y \in \mathbf{P}.Y$ is a collider or not. In Figure 6.1, we illustrate a part of a ground graph corresponding to a CUT $\langle V_X, \mathbf{P}.Y, R.Z \rangle$. One can view that $\{a.X\}$, $\{b.Y, c.Y\}$, and $\{d.Z, e.Z\}$ compose three cells of a row in a flattened representation. There is no clear functional relationship between columns for Y and Z, e.g., $\{b.Y, c.Y\}$ and $\{d.Z, e.Z\}$, as there are two multi-sets with different connections across different rows. The item attribute $e.Z$ is more or less noise with $f.Z$ being treated as unobserved variable. We will overcome CUT-based approach by taking the given relational data into account.

6.4.2.2 Phase-II Outline

Algorithm 12 Outline of Phase-II

Input: σ: relational data, $\mathcal{M}'$: intermediate PRCM, sepset-rule, orientation-rule

1: **for** $P.X \in adj\,(V_Y; \mathcal{M}')$ **for** $Y \in \mathbf{A}$ **such that** $\mathsf{card}\,(P) = \mathsf{many}$, **do**
2: Perform split-RBO tests with sepset-rule, add results to records
3: **for** $P.X, Q.X \in adj\,(V_Y; \mathcal{M}')$ **for** $Y \in \mathbf{A}$ **such that** $P \neq Q$, P and Q are one-to-one **do**
4: Perform pair-RBO tests with sepset-rule, add results to records
5: orientations :=determine orientations tentatively with records & orientation-rule.
6: **for** $P.X, Q.Z \in adj\,(V_Y; \mathcal{M}')$ **for** $Y \in \mathbf{A}$ **such that** $X \neq Z$ **and** ($X \to Y \in$ orientations **or** P is one-to-one) **and** Q is one-to-one **do**
7: **if** $\exists_{R \in \mathsf{newextend}(\tilde{Q}, P)} R.X \notin adj\,(V_Z; \mathcal{M})$ **then**
8: Perform non-RBO tests with sepset-rule, add results to records
9: colliders, non-colliders, undetermined :=determine (non-)colliders for each tested $\langle X, Y, Z \rangle$ with orientation-rule

We outline Phase-II in Algorithm 12. We will use a term 'case' where multiple tests with various sets of conditionals can be conducted to acquire one or more separating sets — a pair of relational dependencies of the same effect compose a single case if they can form a CUT.

In Phase-II, a set of variants of RCI test will be performed to orient what RCI tests on CUTs would do with an independence oracle. We first focus on Relational Bivariate Orientation (RBO) related tests (line 2 and 4, Section 6.4.2.4) which allow us to tentatively determine the orientation between two attribute classes. Then,

non-RBO tests (Section 6.4.2.5) are performed taking an advantage of some of provisional orientations by the previous step (line 8). Then, our choice of orientation rule will determine i) orientations for each pair of attribute classes if a relevant RBO case exists; ii) colliders and non-colliders for a 3-tuple of attribute-class level triples if a relevant non-RBO case exists. Note that they might be undetermined. The information (orientations, colliders, and non-colliders), in the next phase (Phase-III), will be combined to obtain the best partially-oriented RCM in terms of the number of conforming (i.e., non-conflicting) information.

We introduce a mechanism (Section 6.4.2.6) to detect whether a case is *weak* in that determining orientation based on separating sets acquired by subsequent RCI tests on the case is not reliable. Further, the mechanism can be used to determine collider or non-collider at an early stage. The mechanism will be placed before line 2, 4, and 8.

How separating set(s) are acquired for each case is governed by a separating set rule, which is implicit in line 2, 4, and 8. Whether $X \to Y$ or $Y \to X$ when RBO tests are performed between the two attribute classes or whether $X \to Y \leftarrow Z$ or not for non-RBO tests will be decided based on multiple relevant RCI test results between X and Y or among $\langle X, Y, Z \rangle$ with orientation rules. We describe these separating set rules and orientation rules in the following section.

6.4.2.3 Rules for Orientation and Finding Separating Sets

There are two well-known orientation rules given different numbers of supports for contradicting orientations (e.g., $X \to Y$ and $Y \to X$). As said earlier, the 'majority' rule chooses an orientation when the number of supports for the orientation is larger than that of the conflicting counterpart, while 'conservative' rule chooses an orientation when there is no support for the conflicting orientation. The rules 'majority' and 'conservative' can be interpreted as percentage-based rules with 50% and 100% threshold, respectively.

The number of supports can be determined by the results of relevant RCI tests. There are a few ways to perform relevant RCI tests — whether we acquired a *sufficient* number of *reliable* 'separating sets' with respect to some orienting task:

- first: Use a first obtained separating set. The size of a separating set is guaranteed to be minimal (regardless of whether the corresponding RCI test

was correct or not). It is the most efficient but order-dependent. In general, the test itself is unable to correct possible errors unless there are other cases for the same orientation.

- minimal: Use all separating sets of minimal size. For example, if there exists a separating set of size ℓ and no separating set with less than the size of ℓ, it explores all conditionals of size ℓ. It is (quite) efficient and order-independent. Also, a small number of errors can be corrected combined with proper orientation rule (e.g., majority).

- full: Use all separating sets. It is generally inefficient. The rule assumes that every test is equally reliable. Combined with the 'conservative' orientation rule, resulting orientations will be likely precise. Since a large number of conditionals can lead to errors, the rule might not be the best option when the sample size is relatively small when the degree of RCM is large.

Unlike CBNs, where one can find a separating set for X and Z from $adj\,(X; \mathcal{G}') \cup adj\,(Z; \mathcal{G}')$, one needs to deal with two different flattened representations based on two different base item classes where their conditionals might not be compatible. Further, trying to obtain a separating set from one data representation first and the other representation next would not be desirable since there may be no separating set in one representation and increasing the size of candidates for acquiring a separating set can yield less reliable RCI test results. Hence, we design the algorithm for a separating set rule to avoid RCI tests with unnecessarily large conditionals (in the case of 'first' or 'minimal'). A pseudocode is provided in Algorithm 13 where two candidate separating sets are provided.

In addition to the aforementioned rules, one can further consider limiting the number of conditionals (e.g., minimal size + 1). Other measures for the reliability of RCI tests can be included (e.g., the number of rows in a flattened data or p-values or other test statistics from RCI tests). Furthermore, one might omit orientation of 'recovered' relational dependencies in the Phase-I as we already identified that the strength of the dependency is weak.

6.4.2.4 Relational Bivariate Orientation

We describe two types of tests to determine orientation between a pair of attribute classes.

Algorithm 13 Pseudocode for determining separating sets by alternating two sets of conditionals

Input: $\mathbf{W}_1$, $\mathbf{W}_2$, rule, RCI tester:
$\mathbb{S} := \emptyset$
for $0 \leq \ell \leq \max\left(\left|\mathbf{W}_1\right|, \left|\mathbf{W}_2\right|\right)$ **do**
  for s in $\mathcal{U}\left\{(1,2),(2,1)\right\}$ **do**
    for $\mathbf{W}' \in 2_{\ell}^{\mathbf{W}_s}$ **do**
      if RCI test with $\mathbf{W}'$ **then**
        $\mathbb{S} := \mathbb{S} \cup \{\mathbf{W}'\}$
        if rule = first **then**
          break
    if $\mathbb{S} \neq \emptyset$ **and** rule = first **then**
      break
  if $\mathbb{S} \neq \emptyset$ **and** (rule = first or rule = minimal) **then**
    break
return $\mathbb{S}$

6.4.2.4.1 Split Relational Bivariate Orientation A notable difference between the learning the structure of an RCM and that of a CBN is that a single relational dependency can itself form a type of unshielded triples in the case of RCM. Consider a relational dependency $P.X \to V_Y \in \mathbf{D}$. If $\mathsf{card}\left(P\right) = \mathsf{many}$, then for every item attribute $i.Y$ there can be many Xs affecting $i.Y$ (see two left figures in Figure 6.2). In addition, if $\mathsf{card}(\tilde{P}) = \mathsf{many}$, then those Xs might be the cause of Ys other than $i.Y$ (see top left figure). Assuming that the relational skeleton σ contains a sufficient number of items $i \in \sigma\left(I_Y\right)$ such that $\left|P\right|_i^{\sigma}\left| \geq 2\right.$, then, one might be able to perform Relational Bivariate Orientation (RBO) type tests by RCI-like tests between one of many Xs, e.g., $s_i.X$, and the rest of Xs, e.g., $P.X|_i^{\sigma} \setminus \{s_i.X\}$, where $s_i \sim \mathcal{U}\left\{P|_i^{\sigma}\right\}$ (i.e., uniformly selected from the given set) and separating sets are among $adj\left(V_X; \mathcal{M}'\right)$.

The CI test between Xs and the rest of Xs connected through some item attribute whose attribute class is Y can be done by applying a CI test on a data whose row-i can be represented as

$$\left\langle s_i.X,\ P.X|_i^{\sigma} \setminus \{s_i.X\},\ W^{(1)}|_{s_i}^{\sigma}, W^{(2)}|_{s_i}^{\sigma}, \ldots, W^{(m)}|_{s_i}^{\sigma} \right\rangle$$

where $\left\{W^{(1)}, \ldots, W^{(m)}\right\} = adj\left(V_X; \mathcal{M}'\right)$. The data must ensure that the first column is made with unique item attributes.

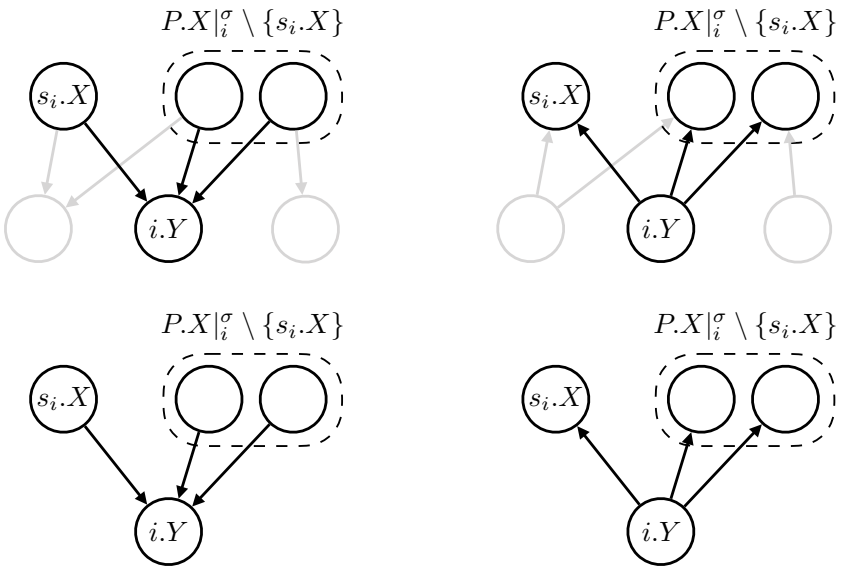


Figure 6.2: Split Relational Bivariate Orientation

Algorithm 14 Pseudocode for Split-RBO

Input: an intermediate PRCM $\mathcal{M}'$

1: $\mathsf{results} = \emptyset$
2: **for** $Y \in \mathbf{A}$ **do**
3: **for** $P.X \in adj\,(V_Y; \mathcal{M}')$ **such that** $\mathsf{card}\,(P) = \mathsf{many}$ **do**
4: $s_i \sim \mathcal{U}\,\{P|_i^\sigma\}$ **for** $i \in \sigma\,(I_Y)$ if $|P|_i^\sigma| \geq 2$
5: $\mathbf{i} := \{i \in \sigma\,(I_Y) \mid |P|_i^\sigma| \geq 2,\ \forall_{j<i} s_j \neq s_i\}$
6: $\mathsf{data} := \left[\left\langle s_i.X,\ P.X|_i^\sigma \setminus \{s_i.X\},\ \ldots W^{(\ell)}|_{s_i}^\sigma, \ldots \right\rangle\right]_{i \in \mathbf{i}}$
7: Perform CI tests between the first two columns with $\mathbf{S} \subseteq adj\,(V_X; \mathcal{M}')$
8: Record CI test results into $\mathsf{results}$.
9: **return** $\mathsf{results}$

A pseudocode is illustrated in Algorithm 14. For each case, select $s_i.X$ randomly (line 4) and find items that yield unique $s_i.X$s. Then, a data is constructed as mentioned above. Multiple CI tests will be performed following a given rule for separating sets. Note that there is no need to seek separating sets *alternatively* (see Algorithm 13). There can be multiple CUTs associated with a *single* relational dependency $P.X - V_Y$ since there are multiple relational paths by extending P and $\tilde{P}$. However, simply changing the base to the center $i.Y$ and freeing from classic 'relational variable' representation, multiple CUTs are *unified* into a single case.

6.4.2.4.2 Pair Relational Bivariate Orientation Once split-RBO is done, there is another type of RBO-related tests where two relational dependencies, e.g., $P.X - V_Y$ and $Q.X - V_Y$, form unshielded triples in a ground graph. We are only interested in cases where both P and Q are one-to-one (i.e., $\mathsf{card}(Q) = \mathsf{card}(\tilde{Q}) = \mathsf{one}$). This guarantees that $P.X|_i^\sigma$ and $Q.X|_i^\sigma$ are both singletons (if not empty). In constructing a data, we ensure that $P.X|_i^\sigma \neq Q.X|_i^\sigma$ which will introduce unintentional dependency not through common ancestors. Let $\{j.X\} = P.X|_i^\sigma$ and $\{k.X\} = Q.X|_i^\sigma$ where $i \in \mathbf{i} = \{i \in \sigma(I_Y) \mid P|_i^\sigma \neq Q|_i^\sigma,\ P|_i^\sigma \neq \emptyset,\ Q|_i^\sigma \neq \emptyset\}$. Then, row-$i$ of the resulting data is

$$\left\langle j.X,\ k.X,\ W^{(1)}|_j^\sigma, W^{(2)}|_j^\sigma, \ldots, W^{(m)}|_j^\sigma \right\rangle \tag{6.3}$$

where $\left\{W^{(1)}, \ldots, W^{(m)}\right\} = adj(V_X; \mathcal{M}')$.

If tests for RBO (split-RBO and pair-RBO) are done, orientations between pairs of attribute classes involved in RBOs will be either oriented (correctly or incorrectly) or unoriented. If one assumes that the orientations are reliable, we can skip RCI tests related to an unshielded triple $i.X - j.Y - k.Z$ in a ground graph where:

- Both orientations between X and Y and between Y and Z are determined;
- $\langle X, Y, Z\rangle$ forms an attribute-class level non-collider, i.e., $X \leftarrow Y$ or $Y \rightarrow Z$.

If orientation between X and Y are undetermined (due to the given orientation rule), although some of RBO tests are performed, then we exclude determining orientations between X and Y through further tests. As a consequence, remaining RCI tests are for unshielded triples with at least one relational dependency with its cause is of cardinality one.

6.4.2.5 Tests for Non-RBO

The two types of RBO tests replace CUTs involving two attribute classes. Now we explore how we can perform RCI tests corresponding to the rest of CUTs where three attribute classes are involved.

Consider two CUTs $\langle V_X, \mathbf{P}.Y, R.Z\rangle$ and $\langle V_Z, \tilde{\mathbf{Q}}.Y, \tilde{R}.Z\rangle$ given $P.X \in adj(V_Y; \mathcal{M}')$ and $Q.Z \in adj(V_Y; \mathcal{M}')$ where $X \neq Z$. Without knowing $X \not\perp\!\!\!\perp_\pi Z$ or $Z \not\perp\!\!\!\perp_\pi X$, a separating set should be sought among $adj(V_X; \mathcal{M}')$ and $adj(V_Z; \mathcal{M}')$. Excluding

skippable cases, the rest of CUTs to be tested satisfy that at least one of P and Q is one-to-one. Without loss of generality, let Q be one-to-one and $\{k.Z\}$ be $Q.Z|_i^\sigma$. Then,

- **Case** P is one-to-one: This is a trivial case where a flattened data contains singletons except conditionals. Let $\{j.X\} = P.X|_i^\sigma$ and $\{k.Z\} = Q.Z|_i^\sigma$. We can construct a pair of data as follows

$$\left\langle j.X,\ k.Z,\ \ldots, W^{(\ell)}|_j^\sigma, \ldots \right\rangle$$

  with I_X as a base item class and

$$\left\langle j.X,\ k.Z,\ \ldots, T^{(\ell)}|_k^\sigma, \ldots \right\rangle$$

  with I_Z as a base item class where $\left\{W^{(1)}, \ldots, W^{(m)}\right\} = adj\left(V_X; \mathcal{M}'\right)$ and $\left\{T^{(1)}, \ldots, T^{(n)}\right\} = adj\left(V_Z; \mathcal{M}'\right)$ given $i \in \mathbf{i} = \left\{i \in \sigma\left(I_Y\right); P|_i^\sigma \neq \emptyset,\ Q|_i^\sigma \neq \emptyset\right\}$.

- **Case** $\mathsf{card}(P) = \mathsf{many}$ and $X \to Y$: Since there can be many Xs, we avoid using multiple values as they are since the use of CI test is valid only when the item attributes corresponding to the base item class are unique singletons. Hence, we select one item attribute among $P.X|_i^\sigma$. Let $s_i \sim \mathcal{U}\left\{P|_i^\sigma\right\}$ for $i \in \sigma\left(I_Y\right)$ where $P|_i^\sigma \neq \emptyset$. We similarly construct two data sets where corresponding row-i are

$$\left\langle s_i.X,\ k.Z,\ \ldots, W^{(\ell)}|_{s_i}^\sigma, \ldots \right\rangle$$

  and

$$\left\langle P.X|_i^\sigma,\ k.Z,\ \ldots, T^{(\ell)}|_k^\sigma, \ldots \right\rangle,$$

  respectively. The first one (i.e., with I_X as base item class) and the second one (i.e., with I_Z as base item class) incorporate two different sets of rows $\mathbf{i}$ and $\mathbf{i}'$ where both $\mathbf{i}$ and $\mathbf{i}'$ satisfy that $P.X|_i^\sigma \neq \emptyset$ and $Q.X|_i^\sigma \neq \emptyset$ and $\mathbf{i}$, additionally, satisfies that $s_a \neq s_j$ for two different items $a, j \in \mathbf{i}$.

6.4.2.6 Detecting Violations for Orientations

The above tests are based on conditions similar to the orientation-faithfulness for CBN where a separating set $\mathbf{S}$ not containing $\tilde{P}.Y$ nor $\tilde{Q}.Y$ (depending on the current choice of base item class) implies that their corresponding attribute classes form a collider $X \to Y \leftarrow Z$ in $\mathcal{G}_{\mathbf{A}}^{\mathcal{M}}$ (X is not necessarily different from Z). Acknowledging false negatives are the main concern through both Phase-I and Phase-II, we would like to address whether a separating set we obtained is reliable.

Without loss of generality, we consider a CI test on a data for a base item class I_X. If we obtained a separating set $\mathbf{S} \subseteq adj\,(V_X; \mathcal{M}')$, whether $\tilde{P}.Y \in \mathbf{S}$ or not provides a support for a non-collider while $\tilde{P}.Y \notin \mathbf{S}$ for a collider $X \to Y \leftarrow Z$. Although we can seek for multiple separating sets and decide orientations with the multiple test results, one can add an additional step to examine reliability of separating sets to be obtained for a given case.

Given a data for a case in split-RBO, pair-RBO, or other non-RBO, one can check the marginal independence between two columns of interests with an empty separating set. Further, we can check conditional independence between them given a column made of $\{i.Y\}_{i \in \mathbf{i}}$. If both tests turn out to be independent, then we might avoid deciding the orientation through additional tests through Algorithm 13. It is important to note that there is a subtle but important difference between a column made of $\{i.Y\}_{i \in \mathbf{i}}$ and $\tilde{P}.Y$. The use of $\tilde{P}.Y$ as conditionals is to ensure iid per condition while the use of $\{i.Y\}_{i \in \mathbf{i}}$ is to activate a collider (assuming it *is* a collider).

If both marginal test and conditional independence test with $\{i.Y\}_{i \in \mathbf{i}}$ are dependent, we can proceed to figure out separating sets based on a separating set rule (Section 6.4.2.3). Every time a separating set $\mathbf{S}$ is found and it does not contain $\tilde{P}.Y$ (or $\tilde{Q}.Y$ if $\mathbf{S} \subseteq adj\,(V_Z; \mathcal{M})$), then we perform an additional test with $\{i.Y\}_{i \in \mathbf{i}}$ added to $\mathbf{S}$. If it is dependent, then we can conclude with more confidence that the separating set without $\tilde{P}.Y$ is reliable and the absence of $\tilde{P}.Y$ in the separating set indeed induces that $X \to Y \leftarrow Z$.

6.4.3 Phase III: Combining Orientations

We have a set of CI test statements for determining adjacencies and orientations. A set of adjacencies is determined by whether we can find *any* separating set

between two relational variables under RCMC. Unlike adjacencies, orientations can be decided by taking different results into account.

The basic idea is quantifying the evidence of orientation, then decide the orientation of a set of relational dependencies, which maximize the given score metric. Our approach is rather simple yet effective. Through RBO-related tests, we counted RCI test results that support $X \to Y$ or $Y \to X$ if there is an RBO case between X and Y. Through non-RBO tests, we counted supports for a collider $X \to Y \leftarrow Z$ or non-collider if there exists a non-RBO case for $\langle X, Y, Z \rangle$. They form four categories of orientations. For any pair for RBO cases and triples for non-RBO cases, the given orientation rule will decide whether to orient, and then how to orient.

In most cases, some of orientations and the (non-)colliders for triples of attribute classes may conflict to each other — e.g., an orientation $X \to Y$ and a collider $Y \to X \leftarrow Z$. We treat each of orientation between two attribute classes, collider, and non-collider are equally important — although some orientations are determined by the large number of CI tests, such a large number is mainly due to the degree of a related canonical relational variable. Hence, we determine the orientation of an RCM, which maximizes the number of conforming the four categories of information. If there are multiple sets with a maximum number of non-conflicting information, then we further refine candidates to choose one, which *mostly* shares the same set of oriented dependencies with other candidate RCMs.

6.5 Empirical Evaluation

We empirically evaluated how proposed features improve the performance of structure learning algorithm for relational causal model. The performance of learning algorithm relies on the quality of conditional independence test. We used a kernel-based conditional independence test named SDCIT (Self-Discrepancy Conditional Independence Test) (Lee and Honavar, 2017b), which is a nonparametric CI test shown to be robust to the choice of kernel parameters with power comparable to other kernel-based CI tests. To handle multiple values for a relational variable (or its variant), we used normalized R-convolution kernel. The choice of kernel reflects how we generated values in this experiment. We assumed all values are real-valued, and used RBF kernel (inside R-convolution kernel) where parameters are chosen

based on median heuristic (Gretton et al., 2007).

6.5.1 Random Relational Schema, RCM, and Relational Skeleton

The company schema is composed of three entity classes and two 'binary' relational classes without forming any cycle among item classes. Attribute classes only exist among entity classes. Hence, we randomly generate relational schemas. We randomly generate 3 (50%), 4 (25%), and 5 (25%) entity classes with specified probabilities. Two to five relationship classes are randomly generated to connect a pair (i.e., binary relationship) of entity classes or a triple with 75% and 25% probability, respectively. Cardinalities are selected uniformly. One to three attribute classes are generated for each entity class and zero or one attribute class is generated for each relationship class. Finally, generated relational schemas that does not satisfy following rules are filtered out: (i) all item classes are connected and (ii) the total number of attribute classes are less than or equal to 8.

RCMs are generated randomly. Given a random relational schema $\mathcal{S}$, max hop length h is selected randomly between 2 to 4. The number of dependencies is determined by $\lfloor \frac{3|\mathbf{A}|}{2} \rfloor$ and randomly selected among *all* relational dependencies within the given h. We limit the maximum number of parents of a canonical relational variable by 3. We reject generated RCMs if there exists an isolated attribute class in their CDGs. Further, if the CPRCM of the generated RCM has no directed dependencies. We adopt a linear model with additive Gaussian noise using average aggregators:

$$i.X := \left(\sum_{P.Y \in pa(V_X;\mathcal{M})} \frac{\beta_{P.Y,V_X}}{|P.Y|_i^\sigma|} \cdot \left(\sum_{j.Y \in P.Y|_i^\sigma} j.Y \right) \right) + \epsilon$$

where $\beta_{P.Y,V_X} = 1 + |\gamma|$ where $\gamma \sim \mathcal{N}(0, 0.1^2)$ for every $P.Y \in pa(V_X;\mathcal{M})$ for every $X \in \mathbf{A}$. $\epsilon \sim \mathcal{N}(0, 0.1^2)$. The set of parameters will likely yield a relational data less hostile for our learning algorithm given that $\beta \geq 1$ and the variance of noise is relatively small. This fulfills our intention to assess the behavior of learning algorithm across different settings. If we wanted to exploit the fact that the generated RCMs are based on an average aggregator, we can incorporate this into the choice of kernel so that R-convolution kernel is not necessary but a simple

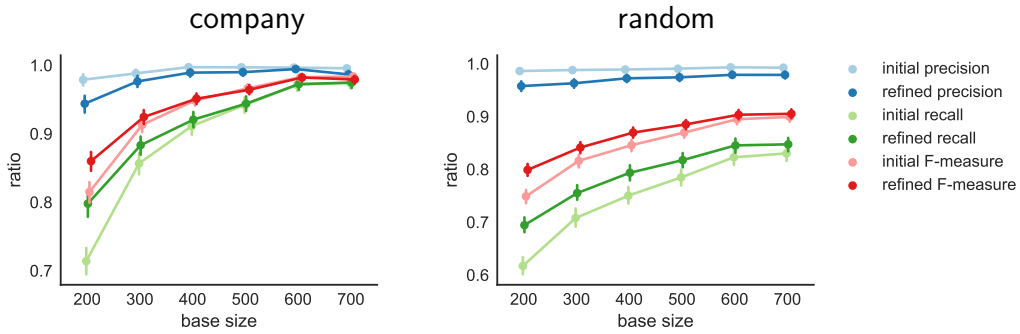


Figure 6.3: Performance of Phase-I with all features enabled (aggregation, order-independence, non-RCMC recovery, and refinement)

RBF kernel on averaged values is sufficient.

Random relational skeletons are generated with a user-specified *base* size n. Given n, the number of relationships (i.e., relationship instances or relationship items) for each relationship class is the twice of the base size if cardinality is many for every its participating entity class and the same as base size otherwise. The number of entities per entity class can be computed as $\lfloor 1.2^k \cdot n \rfloor$ where k is the number of related relationship classes with all-one cardinalities. For the company schema, there exists no relationship with all-one cardinalities. Given $n = 1000$, there are 2000 Develops items and 1000 items for each of other item class.

For company schema, we generated 300 different RCMs (i.e., different parameters with the same company RCM structure). For each RCM, we generate 6 relational skeletons corresponding to base size from 200 to 700, increased by 100. As a whole there are 1800 'companies'. For random data, there are 300 random schemas where each relational schema associates with an RCM. We generated 6 relational skeletons for base sizes from 200 to 700 for each pair of a relational schema and an RCM.

6.5.2 Phase-I

We first report the performance of RRCD for Phase-I. Precision, recall, and F-measure (i.e. F1 score) for undirected dependencies are reported (see Figure 6.3). As the size of data increases, more accurate RCMs are discovered since RCI tests can better catch genuine dependencies. We observe relatively high precision in general even with a small sized relational data, which implies that the main problem of the structure learning is false negatives due to weak dependencies.

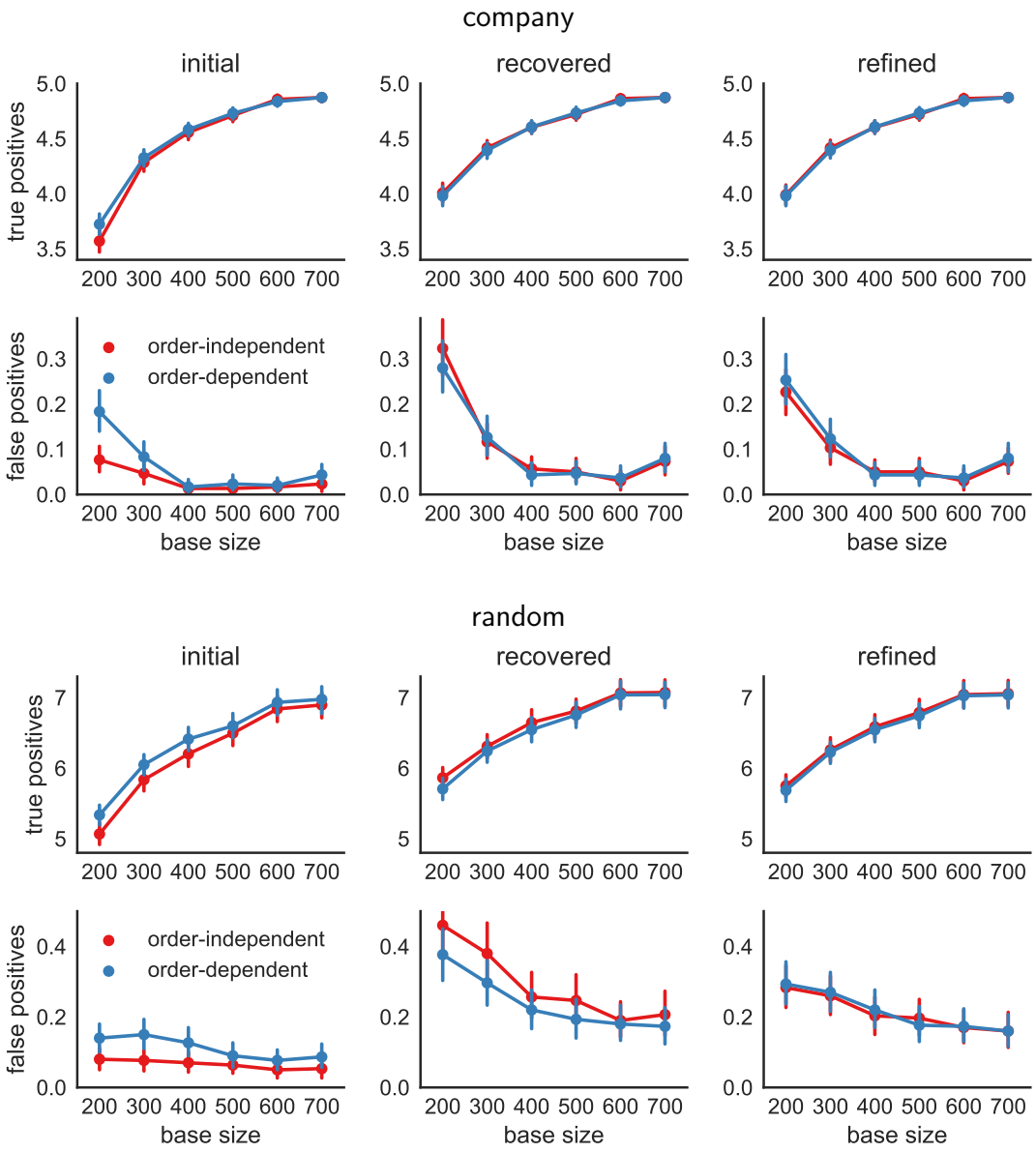


Figure 6.4: Performance on order-independence

6.5.2.1 Order-independence

Plots of performance with and without order-independence is depicted in Figure 6.4 with the average number of true and false positives per different stage of Phase-I with additional aggregation-based test. First, order-dependence can yield higher number of both true and false positives. We can observe that order-independence reduces the number of false positives by about half in small size data sets (200):

$0.170 \rightarrow 0.057$ (without aggregation), $0.183 \rightarrow 0.077$ (with aggregation) for company, and $0.110 \rightarrow 0.077$ (without aggregation), $0.140 \rightarrow 0.080$ (with aggregation) for random. When our post-procedures are applied (*recovered* and *refined*), the benefits of order-dependence vanishes in terms of true positives. The order-independence and the 'recover & refine' are complimentary. The order-independence can be viewed as an implicit refining step per different size of conditionals *before* adjacencies are initially identified.

6.5.2.2 Additional Aggregation-based Tests

In Figure 6.5, aggregation-based CI tests yield higher true positives without increasing false positives much at the initial stage. However, recovery and refining steps dilute the advantage of additional aggregation-based CI tests. Since non-aggregated test and its corresponding aggregated test are correlated, doubling the test does not significantly increase false positive rate.

We explored which types of RCI queries are 'saved' by aggregated tests, i.e., $(U \perp\!\!\!\perp V \mid \mathbf{W}) \wedge (f(U) \not\perp\!\!\!\perp V \mid \mathbf{W})$ such that U is adjacent to V at the end of Phase I. We report three cases: i) false positive, $U \notin adj(V; \mathcal{M})$; ii) right direction, $U \in pa(V; \mathcal{M})$; and iii) reverse direction, $U \in ch(V; \mathcal{M})$. We expected that the aggregation-based test is particularly useful when $U \in ch(V; \mathcal{M})$ since V affects each of item attribute in U 'individually'. Then, averaging values might help reducing noises. In Figure 6.6, we illustrate the average number of saved dependencies in the three categories and their proportions. Note that, an adjacency $P.X - V_Y$, which is also $\tilde{P}.Y - V_X$, can be counted twice. We can first observe that the total number of saved relational dependencies decreases as data size increases since the original (i.e., non-aggregation-based) test will catch weak dependencies better. RCI tests in a reverse direction, e.g., $U \in ch(V; \mathcal{M})$, are mostly saved by aggregation. The use of aggregation will become more useful as the relationships in a relational skeleton becomes more complex.

In an unreported experiment, we observed that aggregation-based test is especially useful when the given kernel-based CI test is weak. Specifically, we adopted a normalized R-convolution kernel with equal-size only, that is, kernel value between two multi-sets of different sizes is 0. The use of aggregation improved performance a lot more than the result reported here. However, an important question about the aggregation-based test is conditions under which the feature can improve precision

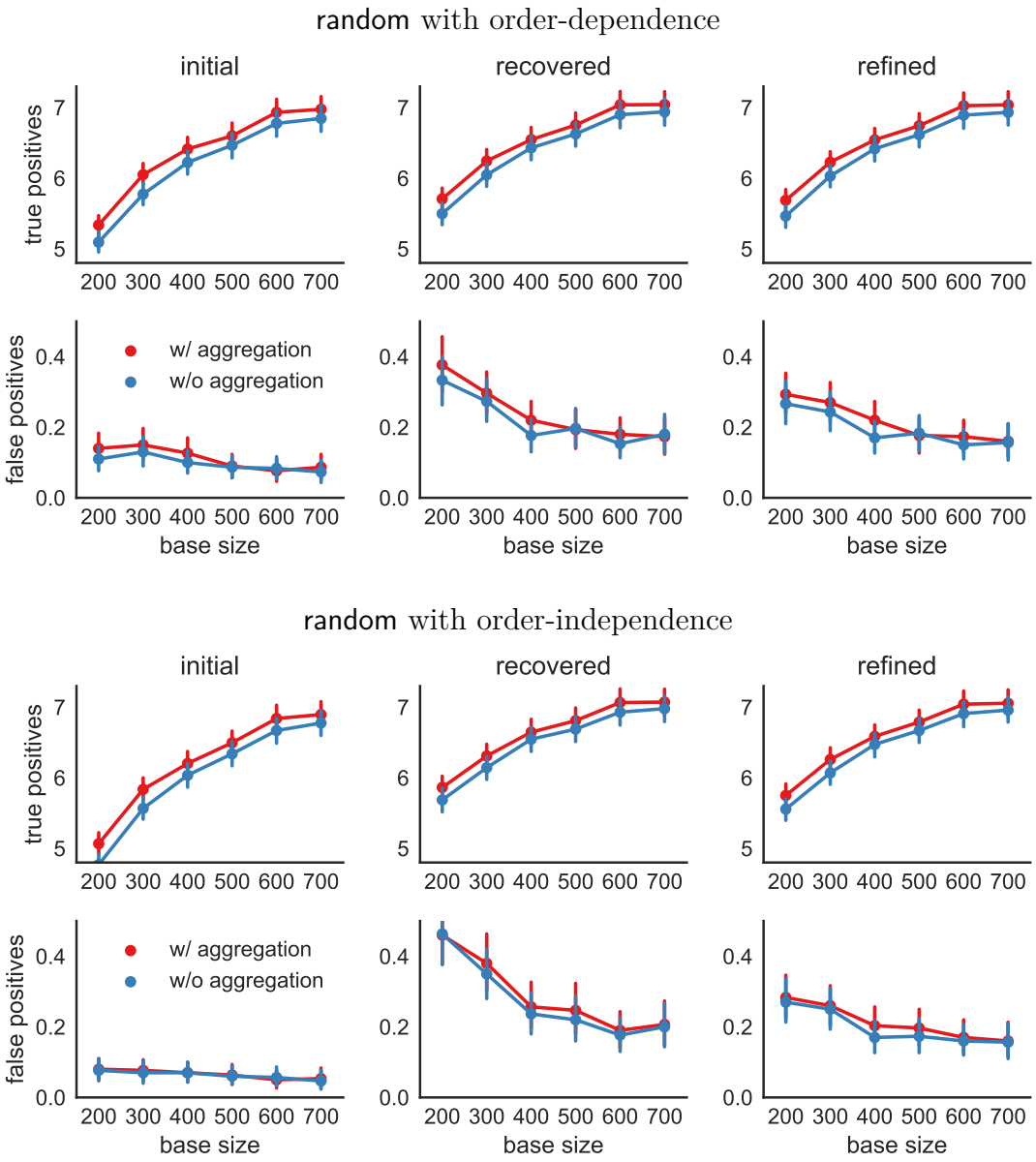


Figure 6.5: Effect of aggregation with order-(in)dependence

and recall — e.g., values are generated with a deterministic function and noise, $i.X := g\left(pa\left(V_X; \mathcal{M}\right)|_i^{\sigma}\right) + \epsilon$.

6.5.2.3 Recovery of non-RCMC and Refinement

We describe how the use of non-RCMC recovery and refinement affects the performance of RRCD in Figure 6.7. Tests for the violation of RCMC can recover some

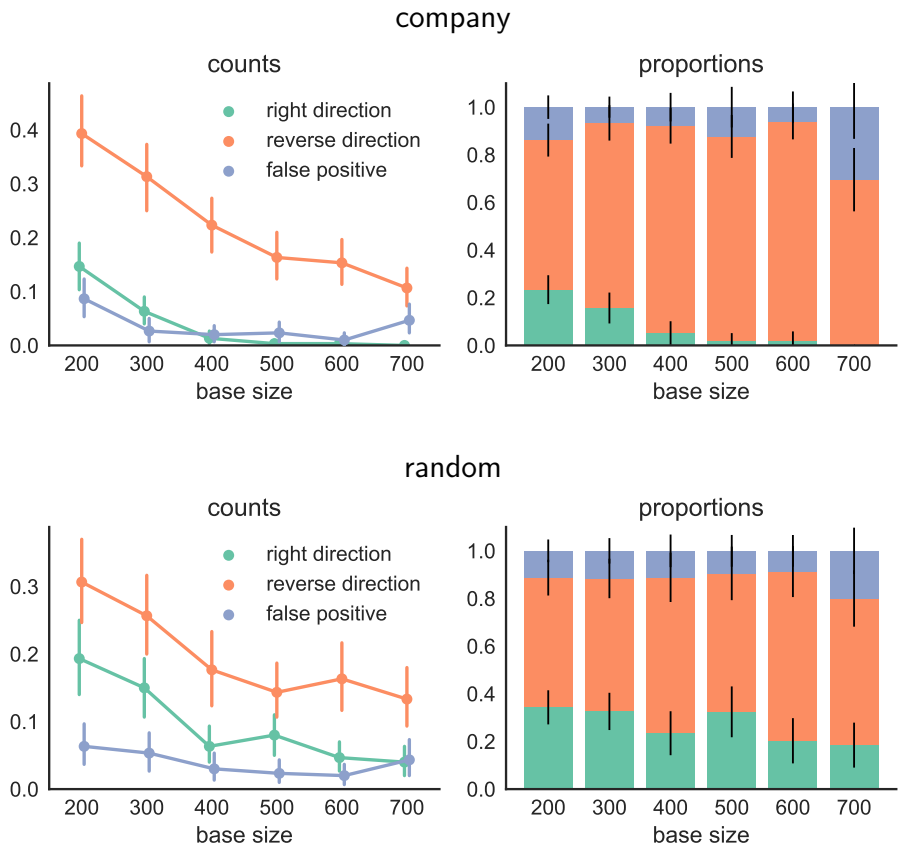


Figure 6.6: RCI query types saved by aggregation-based tests

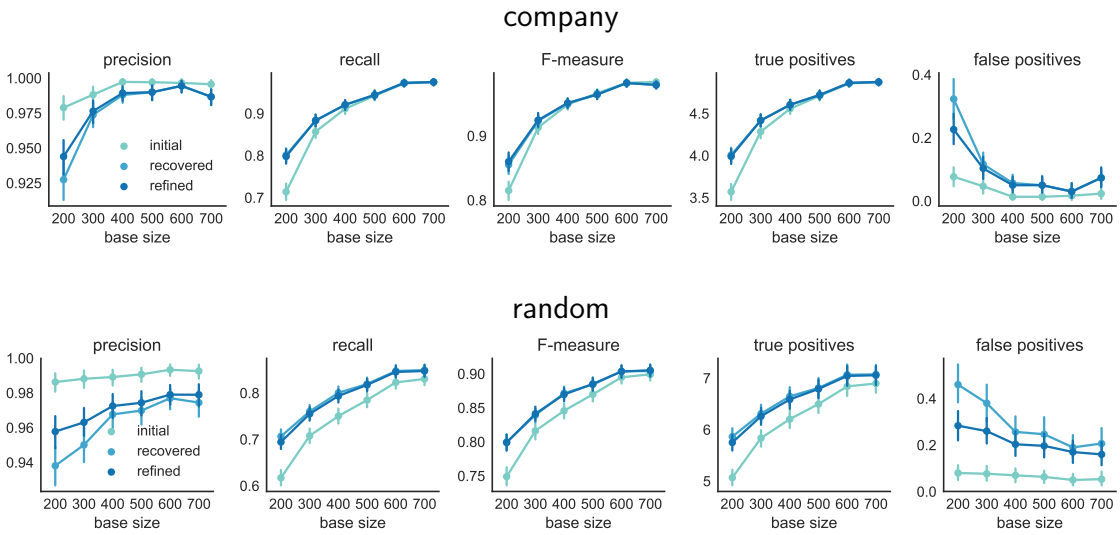


Figure 6.7: Performance changes across three stages of Phase-I

of false negatives increasing recall and F-measure while sacrificing precision. On random data (with aggregation and with order-independence), on average there are 2.23 missing relational dependencies among 8.45 dependencies. Detecting violation of RCMC (i.e., 'recovered') restored 0.63 dependencies where 0.40 were correct false negatives (turned into true positives) while 0.22 were correct true negatives (turned into false positives). That is, the precision of recovering non-RCMC was about 64.34%. The further refining procedure ('refined') put back 0.13 dependencies where 0.0778 dependencies were falsely recovered and 0.0472 were correctly recovered (i.e., refined wrongly). That is, the refinement is of 62.22% precision. The procedure will be likely useful when there are many false negatives (e.g., a large number of relational dependencies), and when researchers want to examine possible false negatives, etc. This additional information can be used in the later stage. For example, one might want to avoid RCI tests for orientation when such 'weak dependencies' are involved.

6.5.3 Phase-II and Phase-III

We first overview how each feature affects the performance of orientation in terms of precision, recall, and F-measure assuming perfect Phase-I, which allows us to better judge how different features work. More specifically, 'correctly directed' relational dependencies lie in the intersection of oriented relational dependencies through Phase-II and III and true relational dependencies. Then, precision and recall are the proportion of correctly directed relational dependencies among directed relational dependencies through Phase-II and III, and among directed relational dependencies in the corresponding CPRCM, respectively. Since, we have 300 relatively small RCMs, we report micro-average for precision, recall, and F-measure. In Table 6.1, we describe performance when we fix a feature with a value on three criteria. The results in general follow our expectation. In addition, detecting conflicts for both RBO and non-RBO increased recall. However, due to the dependency among features, best settings for each criterion cannot be trivially inferred. We additionally present a precision-recall scatter plot annotating Pareto frontiers per each base size. An insight from this figure is the clear trade-offs between precision and recall and the performance improves as the size of data increases.

feature	value	Precision	Recall	F-measure
base size	200	0.8340	0.5395	0.6552
	300	0.8567	0.5893	0.6982
	400	0.8853	0.6244	0.7323
	500	0.8937	0.6566	0.7570
	600	0.9003	0.6735	0.7706
	700	**0.9072**	**0.6922**	**0.7852**
aggregation-based tests	False	**0.8825**	0.6233	0.7306
	True	**0.8794**	**0.6352**	**0.7376**
detect conflicts RBO	False	0.8736	0.6202	0.7254
	True	**0.8882**	**0.6383**	**0.7428**
detect conflicts non-RBO	False	0.8726	0.6214	0.7259
	True	**0.8892**	**0.6370**	**0.7423**
orientation rule	majority	0.8687	**0.6860**	**0.7666**
	conservative	**0.8961**	0.5725	0.6986
separating set rule	first	0.8635	**0.6774**	**0.7592**
	minimal	**0.8936**	0.6458	0.7498
	full	0.8880	0.5646	0.6903

Table 6.1: Performance over individual features for Phase-II (random)

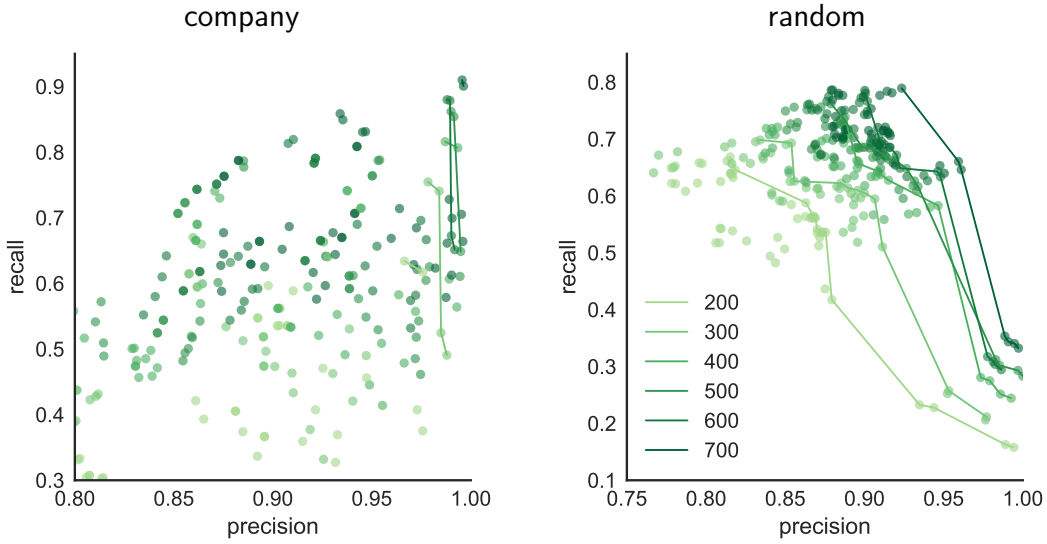


Figure 6.8: Performance of different settings for Phase-II

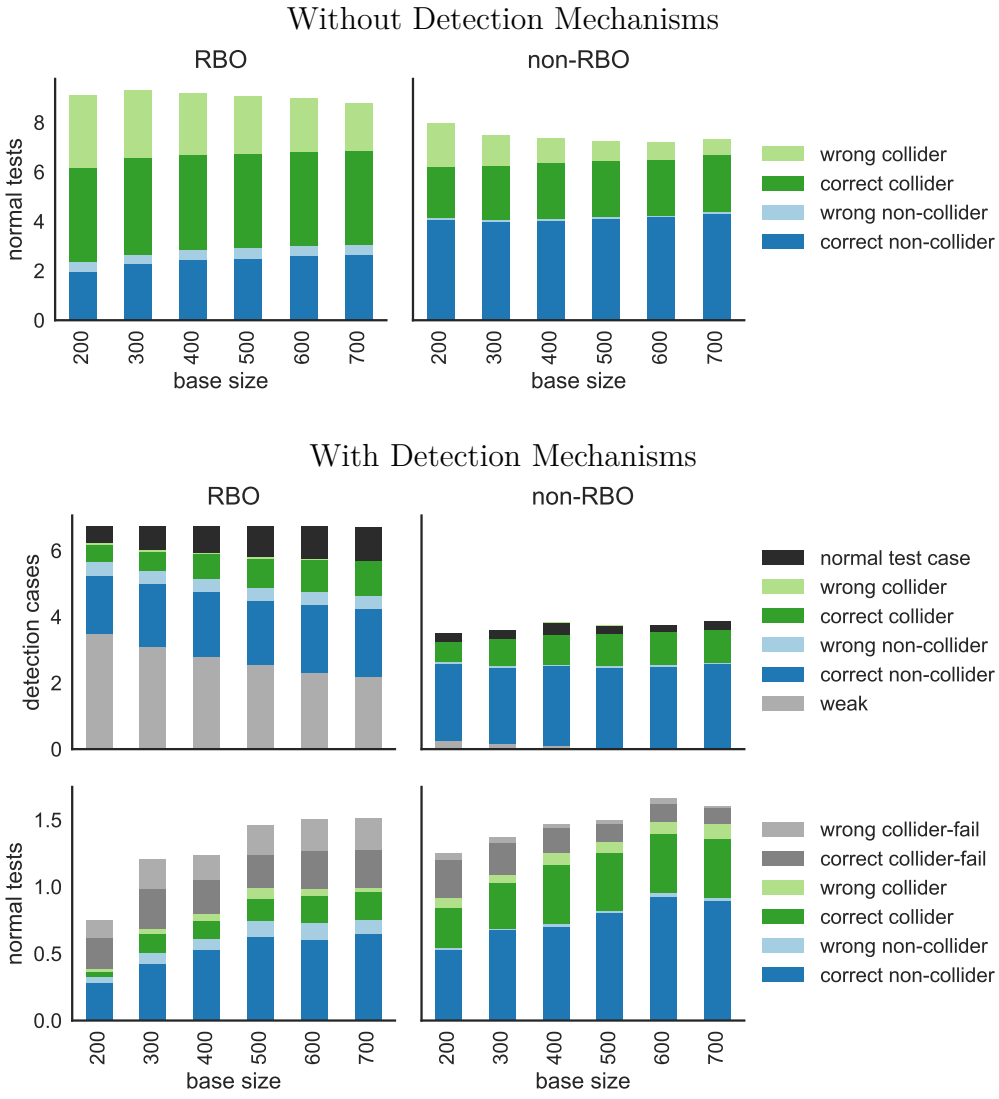


Figure 6.9: Effect of detection mechanism (random data with 'minimal' separating set)

6.5.3.1 Detecting Conflicts for RBO and non-RBO

We investigate how conflict detection mechanisms for RBO and non-RBO work together. In Figure 6.9, we illustrate the average number of RCI tests which turned out to be colliders or non-colliders, and whether the RCI test results are right or wrong. With both detection mechanisms enabled, we can further show how a pair of RCI tests determined weak cases, colliders, non-colliders, or further tests are needed. Without detection, one can observe that there exists non-negligible

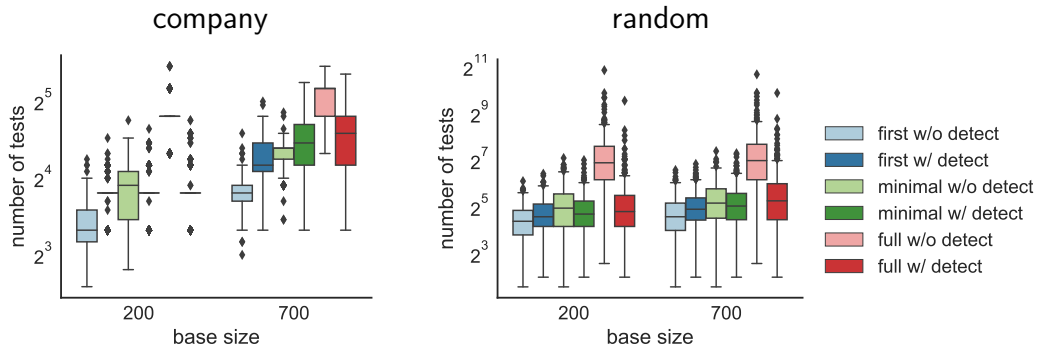


Figure 6.10: Number of tests with respect to the use of detection mechanisms and the choice of separating set rule

amount of wrong collider test results. This implies that a set of conditionals without blocking $\tilde{P}.Y$ (or $\tilde{Q}.Y$) yields wrong independence. This, again, suggests how false negatives dominate the performance of the learning algorithm. The mechanism detects lots of weak cases. However, the amount of weak cases decreases as the size of data increases. Only small amount of cases require multiple RCI tests varying conditionals. The result demonstrates that the most of errors are occurred during RBO-related tests.

6.5.3.2 Number of Tests

We finally represent the number of tests with respect to the rules for separating set and the use of detection mechanisms in Figure 6.10. Since the detection mechanism requires two tests per case, this increases the number of tests when 'first' separating set rule is used. However, the mechanism avoids non-necessary tests on weak cases, and can reduce the number of RCI tests required to orient dependencies.

6.6 Concluding Remarks

In this chapter, we introduced a robust relational causal discovery algorithm adopting traditional CI test given the fact that canonical relational variable is independent and identically distributed given its parents. We adopted and developed several features including order-independence, additional aggregation-based tests, identification of non-RCMC, non-traditional RCI tests, detection of weak cases for orientation, etc. We empirically evaluated how these features help improve some

performance criteria.

However, there are several known limitations in the design of algorithm and the CI test employed in the algorithm. First, the use of a normalized R-convolution-based kernel for sets is under the assumption that two sets of values sampled from the same distribution will yield like results. In our experiments, two similar distributions imply that their averages are also similar, which justifies the use of the normalized R-convolution kernel when an average-aggregator is used to define underlying RCMs. Without such an assumption, we should set kernel values between two sets of different sizes to zero. If we are only interested in a kernel for two equal sized sets, we wonder whether a matching-based similarity measure can be used and what will be the benefits of it over employing an R-convolution-based kernel. Second, if every item attribute value, e.g., $i.Y$, is determined by a large set of values, e.g., $P.X|_i^\sigma$, it becomes harder to correctly estimate how much a single value, e.g., $j.X \in P.X|_i^\sigma$, in the set affects $i.Y$ with respect to the rest $P.X|_i^\sigma \setminus \{j.X\}$. Third, we are unaware of mathematical language or theories to clearly express a given relational data as a sample from a population. We see tackling these challenges as an important future research direction.

Chapter 7
Conclusions

This chapter summarizes contributions of this dissertation and lists many opportunities for future work.

7.1 Summary of Contributions

This dissertation addressed fundamental issues to better understand relational causal model in both theoretical and practical perspectives – (i) reasoning about conditional independence with a given model and from relational data and (ii) learning the structure of relational causal model given an independence oracle and from relational data.

- Path semantics — Maier and his colleagues originally proposed RCM with bridge-burning semantics, which defines how a relational path should be interpreted in a relational skeleton. We revealed that bridge-burning semantics is less suitable for designing a learning algorithm for RCM with an independence oracle and for analyzing Markov equivalence class of RCMs. Path semantics inherits the advantage of bridge-burning semantics over walk semantics as demonstrated in (Maier, 2014). However, path semantics is free from the problematic behavior of bridge-burning semantics.

- Analysis of abstract ground graphs — Abstract Ground Graph (AGG) is proposed as a graphical structure that resides in between an RCM and its ground graphs. We critically examined how an AGG should be revised and must not be used in the task of relational d-separation: (i) we showed that extend to define RVEs, edges between two relational variables in an AGG, does not yield enough edges to identify parents of non-canonical relational variables

(for both semantics); (ii) we proposed newextend (which is tightly related to a CUT enumeration algorithm, Algorithm 6) for RVEs; (iii) we disproved conditions for the *intersectability* of two relational paths (Maier, 2014) (for both semantics) and provided a correct necessary and sufficient condition for path semantics; (iv) we identified that how IVEs are defined does not satisfy the properties they should possess, and introduced *co-intersectability* (Definition 13) and an algorithm (Algorithm 2) to test it. Finally, we showed that even a revised Abstract Ground Graph is not a medium for relational d-separation (for both semantics).

- Sound methods for relational d-separation — Existing methods relying on abstract ground graphs or its variant are neither sound nor complete. We provided several sound methods for relational d-separation: (i) a class dependency graph-based approach, which checks a necessary condition for the existence of a d-connection path by exploring connections among attribute classes; (ii) a randomized approach, which is a sufficient condition for relational d-connection; (iii) a relational-variable based approach, which implicitly builds a lifted representation based on newextend and *intersectability*, and checks a necessary condition for relational d-connection; and (iv) a constructive-approach, which incrementally builds all possible essential structures for d-connection paths, if exists.

- Characterization of Markov equivalence class of an RCM and a sound and complete relational causal discovery algorithm under path semantics — An RCM entails a set of relational conditional independence statements. The Markov equivalence class of an RCM is a set of RCMs sharing the common set of RCI statements. We characterized a necessary and sufficient condition for two RCMs to be Markov equivalent (Theorem 29). We offered a connection between an unshielded triple in a ground graph and an RCI query. We introduced a new concept, Canonical Unshielded Triple (CUT), of an RCM and provided an efficient way to enumerate a sufficient set of CUTs to obtain the pattern of an RCM. Finally, we devised a sound and complete algorithm to obtain a Completed Partially-directed RCM (CPRCM), which can uniquely represent the Markov equivalence class of an RCM.

- Relational conditional independence test — We showed, through examples,

how naively applying a traditional conditional independence test on a flattened representation of a relational data results in false positives. Under assumptions on the relationship between a relational data and an underlying relational structure, we showed that one can reduce false conditional dependence through taking relational structures into account. A kernel-based conditional independence test for relational data (KRCIT) is proposed and other variants are empirically compared. We observed a decrease in false negatives under certain conditions.

- Robust relational causal discovery — We examined how and under which conditions a traditional CI test can be safely adopted to conduct relational conditional independence tests on a given relational data for learning the structure of an RCM. Some of the existing techniques developed for causal Bayesian networks are adopted. We investigated phenomena specific to relational causal models and proposed several methods to make a learning algorithm more robust to the violation of assumptions.

7.2 Future Research Directions

Based on important and novel insights into RCMs this dissertation provided, we see opportunities and limitations of RCMs. Here are some directions for future studies.

- The definition of a relational variable shaped what an RCM can be. A relational variable consists of a relational path and an attribute class. It is the current form of a relational path that limits what 'relational variables' can be, and, thus, what an RCM can be. For example, such form forbids us from representing the union, subtraction, or intersection of two relational variables. In Chapter 5, we introduced a class of variables defined on a relational schema called an *isomorphism-invariant relational variable* (Section 5.3). Adopting this type of relational variable for an RCM will significantly improve the expressivity of resulting RCMs. However, we will have to face new challenges, for example, characterizing properties of RCMs with respect to RCI, Markov equivalence class, etc.

- An RCM generalizes a causal Bayesian network to a relational setting where a CBN assumes that there is no hidden confounder among observed variables.

This would limits the use of RCMs in more realistic settings where we not only have no access to some of unobserved variables but also do not know the existence of some of unobserved confounders. Ancestral graphs (Richardson and Spirtes, 2002; Zhang, 2008) extend the notion of CBNs allowing partly learning the underlying causal structure in the presence of hidden confounders. For the modeling purpose, a structural causal model (SCM) (Pearl, 2000) can represent hidden confounders. Many mathematical tools are developed to help causal inference with an SCM. Hence, it is desirable to consider generalizing ancestral graphs or SCMs to a relational setting.

- We assumed that a class dependency graph corresponding to an RCM can be represented as a directed acyclic graph. This acyclicity constraint at the level of attribute classes prohibits the use of RCMs in domains where self-loops or cycles over attribute classes are appropriate. For example, (biological) parents' genes affect one's genes. Still such a relational dependency will guarantee the resulting ground graphs to be acyclic. Hence, lifting attribute class level acyclicity assumption will be an important step for an RCM to be widely adopted. One way to allow such cycles is extending an RCM to harness explicit temporal relationships, which has recently been investigated (Marazopoulou et al., 2015; Marazopoulou, 2017). Unfortunately, the recent work is problematic as it generalizes RCM's lifted representation, Abstract Ground Graph (Maier et al., 2013b), which is proved to be not sound in reasoning about relational d-separation (Lee and Honavar, 2015) (also see Section 3.4.5). Therefore, extending an RCM to a temporal RCM and solving similar tasks as in this dissertation is still an open problem regardless of the choice of semantics.

- We proposed several sound methods to answer relational conditional independence queries for any valid relational schemas and RCMs. However, no proposed method is complete. An important question is whether there exists a class of queries, schemas, or RCMs where RCI queries can be answered soundly and completely.

- Our constructive approach to relational d-separation (Section 3.4.4) is basically a search algorithm for a d-connection path in some ground graph. With a user-set finite length of d-connection path, the algorithm is complete. However,

we do not know a clear stopping criterion to correctly determine whether there is no d-connection path. We observed that the only situation the current algorithm does not stop is when a given path (i.e., a state in search space) can be extended with a collider. However, it is not very obvious for us that under which conditions one can guarantee that extending a given path with a collider is redundant. Once this question is answered, the constructive approach with a proper stopping criterion can be the first sound and complete algorithm for determining relational d-separation although it might not be efficient.

- Relational conditional independence test seems difficult to quest, if not impossible, especially when a relational data of interest associates with a complex relational skeleton. Our proposed method adjusts non-iid item attribute values by taking their relational structures into account under some assumptions. The method can be improved by incorporating analysis of relational data. How does an underlying relational structure correlate with its values? How can we properly set parameters for a chosen graph kernel? How can we take advantage by assuming a functional form, e.g., additive Gaussian noise?

- Robust relational causal discovery opens the possibility for learning the structure of an RCM without concerning false positives arose by the underlying relational structure. Although we examined how RCMs would differ from CBNs with regard to structure learning, and proposed remedies for various problems specific to relational data sets, more theoretical and practical improvements can be further made. In a theoretical point of view, our faithfulness-like condition for a collection of non-iid random variables is not mathematically rigorous and can be improved by adopting notions developed for theories for stochastic processes, dynamical systems, or network science. For a practical side, whether to use an aggregator, which aggregator to use, which kernel should be used with what parameters, etc need to be decided in a more principled way. If some of such decisions are based on testable assumptions, then checking violations of such assumptions must be incorporated in the learning process of an RCM.

Bibliography

Abellán, J., Gómez-Olmedo, M., and Moral, S. (2006). Some variations on the PC algorithm. In *Proceedings of the Third European Workshop on Probabilistic Graphical Models (PGM' 06)*, pages 1–8.

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.

Angrist, J. D., Imbens, G. W., and Rubin, D. B. (1996). Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*, 91(434):444.

Arbour, D., Garant, D., and Jensen, D. (2016a). Inferring network effects from observational data. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 715–724, New York, New York, USA. ACM Press.

Arbour, D., Marazopoulou, K., and Jensen, D. (2016b). Inferring causal direction from relational data. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*, pages 12–21.

Aronow, P. M. and Samii, C. (2013). Estimating average causal effects under interference between units. *Working Paper*, 06520:1–24.

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

Borboudakis, G. and Tsamardinos, I. (2016). Towards robust and versatile causal discovery for business applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1435–1444, New York, NY. ACM.

Borgwardt, K. M. and Kriegel, H.-P. (2005). Shortest-path kernels on graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 74–81.

Bromberg, F. and Margaritis, D. (2009). Improving the reliability of causal discovery from small data sets using argumentation. *Journal of Machine Learning Research*, 10:301–340.

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

Cano, A., Gómez-Olmedo, M., and Moral, S. (2008). A score based ranking of the edges for the PC algorithm. In *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models (PGM 2008)*, pages 41–48.

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., Heckerman, D., and Meek, C. (2004). Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5:1287–1330.

Chickering, D. M. and Meek, C. (2002). Finding optimal Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 94–102.

Chwialkowski, K., Sejdinovic, D., and Gretton, A. (2014). A wild bootstrap for degenerate kernel tests. In *Advances in Neural Information Processing Systems 27*, pages 3608–3616.

Claassen, T. and Heskes, T. (2013). Bayesian probabilities for constraint-based causal discovery. *IJCAI International Joint Conference on Artificial Intelligence*, pages 2992–2996.

Claassen, T., Mooij, J., and Heskes, T. (2013). Learning sparse causal models is not NP-hard. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 172–181, Corvallis, Oregon. AUAI Press.

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

Cooper, G. F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347.

Cox, D. R. (1958). *Planning of Experiments*. Wiley.

Dash, D. H. and Druzdzel, M. J. (1999). A hybrid anytime algorithm for the construction of causal models from sparse data. In *Proceedings of the Fifteenth Annual conference on Uncertainty in Artificial Intelligence*, pages 142–149.

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

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.

Doran, G., Muandet, K., Zhang, K., and Schölkopf, B. (2014). A permutation-based kernel conditional independence test. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 132–141.

Fisher, R. A. (1958). Cancer and smoking. 182:596.

Flaxman, S. R., Neill, D. B., and Smola, A. J. (2016). Gaussian processes for independence tests with non-iid data in causal inference. *ACM Transactions on Intelligent Systems and Technology*, 7(2):1–23.

Frangakis, C. E. and Rubin, D. B. (2002). Principal stratification in causal inference. *Biometrics*, 58(1):21–29.

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.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620.

Fukumizu, K., Gretton, A., Sun, X., and Schölkopf, B. (2008). Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems 20*, pages 489–496.

Geiger, D., Verma, T., and Pearl, J. (1989). d-separation: From theorems to algorithms. In *Proceedings of the Fifth Conference on Uncertainty in Artificial Intelligence*, pages 118–125.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. J. (2007). A kernel method for the two-sample problem. In *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory. ALT 2005*, pages 63–77. Springer.

Haavelmo, T. (1943). The statistical implications of a system of simultaneous equations. *Econometrica*, 11(1):1–12.

Haussler, D. (1999). Convolution kernels on discrete structures. Technical report, University of California, Santa Cruz.

Heckerman, D. (1995). A Bayesian approach to learning causal networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 285–295, San Francisco, CA. AUAI Press.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243.

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.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.

Holland, P. W. (1986). Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945.

Hudgens, M. G. and Halloran, M. E. (2008). Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482):832–842.

Hyttinen, A., Hoyer, P. O., Eberhardt, F., and Järvisalo, M. (2013). Discovering cyclic causal models with latent variables: A general SAT-based procedure. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 301–310. AUAI Press.

Koller, D. (1999). Probabilistic Relational Models. In *Proceedings of the 9th International Workshop on Inductive Logic Programming*, pages 3–13.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Kriege, N. M., Giscard, P.-L., and Wilson, R. C. (2016). On valid optimal assignment kernels and applications to graph classification. In *Advances in Neural Information Processing Systems 29*, pages 1623–1631.

Lauritzen, S. L. and Wermuth, N. (1989). Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, 17(4):1916–1916.

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. (2016a). A characterization of Markov equivalence classes of relational causal models under path semantics. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 387–396, Corvallis, Oregon. AUAI Press.

Lee, S. and Honavar, V. (2016b). On learning causal models from relational data. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3263–3270, Palo Alto, CA. AAAI Press.

Lee, S. and Honavar, V. (2017a). A kernel conditional independence test for relational data. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon. AUAI Press.

Lee, S. and Honavar, V. (2017b). Self-discrepancy conditional independence test. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon. AUAI Press.

Magliacane, S., Claassen, T., and Mooij, J. (2016). Ancestral causal inference. In *Advances In Neural Information Processing Systems 29*, pages 4466–4474.

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, Bellevue, WA. 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., Oktay, H., and Jensen, D. (2010). Learning causal models of relational domains. In *Proceedings of the Twenty-Fourth National Conference on Artificial Intelligence*, pages 531–538.

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.

Marazopoulou, M. (2017). *Temporal and Relational Models for Causality: Representation and Learning*. PhD thesis, University of Massachusetts Amherst.

Margaritis, D. and Thrun, S. (2000). Bayesian network induction via local neighborhoods. In Solla, S. A., Leen, T. K., and Müller, K., editors, *Advances in Neural Information Processing Systems 12*, pages 505–511. MIT Press.

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.

Neapolitan, R. E. (2003). *Learning Bayesian Networks*. Pearson.

Neville, J. and Jensen, D. (2007). Relational dependency networks. *Journal of Machine Learning Research*, 8:653–692.

Neyman, J., Dabrowska, D. M., and Speed, T. P. (1990). On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, 5(4):465–472.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Pearl, J. (1995). Causal diagrams for empirical research. *Biometrika*, 82(4):669–688.

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

Pellet, J. P. and Elisseeff, a. (2008). Using Markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9:1295–1342.

Ramsey, J., Spirtes, P. L., and Zhang, J. (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, M. and Domingos, P. (2006). Markov logic networks. *Machine Learning Journal*, 62(1-2):107:136.

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

Rosenbaum, P. R. (2007). Interference between units in randomized experiments. *Journal of the American Statistical Association*, 102(477):191–200.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701.

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. (2011). Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561.

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

Sobel, M. E. (2006). What do randomized studies of housing mobility demonstrate? *Journal of the American Statistical Association*, 101(476):1398–1407.

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

Studeny, M. (2005). *Probabilistic Conditional Independence Structures*. Information Science and Statistics. Springer-Verlag, London.

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.

Tian, J. and Pearl, J. (2002). A general identification condition for causal effects. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 567–573, Menlo Park, CA. AAAI Press.

Toulis, P. and Kao, E. (2013). Estimation of causal peer influence effects. *Proceedings of The 30th International Conference on Machine Learning*, 28:1489–1497.

Triantafillou, S. and Tsamardinos, I. (2015). Constraint-based causal discovery from multiple interventions over overlapping variable sets. *Journal of Machine Learning Research*, 16:2147–2205.

Triantafillou, S., Tsamardinos, I., and Tollis, I. G. (2010). Learning causal structure from overlapping variable sets. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 860–867.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78.

Ugander, J., Karrer, B., Backstrom, L., and Kleinberg, J. (2013). Graph cluster randomization: network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 329–337, New York, NY. ACM Press.

Verma, T. and Pearl, J. (1988). Causal networks: Semantics and expressiveness. In *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, pages 352–359, Mountain View, CA.

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.

Wright, S. (1921). Correlation and causation. *Journal of Agricultural Research*, 20:557–585.

Zhalama, Zhang, J., Eberhardt, F., and Mayer, W. (2017). SAT-based causal discovery under weaker assumptions. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon. AUAI Press.

Zhang, J. (2008). On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16–17):1873–1896.

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

Zhang, X., Song, L., Gretton, A., and Smola, A. (2009). Kernel measures of independence for non-iid data. In *Advances in Neural Information Processing Systems 21*, pages 1937–1944.

Vita

Sanghack Lee

Research Interests

- Causal Inference with Multiple Heterogeneous Environments, Causal Discovery from Relational Data

Education

- Ph.D., College of Information Sciences and Technology, Pennsylvania State University, 2018
- M.Sc. Sogang University (South Korea), Department of Computer Science and Engineering, 2006
- B.E. Sogang University (South Korea), Department of Computer Science and Engineering, 2004

Selected Publications

- Lee, S. and Honavar, V. (2017). A kernel conditional independence test for relational data. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon. AUAI Press.
- Lee, S. and Honavar, V. (2017). Self-discrepancy conditional independence test. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon. AUAI Press
- Lee, S. and Honavar, V. (2016). A characterization of Markov equivalence classes of relational causal models under path semantics. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 387–396, Corvallis, Oregon. AUAI Press.
- Lee, S. and Honavar, V. (2016). On learning causal models from relational data. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3263–3270, Palo Alto, CA. AAAI 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.