Learning Causal Models of Relational Domains

Marc Maier and **Brian Taylor** and **Hüseyin Oktay** and **David Jensen**
Knowledge Discovery Laboratory
Department of Computer Science
University of Massachusetts Amherst
{maier, btaylor, hoktay, jensen}@cs.umass.edu

Abstract

Methods for discovering causal knowledge from observational data have been a persistent topic of AI research for several decades. Essentially all of this work focuses on knowledge representations for propositional domains. In this paper, we present several key algorithmic and theoretical innovations that extend causal discovery to relational domains. We provide strong evidence that effective learning of causal models is enhanced by relational representations. We present an algorithm, relational PC, that learns causal dependencies in a state-of-the-art relational representation, and we identify the key representational and algorithmic innovations that make the algorithm possible. Finally, we prove the algorithm's theoretical correctness and demonstrate its effectiveness on synthetic and real data sets.

1 Introduction

The causal mechanisms that underlie many real world domains can be challenging to represent and learn. While statistical associations between variables can be observed and measured, knowledge of the underlying causal model must be inferred from a pattern of such associations and prior knowledge. Any given statistical association may result from one of several different causal structures, and distinguishing among these structures can be difficult.

Despite these difficulties, causal models are vital for effective reasoning in many domains. Causal knowledge is necessary for reasoning about the consequences of actions, whereas knowledge of statistical associations alone can only inform expectations as a passive observer. Causal knowledge can also provide more robust models in the face of changing underlying distributions, whereas the parameters of many associational models must be recalibrated if the underlying distributions change.

As a concrete example, consider the domain of scholarly publishing. A causal model of such a domain (Figure 1) might include entities (authors, papers, and venues), relationships (published-in, authored-by), and attributes (topic, research interest). Such a model might indicate that the research interest of an author affects the topics of the papers that he or she writes. Similarly, the focus of a publishing venue might affect the topic of papers published in that venue. More elaborate causal models could indicate actions that facilitate particular outcomes. For example, a model might allow reasoning about which actions would affect a paper's citation rate.

Methods for discovering causal dependencies from observational data have been the focus of decades of work in AI, statistics, philosophy, and social science. This work has uncovered a number of basic methods, including the PC algorithm for learning the structure of causal dependencies, rules for edge orientation that correctly infer the direction of inferred causal dependencies, and fundamental principles and assumptions necessary for valid causal reasoning (Spirtes, Glymour, and Scheines 2000; Pearl and Verma 1991; Holland and Rubin 1988).

However, the vast majority of this work has focused on a single knowledge representation: directed graphical models of propositional data. Such models effectively represent individual directed causal dependencies, and efficient algorithms exist for reasoning about the conditional independence assumptions that a pattern of such causal dependencies encodes. However, such propositional representations rarely capture causal dependencies between entity types and almost never represent dependencies that involve the existence or cardinality of entities or relationships.

There is a growing community of researchers aiming to learn models of relational domains; however, all algorithms to date are not intended for learning causal models. These algorithms are typically based on the search-and-score paradigm, with the objective of learning a model of high likelihood and not of explicitly learning the generative structure of the data (Getoor et al. 2007; Richardson and Domingos 2006; Taskar, Abbeel, and Koller 2002).

In this paper, we present several key algorithmic and theoretical innovations that enable causal discovery in relational domains. We provide strong evidence that effective learning of causal models in relational domains requires a relational representation. We show that to identify correct causal structures, we must explicitly model the uncertainty over relationships (referred to as existence uncertainty), and we also characterize how patterns of associations can constrain the space of causal models for relational data. Additionally, we present the first algorithm, relational PC, that learns causal dependencies from relational data. We prove its correctness and demonstrate its effectiveness on synthetic and real data.

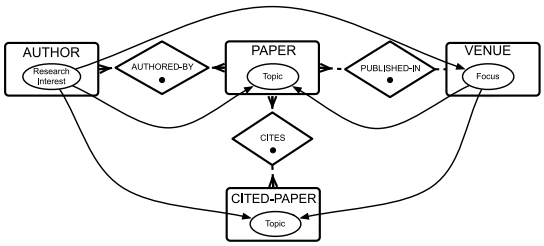

Figure 1: Citation database in the DAPER language.

2 Relational Representations

The directed acyclic probabilistic entity-relationship (DAPER) model is a highly expressive extension of the standard entity-relationship (ER) model that incorporates probabilistic dependencies (Heckerman, Meek, and Koller 2007). The DAPER model has been proven to unify existing representations, including probabilistic relational models and plate models, and is strictly more expressive.

The ER model describes the contents of a relational database $\mathcal{D} = \{\mathcal{E}, \mathcal{R}\}$ as a set of entity classes $\mathcal{E} = \{E_1, \ldots, E_m\}$ and relationship classes $\mathcal{R} = \{R_1, \ldots, R_n\}$. Along with the structure, there exist attribute classes $\mathcal{A}(B)$ for each $B \in \mathcal{E} \cup \mathcal{R}$. The DAPER model consists of an ER model over $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{A}$, and a set of arc classes defined over $\mathcal{A} \times \mathcal{A}$. An arc class corresponds to a probabilistic dependence from some $A.X$ to some $B.Y$, where $A, B \in \mathcal{E} \cup \mathcal{R}$, $X \in \mathcal{A}(A)$, and $Y \in \mathcal{A}(B)$, constrained by an arbitrary first-order expression. The set of arc classes is paired with a corresponding set of local probability distributions. See Figure 1 for an example represented in DAPER. For simplification, first-order expressions are omitted and existence is represented with a dot.

The DAPER model fully encodes a set of conditional independencies defined over its attribute classes. The ground graph of a DAPER model (sometimes referred to as the model "rolled out" over the data), consists of a node for every attribute $a.X \in \mathcal{A}(\sigma_{\mathcal{ER}})$ and an arc from $b.Y$ to $a.X$ if there is an arc class from $B.Y$ to $A.X$ and the arc's constraint is satisfied in the skeleton, $\sigma_{\mathcal{ER}}$. The DAPER model can be viewed as a powerful tool for modeling a joint probability distribution over relational data. As we show in this paper, the DAPER model is also a sufficient language for representing causal dependencies in relational data.

A common alternative to relational representations is *propositionalization*, which transforms relational data into propositional data. It was introduced as a way to continue to rely on existing algorithms instead of developing new ones to handle the increased complexity (Kramer, Lavrač, and Flach 2001). We show that propositionalization is largely inadequate for effective causal learning in relational domains.

2.1 Common causes

Statistical association between two variables X and Y is a necessary but not a sufficient condition for causal dependence. These two variables can be marginally dependent, yet conditionally independent given a common cause. Failing to explicitly condition on a common cause will result in an incorrect inference of causal dependence between X and Y. As a result, identifying causal structure is challenging when common causes are unrepresented or *latent*.

DAPER models and other relational formalisms typically represent a superset of those dependencies that can be represented in any given propositional formalism. Thus, common causes are more likely to be explicitly representable. For example, an author's research interests could be the common cause of the topic chosen for papers and research grants. However, most propositional models would not represent authors or their research interests, so a causal dependence could appear to hold between paper topics and grant topics.

2.2 Common effects

If two variables X and Y are causally independent but have a common effect Z, then X and Y will be conditionally dependent given Z, even though they are marginally independent. This phenomenon is known as Berkson's paradox (Berkson 1946). If Z is a latent variable, and we implicitly condition on its value, then we could falsely conclude that X and Y are causally dependent.

Methods for propositionalizing relational data frequently create implicit conditioning that can produce cases of Berkson's paradox. When testing the association between two variables on different entities linked by a relationship, we implicitly condition on the existence of that relationship. If the relationship is a common effect of the variables (e.g., a citation relationship caused jointly by the topic of the cited paper and the topic of the citing paper), then a propositionalized data set will induce a dependence between the variables.

The importance of this effect for causal discovery has not been identified in prior research. Modeling the probability of link existence—often called existence uncertainty—has been recognized as an inherently interesting topic in its own right (Getoor, Friedman, Koller, and Pfeffer 2007), but its effects on learning other causal dependencies have not been clearly identified. In addition, the distinction between different causal interpretations of link existence is well known in social science research—it is the question of homophily vs. social influence (Friedkin 1998)—but that work has not explored the implications of this distinction for knowledge representation and causal discovery.

2.3 Representation needs

Because of the difficulty of identifying correct causal structure in the presence of latent common causes and common effects, representations should not unnecessarily preclude such variables. As described in Sections 2.1 and 2.2, propositionalized relational data may be unable to represent key variables that are common causes or common effects. Consequently, propositionalization has a high likelihood of violating causal sufficiency (see Appendix A).

Proposition 1 *Propositionalization can violate causal sufficiency.*

Consider a simple relational domain R with entities A and B, a many-to-many relationship AB, and attributes $A.X_1$, $A.X_2$, $B.Y_1$, and $B.Y_2$. The causal structure contains the

following dependencies: The degree (cardinality) of related B entities to A is a common cause of $A.X_1$ and $A.X_2$ and the degree of related A entities to B is a common cause of $B.Y_1$ and $B.Y_2$. Propositionalization of this relational schema will create a single table from the perspective of either A or B in which each column represents a variable that is formed from the attributes and relational structure of R. For example, from A's perspective there might be variables for $A.X_1$ and $A.X_2$, variables for a function of the attributes Y_1 and Y_2 on related B entities, and one for the structural variable that counts the number of related B entities. In this case, however, the structural variable that is a common cause of $B.Y_1$ and $B.Y_2$ is unrepresentable, which violates causal sufficiency. Propositionalization from the perspective of B will also violate causal sufficiency.

Even if a propositionalized data set is assumed to be causally sufficient, it still may violate an important condition for effective causal learning. In order to connect causal structure to probability distributions, a causal representation must satisfy the causal Markov condition (see Appendix A). Propositionalization techniques are responsible for manipulating and generating variables while a propositional algorithm learns a model over the joint set of variables; in other words, feature construction is decoupled from model construction.

Proposition 2 *Propositionalization can violate the causal Markov condition.*

As currently practiced, the process of propositionalization leads to individual attribute values participating in multiple aggregate variables. This duplication leads to statistical dependence among the values of those aggregates and between any aggregate and the original value. Consider a relational domain R with entities A and B, attributes $A.X$ and $B.Y$, and two different paths through the schema connecting A and B. The sole causal dependence is that $A.X$ causes $B.Y$. Let Y_1 and Y_2 be two aggregates of $B.Y$ created by propositionalizing from the perspective of A through the two paths that connect A to B. The aggregate variables Y_1 and Y_2 are non-descendants but could be correlated since they may include common attribute values. Y_1 and Y_2 have the same parent set $(A.X)$ but will be correlated even after conditioning on those parents due to correlated residual variation. This violates the causal Markov condition.

In addition to supporting the causal Markov condition, causal representations must support the transitive, irreflexive, and antisymmetric nature of causation (Spirtes, Glymour, and Scheines 2000). We show that the DAPER model is capable of representing a causal model over relational data.

Theorem 1 *The DAPER model is a sufficient representation of causality in relational domains.*

Proof Sketch For propositional domains, the directed acyclic graph (DAG) has been shown to be sufficient to represent causal knowledge (Pearl 2000). For relational domains, the set of possible ground graphs of DAPER models is a strict subset of all possible DAGs since the relational representation explicitly encodes parameter tying (Heckerman, Meek, and Koller 2007). Therefore, DAPER is sufficient for representing causality in relational domains. □

Many other relational representations, such as relational Markov networks (Taskar, Abbeel, and Koller 2002) and Markov logic (Richardson and Domingos 2006), are insufficient to represent causality because they fail to satisfy the requirement that causality be directed and antisymmetric.

3 Relational PC

Given the representation requirements detailed in Section 2, we developed the first algorithm, relational PC (RPC),[1] that explicitly learns causal models of relational data. RPC is an extension of the PC algorithm for propositional data (Spirtes, Glymour, and Scheines 2000). The algorithm retains the same essential strategies employed in PC for identifying causal structure, but there are several key innovations that enable learning in relational domains.

3.1 PC algorithm

The PC algorithm takes as input a propositional data set and outputs a partially directed acyclic graph corresponding to the equivalence class of statistically indistinguishable causal models consistent with the data. It is guaranteed to be correct under the three standard assumptions of causal discovery (the causal Markov condition, causal sufficiency, and faithfulness, see Appendix A).

The first phase, skeleton identification, discovers the undirected graphical structure that encodes the set of conditional independencies present in the data. An edge between two variables indicates statistical dependence, whereas the absence of an edge corresponds to marginal or conditional independence. The algorithm iteratively tests all pairs of variables for marginal independence followed by conditional independence over all possible sets of conditioning variables.

The second phase, edge orientation, orients the edges by applying specific constraints that limit the set of causal models to those consistent with the learned correlations from phase I. PC uses three sets of rules for orienting edges (collider detection, known non-colliders, and cycle avoidance, see Appendix B), which exploit the rules of d-separation and acyclicity of causal models.

The PC algorithm is a constraint-based method. An alternative class of approaches, the search-and-score paradigm, searches the space of models for the structure with the highest likelihood over the data. Although these methods accurately model the probability distribution of the data, they are computationally intensive. Since the search space of relational models is much larger than that for propositional data, the need to constrain the search space is even more pressing.

Recent experimental results in the propositional setting provide strong evidence that *hybrid* algorithms (combining constraint-based and search-and-score approaches) are

[1] We implemented RPC in Prolog to identify unit classes, Java for skeleton identification and edge orientation, R for statistical tests, and Postgres for data storage and access. For source code, visit www.kdl.cs.umass.edu/causality.

more effective for learning the structure of directed graphical models than are methods that use only one of the component techniques (Tsamardinos, Brown, and Aliferis 2006). However, all algorithms that learn statistical models of relational data are based exclusively on search-and-score approaches. RPC is the first constraint-based algorithm for learning causal models of relational data, and it can be used alone or as a component of a hybrid algorithm.

3.2 Key differences of relational data

RPC takes as input an ER model represented in first-order logic and the corresponding instantiated relational database. RPC then outputs a partially directed DAPER model representing the equivalence class of statistically indistinguishable causal models. The RPC algorithm retains the essential strategy used in PC; however, extending the algorithm to the relational domain requires several key innovations.

Variable space To learn a causal model of relational data, we must transform the contents of the database into the necessary components for a relational causal learning algorithm. First, we define the notion of a *unit class*. Given a relational database, we can construct a set of unit classes $\mathcal{U} \subset (\mathcal{E} \cup \mathcal{R})^+$ such that each unit class U consists of one or more entity or relationship classes. The combination of entity and relationship classes must be relationally connected, as governed by the ER schema, and they combine to form a subgraph over the data. The unit class is defined from the perspective of a base entity or relationship class, $b(U)$.

A *unit attribute class* is any attribute class defined over an entity or relationship class within the unit. More formally, $\mathcal{A}(U) = \bigcup_{B \in U} \mathcal{A}(B)$. The set of potential causal dependencies we consider is a subset of all possible arc classes over $\mathcal{A}(U) \times \mathcal{A}(U)$. A potential causal dependency is composed of a *treatment* variable $\mathcal{T} \in \mathcal{A}(U)$ and an *outcome* variable $\mathcal{O} \in \mathcal{A}(b(U))$. The constraint on any arc class in this setting is a first-order expression corresponding to the relational path from $b(U)$ to C, where $C \in U$ is the class for which the treatment variable is defined. This is substantially different than causal learning in a propositional setting, in which the set of potential causal dependencies is equivalent to the set of all possible variable pairs.

For example, let unit class $U = \{$AUTHOR (A), AUTHOREDBY (AB), PAPER (P), PUBLISHEDIN (PI), VENUE (V)$\}$ and $b(U) =$ A. Each unit in this unit class consists of an author, the papers that he or she writes, and the venues in which those papers are published. The base item is the author. The set of unit attribute classes, $\mathcal{A}(U)$, could include A.RESEARCH-INTEREST, P.TOPIC, and V.FOCUS, among others. From this set, the only potential causal dependencies considered are $\langle$P.TOPIC, A.RESEARCH-INTEREST$\rangle$ and $\langle$V.FOCUS, A.RESEARCH-INTEREST$\rangle$. A causal dependency with outcome P.TOPIC exists in a separate unit class, in which PAPER is the base.

In relational domains, the set of possible causal dependencies has several restrictions. Since the data are atemporal, no attribute can be time-varying, so treatment and outcome variables must be different. Because the existence of the relationship is a necessary precondition for the attribute itself to exist, relationship attributes cannot be treatment variables when the outcome is the relationship existence.

The first phase of RPC requires identifying the set of possible common causes for the treatment and outcome variables of a potential dependency $\langle A.X, B.Y \rangle$. RPC enumerates all possible pairs of unit attribute classes subject to the aforementioned restrictions. The set of potential common causes for $\langle A.X, B.Y \rangle$ is the union of the set of treatment variables when $A.X$ and $B.Y$ are considered outcomes $\{C.Z | \langle C.Z, A.X \rangle\} \cup \{D.Z | \langle D.Z, B.Y \rangle\}$. The union, rather than the intersection, is required because the set of *direct* common causes of the treatment and outcome may be empty, whereas a direct cause of one could be an *indirect* cause of the other. As the algorithm identifies marginal and conditional independencies, the set of possible common causes reduces because only those for which a possible dependence exists are included.

Aggregates and asymmetry Relational learning algorithms construct variables based on attributes of related entities and relationships. Unless the entity classes are connected via a one-to-one relationship, a unit attribute class consists of set-valued attributes (e.g., the topics of all papers an author writes). As a result, RPC uses aggregation functions, a common technique used in relational learning algorithms, to create a single value for each unit attribute, i.e., $f(A.X)$. Currently these functions are limited to mode and count.

The implications of aggregate causality are ambiguous. It is not entirely clear how a treatment affects an aggregated outcome because the dependence would not specify anything about the individual values that comprise the aggregation. Because of this, RPC limits the definition of potential causal dependencies to those pairs of unit attribute classes for which the outcome is a non-aggregated single value (i.e., an attribute of the base class of the unit). Any found dependencies provide knowledge of the cause of individual values, which themselves can be used as inputs to an aggregate.

Since relational variables may require aggregates, there is an inherent *asymmetry* for pairs of unit attributes depending on the perspective of the base class of the unit. In propositional data, testing the correlation between two variables X and Y is identical to the test of correlation between Y and X. But because of asymmetry, RPC tests the association from both perspectives, $\langle f(X), Y \rangle$ and $\langle f(Y), X \rangle$. If a statistical dependence is detected in either, we conclude that the association exists between X and Y (irrespective of the direction of causality). This provides stronger guarantees under practical sample size limitations for which the power of a test from one perspective may be limited.

Structural variables RPC learns dependencies for the *relational structure* of the data. To correctly learn the causal structure of relational data, existence uncertainty must be explicitly represented, see Section 2.2. Thus, RPC includes existence as an attribute for each relationship class.

Existence uncertainty behaves differently when viewed as a treatment or as an outcome variable. As a treatment, RPC uses the count aggregate to represent cardinality, avoiding the degree disparity problem of relational data identified by

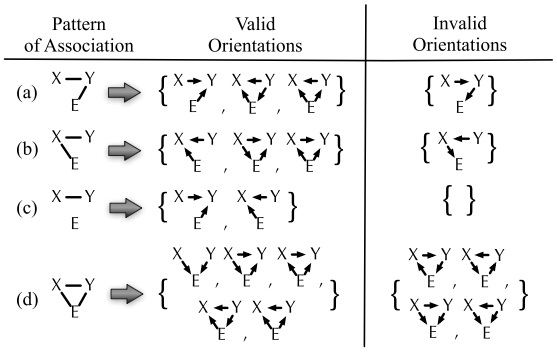

Figure 2: The four sets of possible causal models that can explain a given pattern of association for two relational variables and the existence of a relationship between them.

Jensen, Neville, and Hay (2003). As an outcome, RPC explicitly tests for association between a treatment variable and the existence of the relationship. We rely on statistical details described by Getoor et. al. (2007) to test the association of relationship existence with the treatment variable.

Edge orientation RPC uses the original rules from PC, as well as a new set of constraints, referred to as *restricted existence models* (REM), introduced by the representation of existence uncertainty. There are four unique patterns of association involving two unit attribute classes X and Y and the relationship existence attribute class E for a relationship class on the path between them. We consider only the cases in which a dependence is observed between X and Y. Figure 2 illustrates the four cases and the set of plausible causal models that can explain the observed correlations.

We derive the restricted sets of causal models using acyclicity (atemporal causal models contain no cycles), existence precondition (X can only have an effect on Y if a relational connection between them exists, so E must be a causal precondition for Y), and common effects (existence uncertainty can induce a spurious correlation between X and Y, see Section 2.2). For example, in case (a) the invalid causal model is eliminated because if X caused Y, then the existence precondition would require E (the existence of the relationship between X and Y) to also be a cause of Y.

A direct consequence of implicitly conditioning on existence uncertainty is that relationship existence always appears in the separating set of two relational variables. Consequently, collider detection never applies when the colliding variable is relationship existence. If two variables have a common effect of relationship existence, then a pattern of association could result in the removal of an edge (see the first potential causal model in case (d) in Figure 2). As a result, collider detection can only orient triples ($X \rightarrow Y \leftarrow Z$) for which neither $\langle X, Y \rangle$ nor $\langle Y, Z \rangle$ exhibit associations to a common existence variable.

3.3 Complexity and correctness

In the best case, no associations are present in the data, requiring a single iteration of RPC in $\Omega(n)$ time, where n is the number of potential dependencies. In the theoretical worst case, every pair of variables has every other variable as a potential common cause, and the hypothesis tests never conclude independence. This would require testing every possible marginal and conditional test in $O(n^{d+1})$ time, where d is the size of the maximum conditioning set used in the first phase of RPC. When $d \ll n$, RPC is a polynomial-time algorithm; however, the number of conditional independence tests grows exponentially in d (Spirtes, Glymour, and Scheines 2000). In practice, the algorithm has reasonable average-case complexity for moderately sized data sets. This is in contrast to the computationally intensive search-and-score approach, for which identifying the correct model structure is $\mathcal{NP}$-hard (Chickering 1996). For both propositional and relational data, the size of the space of possible models is exponential with the number of potential dependencies. RPC and other constraint-based algorithms are efficient because they exploit constraints (e.g., conditional independencies) to reduce the space of causal models.

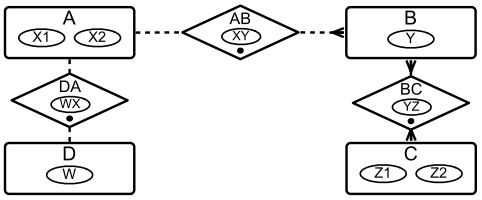

Figure 3: The entity-relationship schema used in experiments includes relationships with different cardinalities.

Theorem 2 *RPC correctly identifies the equivalence class of statistically indistinguishable causal models.*

Proof Sketch The proof is implied by the correctness of both phases of the algorithm and the assumptions listed in Section 3.1.

Phase I correctness: (Extension of proof for PC (Spirtes, Glymour, and Scheines 2000)). Assuming faithfulness, the data support exactly the conditional independencies present in the true model. Given correct hypothesis tests and that RPC runs to a sufficient depth to exhaust the set of potential common causes for each dependency, an edge will be present in the learned DAPER model if and only if it exists in the true model. Unlike in the propositional setting, the correct skeleton includes an edge between two variables if they have a common effect over existence uncertainty since the presence or absence of the edge cannot be differentiated statistically. In this case, REM applied in phase II will remove the edge if that constraint can be uniquely identified.

Phase II correctness: The collider detection and known non-collider rules correctly orient edges given the rules of d-separation. By Theorem 1, the DAPER representation supports d-separation, so these two rules are correct for RPC. The cycle avoidance rule correctly orients edges since the true causal model is known to be acyclic. Finally, the restricted existence models were previously shown to limit the set of causal models to smaller sets of statistically indistinguishable models. □

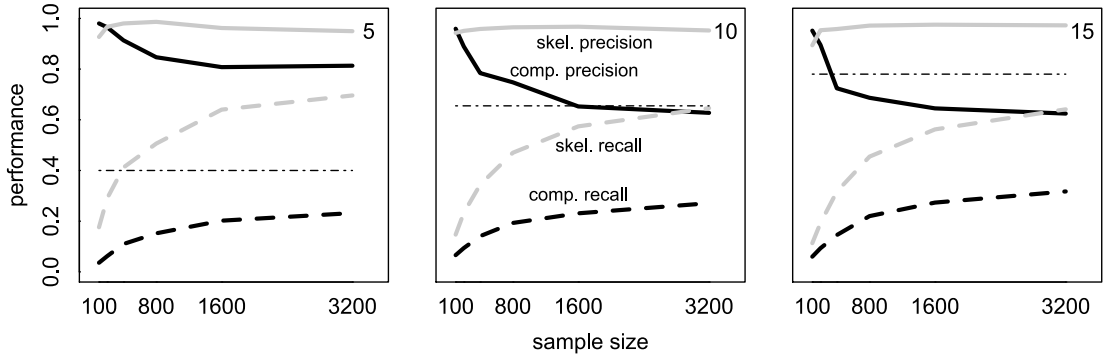

Figure 4: Precision and recall after skeleton identification and edge orientation for synthetic data over increasing sample sizes for 5, 10, and 15 true dependencies. The thin dashed line indicates the ceiling for compelled recall given a correct skeleton.

4 Analysis

To demonstrate the effectiveness of RPC, we evaluated learned models against models with known ground truth. We implemented a simple causal model generator that randomly chooses a valid causal model with a pre-specified number of dependencies. The generator creates conditional probability tables from either a Dirichlet distribution for attributes or a bounded Pareto distribution for relationship existence. Attributes with no direct causes have a uniform prior, while relationships with no causes are generated from a Poisson distribution. We populated the database given the generative model and its underlying distributions. For all synthetic experiments, we used the schema depicted in Figure 3 and limited the set of potential causal dependencies to unit classes with at most five entity and relationship classes.

Figure 4 presents precision and recall after each phase of RPC. Precision is the proportion of edges included in the *learned* model that are correct, while recall is the proportion of edges in the *true* model that are included in the learned model. The results are averages over 20 random causal models with 5, 10, and 15 dependencies and generated data over 5 parameterizations of each model. For each class of causal models, we examine the effect of increasing sample size (from 100 to 3200) by examining precision and recall.

These graphs indicate that RPC makes few type I errors (high skeleton precision), and type II errors are reduced by increasing the power of the statistical tests through larger sample sizes (skeleton recall increases). Skeleton errors made by the algorithm necessarily decrease the precision and recall of the compelled model. Edge orientation rules are guaranteed to be correct only for a correct underlying skeleton. Even with perfect information, only the partially directed model consistent with the conditional independencies can be identified. The thin dashed line in Figure 4 indicates the corresponding ceiling for compelled recall.

False negatives occurring in phase I decrease the compelled precision since the edge orientation rules can incorrectly orient edges with incomplete information. The compelled recall similarly increases with sample size as more undirected edges are discovered in phase I. As the complexity of the true causal model increases, additional erroneous decisions can occur during skeleton identification, which accounts for the drop in performance as we increase the number of true causal dependencies.

For the same randomly generated and parameterized causal models with 10 dependencies as in the previous experiment, we record why each dependency is or is not detected by RPC for each sample size (see Table 1). Each dependency can either be correctly detected, incorrectly insignificant at the marginal or conditional level, or insubstantive (below a strength of effect threshold of 0.1) at the marginal or conditional level. In summation, these five categories completely explain the outcome of all 10 true dependencies. The two contributing factors that lead to error are low sample size and parameterizations with small effects.

Given the practical limitations of identifying dependencies (i.e., finite data samples rather than in the sample limit), we complement the previous set of experiments with a data-independent assessment of the edge orientation rules. We generate a true causal model with varying numbers of dependencies and treat it as the input to phase II of RPC by removing directional information. This provides a correct skeleton for edge orientation to identify the partially directed model consistent with the conditional independencies encoded by the skeleton. For each setting of dependencies, we generate 1,000 causal models and average the precision and recall of the resulting compelled model (see Figure 5).

As expected, precision is 1.0 regardless of the causal model since Theorem 2 guarantees the correctness of the edge orientation rules. The recall, however, increases with the complexity of the causal model. This is a consequence of a chain reaction occurring within phase II; as each edge is oriented, it may inform other edge orientation rules. For example, if collider detection can orient an edge, the REM rule may be able to further limit the set of possible causal models. To determine the applicability and utility of the REM orientation rule, we compare the performance of RPC edge orientation with and without REM. Although the average recall using REM is not significantly greater than without, the rule is still used occasionally. The applicability of REM is infrequent, leading to a minor overall improvement on orientation; however, the constraints on the space of causal models exploited by REM can be useful in limited situations.

Table 1: Breakdown of skeleton dependencies for causal models with 10 true dependencies over increasing sample size.

Sample Size	Detected	Missed Marginal	Missed Cond.	Insubstantive Marginal	Insubstantive Cond.
100	1.47 ± 0.106	7.45 ± 0.140	0.83 ± 0.088	0.21 ± 0.046	0.04 ± 0.020
200	2.33 ± 0.116	6.10 ± 0.142	1.11 ± 0.098	0.45 ± 0.066	0.01 ± 0.010
400	3.44 ± 0.134	4.50 ± 0.138	1.42 ± 0.105	0.62 ± 0.078	0.02 ± 0.020
800	4.70 ± 0.153	3.51 ± 0.128	1.09 ± 0.094	0.64 ± 0.094	0.06 ± 0.024
1600	5.74 ± 0.155	2.49 ± 0.134	1.05 ± 0.091	0.68 ± 0.083	0.04 ± 0.020
3200	6.46 ± 0.152	1.88 ± 0.115	0.69 ± 0.071	0.85 ± 0.090	0.12 ± 0.033

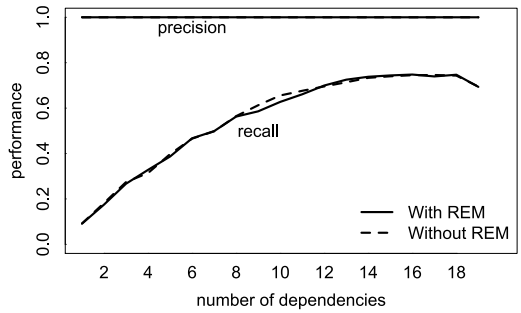

Figure 5: Precision and recall of edge orientation assuming a correct skeleton.

We recorded the frequency with which each edge orientation rule applies. Collider detection is the dominant rule, directing 62–86% of the edges, while the known non-collider rule exhibits relative frequencies of 14–31% of the edges. Cycle avoidance and REM are used the most infrequently, orienting at most 5% and 2% of the edges, respectively.

We applied RPC to the MovieLens+ database, a combination of the UMN MovieLens database (www.grouplens.org) and box office, director, and actor information collected from DBpedia (dbpedia.org) and IMDb (www.imdb.com). Of the 1,285 movies with this additional information, we sampled 10% of the user ratings yielding over 62,000 ratings. RPC generated the model shown in Figure 6. The number of actors in a film influences the movie's budget, as does the age of the director. RPC likely oriented three dependencies incorrectly from the number of ratings or the rating attribute. More plausibly, popular movies and genres have more ratings, and a movie's genre influences those ratings. Temporal information, unavailable to RPC, could be used to improve orientation, a direction we hope to pursue in the future.

We also applied RPC to Rexa, a citation data set of scientific papers in computer science (www.rexa.info). We constructed a subset of over 5,000 authors and 29,000 papers, eliminating entries with missing values. The model shown in Figure 1 is actually the learned model for Rexa, except the edges could not be oriented. RPC discovered associations between paper topics and author research interest, as well as paper topics and venue focus. However, collider detection typically triggers the other edge orientation rules, but did not apply for this data set. We expect RPC to orient edges as more variables are included in the domain.

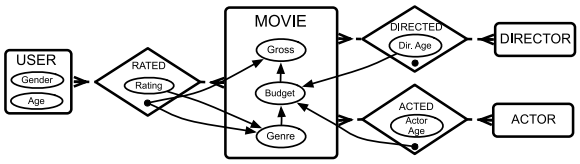

Figure 6: Learned causal model of the MovieLens+ data set.

5 Conclusions and Future Work

This work presented several key algorithmic and theoretical innovations that enable causal discovery in relational domains. We provided strong evidence that effective learning of causal models in relational domains requires a relational representation to explicitly represent potential common causes and effects. We introduced the relational PC algorithm, based on the constraint-based paradigm, that learns causal dependencies of relational domains under certain limited assumptions.

The causal mechanisms underlying real world domains are complex and challenging to learn and represent, as the learned models of the citation database and the movie industry suggest. Hence, there are several avenues to follow for future work. To fully learn the complexity of the real world, we must represent and reason about time. Causal discovery algorithms, such as RPC, can benefit from additional constraints, such as incorporating prior expert knowledge of given domains. Combining the constraint-based approach into a hybrid algorithm could increase the effectiveness of learning the causal structure and better approximate the underlying probability distributions of the data.

Acknowledgments

The authors wish to thank Cindy Loiselle for her helpful comments, as well as Andrew McCallum and the students and staff of the Information Extraction and Synthesis Laboratory at the University of Massachusetts for use of the Rexa data set. This effort is supported by AFRL and MIT Lincoln Laboratory under contract numbers FA8750-09-2-0187 and 7000032881. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of AFRL, MIT or the U.S. Government.

A Causal Assumptions

Three assumptions are commonly made by causal learning algorithms to allow for causal interpretations of learned models (Spirtes, Glymour, and Scheines 2000). We define them here with respect to relational domains as described in Sections 2 and 3.2. For the following definitions, let $\mathcal{V}$ be the variable space derived from database $\mathcal{D}$, let $\mathcal{P}$ be the probability distribution over $\mathcal{V}$, and let $\mathcal{G}$ be the causal structure that encodes the conditional indepenencies present in $\mathcal{V}$.

Definition A.1 (Causal sufficiency) *$\mathcal{V}$ is causally sufficient if and only if for all potential causal dependencies $\langle A.X, B.Y\rangle \in \mathcal{V} \times \mathcal{V}$, all common causes are measured and included in $\mathcal{V}$.*

As discussed in Section 2.1, if there exist latent common causes of two variables, then we may incorrectly conclude causal dependence between them instead of choosing the correct, albeit unrepresented, causal model.

Definition A.2 (Causal Markov condition) *Given that $\mathcal{V}$ is causally sufficient, $\mathcal{P}$ is Markov to $\mathcal{G}$ if and only if each variable $A.X \in \mathcal{V}$ is conditionally independent of its non-effects given its direct causes.*

The causal Markov condition provides a connection between the causal structure and the probability distribution of a set of variables. Without satisfying this condition, two causally unrelated variables may remain correlated after conditioning on all common causes.

Definition A.3 (Faithfulness) *$\mathcal{P}$ is faithful to $\mathcal{G}$ if and only if there exist no conditional independencies in $\mathcal{P}$ that are not entailed by the causal Markov condition on $\mathcal{G}$.*

The faithfulness assumption ensures that the probability distribution will not indicate that two variables are conditionally independent when they are actually dependent according to the causal structure. Together with the causal Markov condition, this guarantees that exactly the conditional independencies entailed by $\mathcal{G}$ exist in $\mathcal{P}$. All three assumptions are enough to prove that a constraint-based algorithm, such as PC or RPC, will discover the correct skeleton over $\mathcal{V}$.

B Edge Orientation Rules

The PC algorithm employs the following set of three edge orientation rules that correctly infer causal dependence from patterns of association (Spirtes, Glymour, and Scheines 2000). These rules are also used in RPC, along with the constraints implied by modeling existence uncertainty, and are subject to the caveats unique to relational data as described at the end of Section 3.2.

Definition B.1 (Collider Detection) *If $X - Y - Z$ and $Y \notin sepset(X, Z)$, then orient as $X \rightarrow Y \leftarrow Z$.*

This rule exploits a concept used in d-separation—two variables become dependent conditional on a common effect. If a third variable Y does not render X and Z conditionally independent yet exhibits association with both of them, it must be a collider.

Definition B.2 (Known Non-Colliders) *If $X \rightarrow Y - Z$ and $\langle X, Y, Z\rangle$ is not a collider, then orient as $X \rightarrow Y \rightarrow Z$.*

Collider detection enables additional orientations. If $\langle X, Y, Z\rangle$ is not oriented as a collider, but X is a known cause of Y, then only a single causal model can explain the association between Y and Z (namely, Y causes Z).

Definition B.3 (Cycle Avoidance) *If $X - Y$ and $X \rightarrow V_1 \cdots \rightarrow V_k \rightarrow Y$, then orient as $X \rightarrow Y$.*

The third rule stems from assuming the data are atemporal and that causality is transitive.

References

Berkson, J. 1946. Limitations of the application of fourfold table analysis to hospital data. *Biometrics Bulletin* 2(3):47–53.

Chickering, D. M. 1996. Learning Bayesian networks is NP-complete. In Fisher, D. H., and Lenz, H.-J., eds., *Learning from Data: Artificial Intelligence and Statistics V*. Springer. 121–130.

Friedkin, N. E. 1998. *A Structural Theory of Social Influence*. New York, NY: Cambridge University Press.

Getoor, L.; Friedman, N.; Koller, D.; Pfeffer, A.; and Taskar, B. 2007. Probabilistic relational models. In Getoor, L., and Taskar, B., eds., *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press. 129–174.

Heckerman, D.; Meek, C.; and Koller, D. 2007. Probabilistic entity-relationship models, PRMs, and plate models. In Getoor, L., and Taskar, B., eds., *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press. 201–238.

Holland, P. W., and Rubin, D. B. 1988. Causal inference in retrospective studies. *Evaluation Review* 12(3):203–231.

Jensen, D.; Neville, J.; and Hay, M. 2003. Avoiding bias when aggregating relational data with degree disparity. In *Proceedings of the 20th ICML*, 274–281.

Kramer, S.; Lavrač, N.; and Flach, P. 2001. Propositionalization approaches to relational data mining. In Džeroski, S., and Lavrač, N., eds., *Relational Data Mining*. New York, NY: Springer-Verlag. 262–286.

Pearl, J., and Verma, T. 1991. A theory of inferred causation. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, volume 11, 441–452.

Pearl, J. 2000. *Causality: Models, Reasoning, and Inference*. New York, NY: Cambridge University Press.

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

Spirtes, P.; Glymour, C.; and Scheines, R. 2000. *Causation, Prediction and Search*. Cambridge, MA: MIT Press, 2nd edition.

Taskar, B.; Abbeel, P.; and Koller, D. 2002. Discriminative probabilistic models for relational data. In *Proceedings of the 18th UAI*, 485–492.

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.