Learning the Structure of Causal Models with Relational and Temporal Dependence

Katerina Marazopoulou
kmarazo@cs.umass.edu

Marc Maier
maier@cs.umass.edu

David Jensen
jensen@cs.umass.edu

College of Information and Computer Sciences
University of Massachusetts Amherst
Amherst, MA 01003

Abstract

Many real-world domains are inherently *relational* and *temporal*—they consist of heterogeneous entities that interact with each other over time. Effective reasoning about causality in such domains requires representations that explicitly model relational and temporal dependence. In this work, we provide a formalization of temporal relational models. We define temporal extensions to abstract ground graphs—a lifted representation that abstracts paths of dependence over all possible ground graphs. Temporal abstract ground graphs enable a sound and complete method for answering *d*-separation queries on temporal relational models. These methods provide the foundation for a constraint-based algorithm, TRCD, that learns causal models from temporal relational data. We provide experimental evidence that demonstrates the need to explicitly represent time when inferring causal dependence. We also demonstrate the expressive gain of TRCD compared to earlier algorithms that do not explicitly represent time.

1 INTRODUCTION

Recent work in artificial intelligence has devoted increasing attention to learning and reasoning with causal knowledge. Causality is central to understanding the behavior of complex systems and to selecting actions that will achieve particular outcomes. Thus, causality is implicitly or explicitly central to mainstream AI areas such as planning, cognitive modeling, computational sustainability, game playing, multiagent systems, and robotics. Causal inference is also central to many areas beyond AI, including medicine, public policy, and nearly all areas of science.

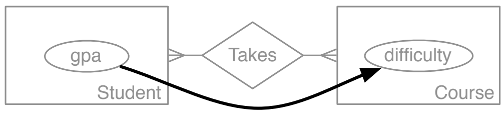
Substantial research advances have been made over the past several decades that allow the structure and parameters of causal graphical models to be learned from observational data. As early as the 1990s, researchers developed constraint-based algorithms, such as the IC [Pearl and Verma, 1991] and PC [Spirtes *et al.*, 2000] algorithms, that leverage the connection between the graphical criterion of *d*-separation and the conditional independencies that are inferred to hold in the underlying distribution, in order to learn the structure of causal graphical models from data.

In this work, we significantly extend the expressiveness of the models learnable with constraint-based algorithms. Specifically, we provide a formalization of temporal relational models, an expressive class of models that can capture probabilistic dependencies between variables on different types of entities within and across time points. We extend the notion of abstract ground graphs [Maier *et al.*, 2013b]—a lifted representation that allows reasoning about the conditional independencies implied by a relational model—for temporal relational models, and we show that temporal abstract ground graphs are a sound and complete abstraction for ground graphs of temporal relational models. Temporal abstract ground graphs can be used to answer *d*-separation queries for temporal relational models. We also extend an existing constraint-based algorithm for inferring causal dependence in relational data—the relational causal discovery (RCD) algorithm—to incorporate time, thus providing a constraint-based method that learns causal models from temporal relational data.

2 RELATIONAL MODELS

Propositional representations, such as Bayesian networks, describe domains containing a single entity class. Many real world systems comprise instances from multiple entity classes whose variables are interdependent. Such domains are often referred to as *relational*. In this section, we introduce basic concepts of relational representations that exclude time.

A *relational schema* $\mathcal{S} = (\mathcal{E}, \mathcal{R}, \mathcal{A}, card)$ specifies the types of entities, relationships, and attributes that exist in a domain. It includes a cardinality function that constrains



$[Course, Takes, Student].gpa \rightarrow [Course].difficulty$

Figure 1: Relational model for the example domain. The underlying relational schema (ER diagram) is shown in gray.

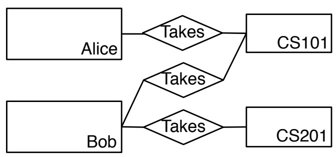


Figure 2: Example relational skeleton for the schema shown in Figure 1.

the number of times an entity instance can participate in a relationship. A relational schema can be depicted with an Entity-Relationship (ER) diagram. Figure 1 shows an example ER diagram (in gray) that describes a simplified university domain. The domain consists of two entity classes ($Student$ and $Course$), and one relationship class ($Takes$). The entity class $Student$ has one attribute, gpa, and the entity class $Course$ has one attribute, $difficulty$. The cardinality constraints are shown with crow's feet notation—students can enroll in multiple courses and a course can be taken by multiple students. A *relational skeleton* is a partial instantiation of a relational schema. It specifies the entity and relationship instances that exist in the domain. Figure 2 shows an example relational skeleton for the relational schema of Figure 1. The skeleton consists of two $Student$ instances, Alice and Bob, and two $Course$ instances, CS101 and CS201. Alice is taking CS101 and Bob is taking both courses.

Given a relational schema, we can specify *relational paths*, which intuitively correspond to ways of traversing the schema. For the schema shown in Figure 1, possible paths include $[Student, Takes, Course]$ (the courses a student takes), as well as $[Student, Takes, Course, Takes, Student]$ (other students that take the same courses). *Relational variables* consist of a relational path and an attribute that can be reached through that path. For example, the relational variable $[Student, Takes, Course].difficulty$ corresponds to the difficulty of the courses that a student takes. Probabilistic dependencies can be defined between relational variables. Dependencies are said to be in canonical form when the path of the effect (or *outcome*) relational variable is a single item. For canonical dependencies, the path of the cause (or *treatment*) relational variable describes how dependence is induced. As an example, consider the

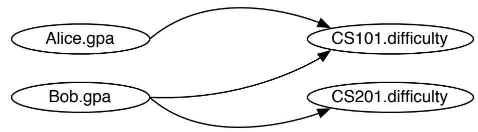


Figure 3: Ground graph for the model of Figure 1 applied on the relational skeleton of Figure 2.

following *relational dependency*

$$[Course, Takes, Student].gpa \rightarrow [Course].difficulty$$

which states that the *difficulty* of a course is affected by the *gpa* of students taking that course. Presumably, instructors adjust the difficulty of the course based on the grade-point average of enrolled students.

A *relational model* $\mathcal{M} = (\mathcal{S}, \mathcal{D}, \Theta)$ is a collection of relational dependencies defined over a single relational schema along with their parameterizations (a conditional probability distribution for each attribute given its parents). The structure of a relational model can be depicted by superimposing the dependencies on the ER diagram of the relational schema, as shown in Figure 1, and labeling each arrow with the corresponding relational dependency.

Given a relational model $\mathcal{M}$ and a relational skeleton σ, we can construct a *ground graph* $GG_{M\sigma}$ by applying the relational dependencies as specified in the model to the specific instances of the relational skeleton. Figure 3 shows the ground graph for the model of Figure 1 applied on the relational skeleton of Figure 2. In this work, we restrict our attention to relational models that do not contain the kind of relational autocorrelation that gives rise to cycles in the ground graph.

3 TEMPORAL RELATIONAL MODELS

Relational models can be extended with a temporal dimension to model probabilistic dependencies over time. Such an extension is similar to the way in which dynamic Bayesian networks (DBNs) extend Bayesian networks [Murphy, 2002]. A temporal relational model can be thought of as a sequence of time points, each of which is associated with a (non-temporal) relational model, and a set of dependencies that cross time points. Because dependencies in this model have a causal interpretation, dependencies across time points are only directed from the past to the future.

In this section, we extend the relational notions presented in Section 2 to include time. We assume that (1) time is discrete; (2) the schema is static; (3) relational dependencies do not change over time; (4) the temporal relational skeleton is given *a priori*; (5) the first-order Markov assumption

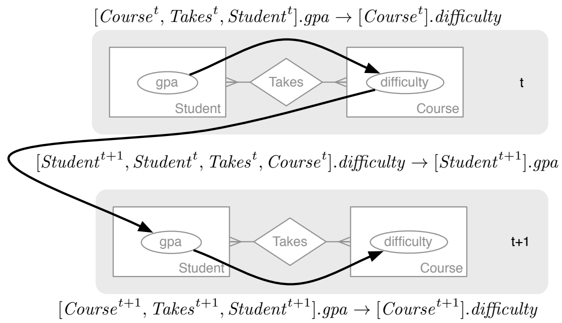


Figure 4: Example structure of a temporal relational model.

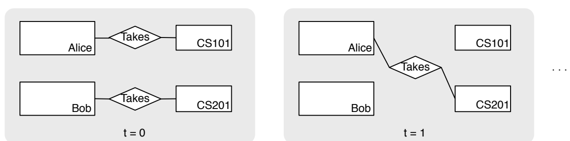


Figure 5: Example temporal relational skeleton for the schema shown in Figure 4.

holds (i.e., treatment and outcome can be at most one time point apart); and (6) all entities participating in a relationship are contemporaneous with the relationship.

Under these assumptions, the structure of a temporal relational model can be represented by using only two time points, as shown in Figure 4. Every time point has the same relational schema, shown in gray. A *temporal relational skeleton* provides a partial instantiation of a temporal relational schema. It specifies the entity and relationship instances that exist in each time point. Note that different sets of entity and relationship instances may be present in each time step. An example temporal relational skeleton is shown in Figure 5. In this case, both time points have the same set of entity instances (Alice and Bob are instances of the $Student$ entity, CS101 and CS201 are instances of the $Course$ entity). However, the instances of the relationship $Takes$ differ. In $t = 0$, Alice takes CS101 and Bob takes CS201. In $t = 1$, Alice takes CS201 and Bob takes no classes.

Temporal relational paths capture possible ways to traverse the temporal schema; therefore, they can cross time points. For example, a temporal relational path is $[Student^{t+1}, Student^t, Takes^t, Course^t]$, which describes the classes that a student took in the previous semester. More formally, a temporal relational path is a sequence of non-temporal relational paths (relational paths within a time point) and "jumps" between neighboring time points. These jumps can happen at both entities and relationships because each choice encodes a distinct semantics. For example, the relational path $[Student^{t+1}, Student^t, Takes^t, Course^t]$ describes the courses that a student took in the previous semester, while the path $[Student^{t+1}, Takes^{t+1}, Course^{t+1}, Course^t]$ first finds the courses that a student is taking this semester, and then finds those courses in the previous semester. In the example skeleton of Figure 5, for Alice, the first path would reach CS101 at $t = 0$, while the second path, will reach CS201 at $t = 0$.

Definition 1. A *temporal relational path* P is a sequence of non-temporal relational paths $P_0^{t_0}, \ldots, P_k^{t_k}$ ($k \geq 0$) such that for any two consecutive paths $P_i^{t_i}, P_j^{t_j}$ in P the following hold:

1. $|t_i - t_j| = 1$
2. The last item class of $P_i^{t_i}$ is the same as the first item class of $P_j^{t_j}$.
3. No subpath of P is of the form $[I_k^t, \ldots, I_k^t]$, where all relations in the subpath are one-to-one.

We use the notation $time(P)$ to denote the set of all time points that appear in path P.

Temporal relational variables consist of a temporal relational path and an attribute that can be reached through that path. *Temporal relational dependencies* define probabilistic dependencies between two temporal relational variables. Temporal probabilistic dependencies are never directed backwards in time. Therefore, at the model level, there are no dependencies going back in time. However, the temporal constraints associated with a dependency are also implicitly encoded by the temporal relational path that annotates the dependency. To account for this, we forbid the temporal relational path of the treatment to go through any time points later than the time point of the outcome.

Definition 2. A *temporal relational dependency* consists of two temporal relational variables with a common base item, $[I_1^t, \ldots, I_k^{t'}].V_k \rightarrow [I_1^t].V_1$ such that

$$\max\Big(time\big([I_1^t, \ldots, I_k^{t'}]\big)\Big) \leq t$$

The first-order Markov assumption for temporal causal models implies that for every probabilistic dependency, if the treatment is in time point t, then the outcome is either in t or in $t + 1$. In the case of relational domains, the relational path of the treatment carries some temporal information since it can contain multiple time points. For the first-order Markov condition to hold, we require the relational path of the treatment to only go through the current and the next time points. More formally, a temporal relational dependency $[I_1^t, \ldots, I_k^{t'}].V_k \rightarrow [I_1^t].V_1$ follows the first-order Markov assumption if the following two conditions hold:

$$t = t' \text{ or } t = t' + 1 \quad (1)$$

$$time([I_1^t, \ldots, I_k^{t'}]) \subseteq \{t, t-1\} \quad (2)$$

The structure of a 2-slice temporal relational model is defined as a set of temporal relational dependencies over a relational schema.

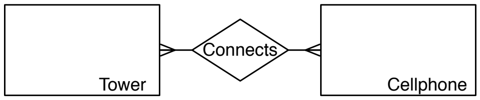


Figure 6: Relational schema for the reality mining dataset.

Definition 3. The structure of a *2-slice temporal relational model* is a pair $\mathcal{M} = \langle \mathcal{S}, \mathcal{D}_T \rangle$, where $\mathcal{S}$ is a relational schema and $\mathcal{D}_T$ is a set of temporal relational dependencies that adhere to the first-order Markov assumption.

Figure 4 shows the structure of a 2-slice temporal relational model for the university domain. The temporal dependency shows that the difficulty of a course in the fall semester affects the spring GPA of the students that took this class in the fall. If, for example, a student takes many difficult classes, it is more likely that the student's GPA will drop in the next semester. Finally, given a temporal relational skeleton σ_T and a temporal relational model $\mathcal{M}$, we can construct a *temporal ground graph* $GG_{\mathcal{M}\sigma_T}$ in the same way as in the non-temporal case.

4 EXPRESSIVENESS OF TEMPORAL RELATIONAL MODELS

The temporal relational model described so far subsumes propositional directed networks (Bayesian networks), propositional temporal directed networks (dynamic Bayesian networks), and relational non-temporal models. This added expressivity comes at the cost of increased complexity. This raises an obvious and important question: What is the value of this added expressivity?

As an example of the practical utility of this added expressivity, we consider a real dataset and show how a path with temporal jumps leads to different terminal sets. Specifically, we used the Reality Mining dataset [Eagle and Pentland, 2006]. The dataset contains two entities, $Tower$ and $Cellphone$, and one relationship, $Connects$. The relational schema for this domain is shown in Figure 6. For this dataset, entity instances (i.e., the set of towers and cellphones) do not change over time. However, the set of relationship instances (the connections) are different at every time point. In total, there are 95 cellphones, 32,579 towers, and 3,308,709 connections. Every connection is timestamped with precision of 1 second. For our work, the time granularity was coarsened to a day.

We computed the terminal sets for the following three paths:

$P1 : [Tower^{t+1}, Tower^{t}, Connects^{t}, Cellphone^{t}]$

$P2 : [Tower^{t+1}, Connects^{t+1}, Connects^{t}, Cellphone^{t}]$

$P3 : [Tower^{t+1}, Connects^{t+1}, Cellphone^{t+1}, Cellphone^{t}]$

The first path corresponds to the cellphones that were connected to a tower in the previous timestep. The second path corresponds to the cellphones that connected to a tower both in the current and in the previous timestep. The third path corresponds to the cellphones that connected to a tower in the current time step, and gets the state of those cellphones in the previous timestep.

Table 1: Jaccard distance between the terminal sets of the different paths for the reality mining dataset (100 sample dates, distance is averaged across dates and across towers).

Jaccard distance	P1 vs. P3	P1 vs. P2	P2 vs. P3
mean	0.47	0.31	0.31
min	0	0	0
max	1	1	1
median	0.5	0	0

We randomly selected 100 dates from the dataset. For each of these dates and for each tower that was used in these dates, we computed the terminal sets for the above paths. For a given tower and date, we computed the Jaccard distance between the different terminal sets. The Jaccard distance between two sets A and B is defined as $J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$. Intuitively, this quantifies the overlap of two sets, while accounting for the size of both. Table 1 shows the average Jaccard distance between the terminal sets, averaged across the dates and the towers. The results indicate that, on average, the terminal sets reached through the three paths will be different; therefore, this more expressive representation could be of use in real data.

5 TEMPORAL ABSTRACT GROUND GRAPHS

An abstract ground graph is a lifted representation that abstracts paths of dependence over all possible ground graphs for a given relational model [Maier *et al.*, 2013b]. Abstract ground graphs are shown to be sound and complete in the sense that every edge in the abstract ground graph corresponds to an edge in some ground graph, and every edge in an arbitrary ground graph is represented by an edge in an abstract ground graph. In this section we adapt the definition of abstract ground graphs for the case of a 2-slice temporal relational model.

Definition 4. A *temporal abstract ground graph* $tAGG_{\mathcal{M}Bh} = \langle V, E \rangle$ for a 2-slice temporal relational model $\mathcal{M} = \langle \mathcal{S}, \mathcal{D}_T \rangle$, perspective $B \in \mathcal{E} \cup \mathcal{R}$, and hop threshold $h \in \mathbb{N}^0$ is an abstraction of the dependencies $\mathcal{D}_T$ for all possible ground graphs $GG_{\mathcal{M}\sigma_T}$ of $\mathcal{M}$ on arbitrary temporal skeletons σ_T. The temporal abstract ground graph is a directed graph with the following nodes and edges:

1. $V = RV \cup IV$, where

(a) RV is the set of *temporal relational variables* with a path of length at most $h+1$.

$$RV = \{[B^t, \ldots, I_j^{t'}].V \mid \mathit{length}([B^t, \ldots, I_j^{t'}]) \leq h+1\}$$

(b) IV are *intersection variables* between pairs of temporal relational variables that could intersect[1].

$$\begin{aligned} IV = \{X \cap Y \mid\ & X, Y \in RV \\ & \text{and } X = [B^t, \ldots, I_k^{t'}, \ldots, I_j^{t''}].V \\ & \text{and } Y = [B^t, \ldots, I_l^{t'''}, \ldots, I_j^{t''}].V \text{ and } I_k^{t'} \neq I_l^{t'''}\} \end{aligned}$$

2. $E = RVE \cup IVE$, where

(a) $RVE \subset RV \times RV$ are the *relational variable edges*:

$$\begin{aligned} RVE = \{[B^t, \ldots, I_k^{t'}].V_k \rightarrow [B^t, \ldots, I_j^{t''}].V_j \mid \\ [I_j^{t''}, \ldots, I_k^{t'}].V_k \rightarrow [I_j^{t''}].V_j \in \mathcal{D}_T \text{ and} \\ [B^t, \ldots, I_k^{t'}] \in \mathit{extend}([B^t, \ldots, I_j^{t''}], [I_j^{t''}, \ldots, I_k^{t'}])\} \end{aligned}$$

(b) $IVE \subset \big(IV \times RV\big) \cup \big(RV \times IV\big)$ are the *intersection variable edges*. This is the set of edges that intersection variables "inherit" from the relational variables that they were created from.

The extend method converts dependencies of the model, specified in the canonical form, into dependencies from the perspective of the abstract ground graph.

Temporal abstract ground graphs can be shown to be a correct abstraction over all possible temporal ground graphs. The proof follows the one provided for the non-temporal abstract ground graphs, as presented by Maier *et al.* [2013b].

The temporal abstract ground graph for a model on the student-courses domain with the dependency

$$[\mathit{Student}^{t+1}, \mathit{Student}^{t}, \mathit{Takes}^{t}, \mathit{Course}^{t}].\mathit{difficulty} \rightarrow [\mathit{Student}^{t+1}].\mathit{gpa}$$

is shown in Figure 7. This abstract ground graph is from the perspective of $\mathit{Student}$ and for hop threshold $h = 4$. Disconnected nodes are omitted.

6 d-SEPARATION IN TEMPORAL ABSTRACT GROUND GRAPHS

The rules of d-separation provide a graphical criterion that specifies whether two sets of variables in a directed acyclic graph are conditionally independent given a third set of variables. Specifically, let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be disjoint sets of variables in a directed acyclic graph. $\mathbf{X}$ is d-separated from $\mathbf{Y}$ given $\mathbf{Z}$ if every path between variables in $\mathbf{X}$ and $\mathbf{Y}$ is not a d-connecting path given $\mathbf{Z}$. A path is d-connecting given $\mathbf{Z}$ if for every collider W on the path, either $W \in \mathbf{Z}$ or a descendant of W is in $\mathbf{Z}$, and every non-collider on the path is not in $\mathbf{Z}$. Geiger and Pearl [1988] and Verma and Pearl [1988] showed that d-separation is sound and complete.

[1] Only relational variables in the same time point can intersect.

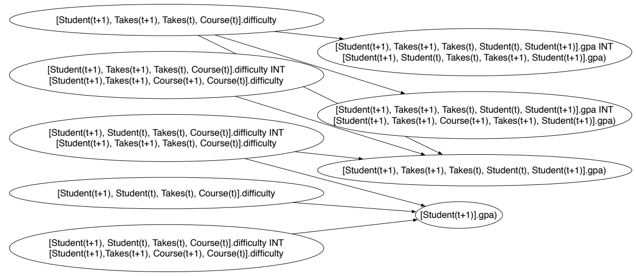


Figure 7: Temporal abstract ground graph from the perspective of $\mathit{Student}$ and hop threshold $h = 4$ for a model with one dependency: $[\mathit{Student}^{t+1}, \mathit{Student}^{t}, \mathit{Takes}^{t}, \mathit{Course}^{t}].\mathit{difficulty} \rightarrow [\mathit{Student}^{t+1}].\mathit{gpa}$. INT denotes intersection variables between pairs of temporal relational variables.

Ground graphs (and temporal ground graphs) are directed acyclic graphs; therefore, the rules of d-separation can be applied to them. In the case of domains where the first-order Markov assumption holds, a d-separating set for variables in the same time point can be found by examining only one time point in the past.

Proposition 5. Let G be a temporal directed acyclic graph that follows the first-order Markov assumption. (i) If X^{t+1} and Y^{t+1} are conditionally independent given some set $\mathbf{Z}$, then there exists a separating set $\mathbf{W}$ such that $\mathit{time}(\mathbf{W}) \subseteq \{t, t+1\}$. (ii) Similarly, if X^t and Y^{t+1} are conditionally independent given some set $\mathbf{Z}$, then there exists a separating set $\mathbf{W}$ such that $\mathit{time}(\mathbf{W}) \subseteq \{t, t+1\}$.

Proof. For each case, we will construct an appropriate separating set:

(i) Let $\mathbf{W} = \mathit{parents}(X^{t+1}) \cup \mathit{parents}(Y^{t+1})$. Because of the first-order Markov assumption, $\mathit{time}(\mathbf{W}) \subseteq \{t, t+1\}$. Moreover, conditioning of $\mathbf{W}$ renders X^{t+1}, Y^{t+1} independent because of the local Markov property (either X^{t+1} is a descendant of Y^{t+1} or vice versa, or none of them is a descendant of the either).

(ii) In this case, let $\mathbf{W} = \mathit{parents}(Y^{t+1})$. Because of the temporal semantics, X^t is a non-descendant of Y^{t+1}; therefore, Y^{t+1} is conditionally independent of its non-descendants (that include X^t) given $\mathbf{W}$.

□

The significance of the above proposition is that, given a model where the first-order Markov assumption holds, we could infer conditional independencies (and use them to learn the structure of the model) by only considering consecutive time points.

Maier *et al.* [2013b] showed that d-separation cannot be applied directly to a relational model. To correct for that, they introduced *relational d-separation*, a graphical criterion that can be applied to abstract ground graphs and used to infer conditional independencies that hold across all possible ground graphs of the model. Here, we show that the notion of relational d-separation can be generalized for temporal abstract ground graphs as well. In the following definition, $X|_b$ denotes the terminal set of X starting at b, i.e., the set of X instances that can be reached if we start from instance b and follow the relational path of X on the relational skeleton.

Definition 6 (Temporal relational d-separation)**.** Let $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ be three disjoint sets of temporal relational variables from perspective B for a 2-slice temporal relational model $\mathcal{M}$ such that not both $\mathbf{X}$ and $\mathbf{Y}$ contain variables in t. Then $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ if and only if, for any temporal skeleton σ_T, $\mathbf{X}|_b$ and $\mathbf{Y}|_b$ are d-separated by $\mathbf{Z}|_b$ in ground graph $GG_{\mathcal{M}\sigma_T}$ for all $b \in \sigma_T(B)$.

The following theorem shows that temporal relational d-separation is sound and complete up to a specified hop threshold. We use the notation $\bar{\mathbf{X}}$ to denote the set of relational variables $\mathbf{X}$ augmented with the set of intersection variables they participate in.

Theorem 7. Let $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ be three disjoint sets of temporal relational variables for perspective B such that not both $\mathbf{X}$ and $\mathbf{Y}$ contain variables in t. Then, for any temporal skeleton σ_T and for all $b \in \sigma(B)$, $\mathbf{X}|_b$ and $\mathbf{Y}|_b$ are d-separated by $\mathbf{Z}|_b$ up to h in ground graph $GG_{\mathcal{M}\sigma_T}$ if and only if $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ are d-separated by $\bar{\mathbf{Z}}$ on the abstract ground graph $tAGG_{\mathcal{M}Bh}$.

Proof. We prove this by defining a non-temporal relational model that is equivalent to the given temporal model. Since d-separation is sound and complete for non-temporal relational models, the result extends to the equivalent model, and therefore, the temporal relational model. Given a 2-slice temporal relational model $\mathcal{M}_T = \langle \mathcal{S}, \mathcal{D}_T \rangle$, construct a relational model $\mathcal{M} = \langle \mathcal{S}', \mathcal{D} \rangle$ as follows:

- For every item in $\mathcal{S}$, add an item in $\mathcal{S}'$ with superscript t and one with superscript $t+1$: $\mathcal{S}' = \{I^t, I^{t+1} \mid I \in \mathcal{S}\}$.
- For every entity $E \in \mathcal{S}$, add in $\mathcal{S}'$ a 1-1 relation between E^t and E^{t+1}.
- The set of relational dependencies is the set of temporal relational dependencies: $\mathcal{D} = \mathcal{D}_T$.

It can be shown that these two models are equivalent in the sense that there is a one-to-one correspondence between valid paths in $\mathcal{M}_T$ and $\mathcal{M}$. Leveraging the temporal constraints of d-separation described by Proposition 5 and the soundness and completeness of d-separation for abstract ground graphs, we conclude that d-separation is sound and complete (up to a hop threshold) in temporal abstract ground graphs. □

7 TEMPORAL RCD

The theory of temporal relational d-separation allows us to derive all conditional independence facts that are consistent with the structure of a temporal relational model, the same way that d-separation connects the structure of a Bayesian network and the conditional independencies of the underlying distribution. This is precisely the connection that constraint-based algorithms leverage in order to learn the structure of models. Thus, temporal relational d-separation (and temporal abstract ground graphs) enable a constraint-based algorithm, TRCD[2], that learns the structure of temporal and relational causal models from data.

TRCD extends RCD [Maier *et al.*, 2013a] to operate over a 2-slice temporal relational model. More specifically, it constructs a set of temporal abstract ground graphs, one from the perspective of each entity, and uses the theory of temporal relational d-separation on temporal abstract ground graphs to decide conditional independence facts. TRCD uses the temporal abstract ground graphs to determine which conditional independence facts should be checked in the data and which dependencies (edges) in the temporal relational model are implied by those facts. As in the case of RCD, for practical reasons, the space of potential dependencies is limited by a domain-specific hop threshold.

TRCD operates in two phases. Phase I learns a set of undirected dependencies and Phase II employs a set of orientation rules to orient those dependencies. Phase II of TRCD uses the same orientation rules as RCD—collider detection, known non-colliders (KNC), cycle avoidance (CA), Meek rule 3 (MR3), and relational bivariate orientation (RBO). Additionally, TRCD orients dependencies that cross time points from the past to the future.

RCD was shown to be sound and complete in the sample limit (i.e., with perfect tests of conditional independence) and for infinite hop threshold, under the standard assumptions of the causal Markov condition, faithfulness, and causal sufficiency for relational domains. By leveraging the soundness and completeness of temporal relational d-separation, TRCD can be shown to be sound and complete (under the same assumptions).

[2]Code available at `kdl.cs.umass.edu/trcd`.

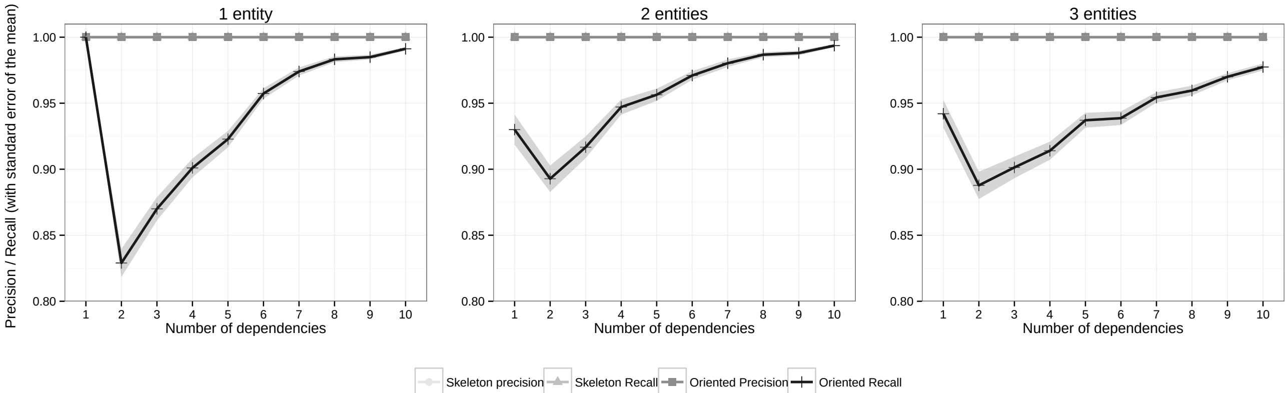


Figure 8: Precision and recall for TRCD after Phase I (unoriented) and after Phase II (oriented) when using an oracle for answering d-separation queries. The y-axis values start at 0.8.

8 EXPERIMENTS

8.1 ORACLE EXPERIMENTS

The goal of this set of experiments is twofold. First, we evaluate the theoretical performance of TRCD in the large sample limit. Towards that end, we employ an oracle for answering d-separation queries in the place of statistical tests of independence. Second, these experiments provide experimental evidence about the correctness of temporal relational d-separation.

We generated synthetic schemas with number of entities varying from 1 to 3, number of relationships fixed to one less than the number of entities, and number of attributes for each item drawn from $Poisson(\lambda = 1) + 1$. The cardinalities were uniformly selected at random. For each number of entities specified, we generated 500 different schemas. This generating process yielded 1,500 different schemas. For a given schema, we generated 10 different models with number of dependencies ranging from 1 to 10. Specifically, we generated all possible dependencies, up to hop-threshold $h = 3$. Then, we chose the desired number of dependencies from that space at random, subject to the following constraints: Each relational variable has at most 3 parents and the generated model contains no cycles. Moreover, every model must contain at least one dependency with a temporal relational path that contains more than one time point. This procedure resulted in a total of 15,000 models.

We ran TRCD with a d-separation oracle for each model. Figure 8 shows average precision and recall after Phase I (unoriented dependencies) and after Phase II (partially oriented dependencies). As expected, given that these experiments use an oracle, the algorithm always learns the correct set of unoriented dependencies (unoriented precision and recall are always 1). The algorithm makes no mistakes in the orientation (oriented precision is always 1); however,

Table 2: Frequency of the most-used orientation rules during Phase II of TRCD for the oracle experiments. For the rest of the rules, the frequency was less than 1%. Temporal dependencies are not oriented by the orientation rules.

Number of entities	Collider detection	KNC	RBO	Percent of temporal dependencies
1	71%	28%	0%	66%
2	66%	11%	23%	68%
3	53%	11%	36%	65%

it is not possible to orient all dependencies (oriented recall is lower than 1). Note that comparing to an oracle version of RCD is not straightforward. The oracle requires a true model that is fully directed. Converting the true temporal model to a non-temporal one would often result in cycles or undirected edges, and the relational d-separation oracle cannot be used.

Table 2 shows how often each orientation rule was used in Phase II. Collider detection, known non-colliders (KNC), and relational bivariate orientation (RBO) are orienting the majority of the edges. As expected, in the case of propositional models (one entity), the relational bivariate orientation rule, a rule for edge orientation that is unique to relational data, is never used. We observe that the other two rules—cycle avoidance and Meek's rule 3—do not fire often. That can be explained by the fact that those rules would never fire in the presence of a temporal edge. Consider cycle avoidance in the propositional case: If the pattern $X \rightarrow Y \rightarrow Z$ and $X - Y$ is encountered, then cycle avoidance orients $X \rightarrow Y$. If any of the dependencies $X \rightarrow Y \rightarrow Z$ crosses time points, then X and Y would be in different time points and the edge between them would be oriented based on temporal precedence. A similar argument can be made for the case of Meek's rule 3.

8.2 EXPERIMENTS ON SYNTHETIC DATA

This experiment showcases the use of TRCD on data, i.e., without the use of an oracle to decide conditional independence. Towards that end, we generated synthetic models, generated data from these models, and applied TRCD on them. The use of synthetic data allows us to have access to the ground truth, so we can measure the accuracy of the learned models. The data-generating process is described in detail below.

Using the same process as described in 8.1, we generated 5 synthetic schemas with 2 entities and 5 synthetic schemas with 3 entities. For each schema, we generated 10 models with number of dependencies ranging from 1 to 10. This resulted in 100 different models. For each model we created 3 different relational skeletons over 300 timepoints. The number of entity instances at each time point was drawn from $Poisson(\lambda) + 1$, where $\lambda \sim \mathcal{U}(5, 10)$. The degree distribution for the relationship instances was drawn from $Poisson(\lambda)+1$, where $\lambda \sim \mathcal{U}(1, 5)$. Regarding the parameters of the graphical model, the marginals were parameterized as normal distributions $\mathcal{N}(\mu, \sigma)$, where $\mu \sim \mathcal{U}(0, 5)$ and $\sigma \sim \mathcal{U}(0, 1)$. The conditional distribution for a relational variable X was $\sum_{Y \in parents(X)} \big(avg(Y)\big) + 0.1 * \mathcal{N}(0, 1)$. This resulted in 300 datasets, each over 300 time points.

In order to assess statistical independence, we fitted a standard linear least-squares regression equation to the outcome variable using the treatment and the variables in the conditioning set as covariates. For relational variables in the conditioning set, we used the average as the aggregation function. Then, we directly used the t-test of the coefficient of the treatment variable to assess independence ($p > 0.05$ or effect size < 0.01). Figure 9 shows average precision and recall of TRCD after Phase I and Phase II, when applied to the synthetic datasets. While precision after Phase I is more than 0.75 in most cases, the recall after Phase I is relatively low. That implies that we concluded independence (and therefore we removed an edge) more often than we should. This corresponds to Type II errors and can be attributed to the lack of good conditional independence tests for relational data.

Finally, to demonstrate the difference in expressiveness between TRCD and RCD (the only constraint-based algorithm for relational data), we ran RCD on a "temporally flattened"[3] version of the synthetic data. The true model and the model that TRCD learned are shown in Figure 10. The model learned by RCD is shown in Figure 11. TRCD correctly learns and orients three of the edges, with the correct path specification. Those dependencies cannot even be expressed in the space of dependencies for RCD.

[3] RCD is ignoring temporal information. An instance is uniquely identified by instance id and time point.

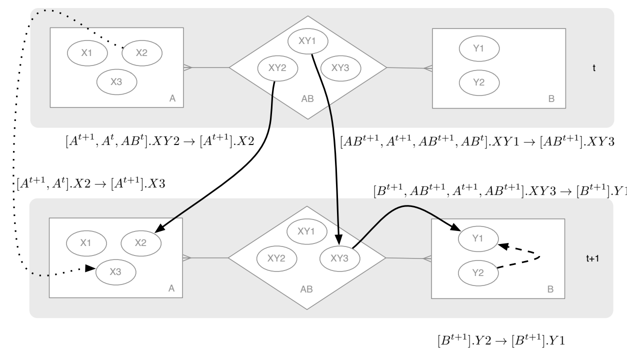


Figure 10: True temporal relational model. The dotted edge was not learned by TRCD, while the dashed edge was left unoriented.

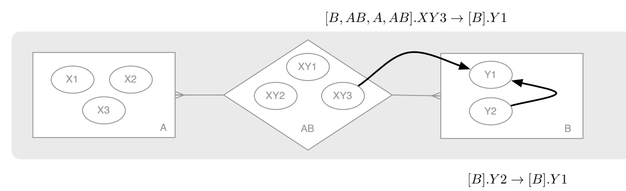


Figure 11: Model learned by RCD for data generated from the temporal model of Figure 10.

9 RELATED WORK

For the propositional case, there are several approaches for learning the structure of temporal probabilistic models. Most of them are not constraint-based methods, but follow the search-and-score paradigm. Friedman *et al.* [1998] present an algorithm to learn the structure of DBNs from complete data using a search-and-score algorithm, and from incomplete data using structural EM. Lähdesmäki and Shmulevich [2008] use Bayesian methods to learn the structure of a DBN from steady state measurements or from time-series data and steady state measurements. Robinson and Hartemink [2010] present an MCMC algorithm to learn the structure of a DBN that changes over time. A different approach that learns a causal temporal model from time series data is the difference-based causality learner [Voortman *et al.*, 2010]. This framework is based on dynamic Structural Equation Models.

A constraint-based method for temporal propositional domains is presented by Entner and Hoyer [2010] who extend the FCI algorithm [Spirtes *et al.*, 2000] to temporal domains. FCI relaxes the causal sufficiency assumption, i.e., it allows the presence of latent common causes. Our approach is the first approach that uses a constraint-based algorithm for data that is both temporal *and* relational (although under the assumption of causal sufficiency).

Another widely used method for inferring causal relationships from temporal data is Granger causality [Granger, 1969]. The main idea underlying Granger causality is that

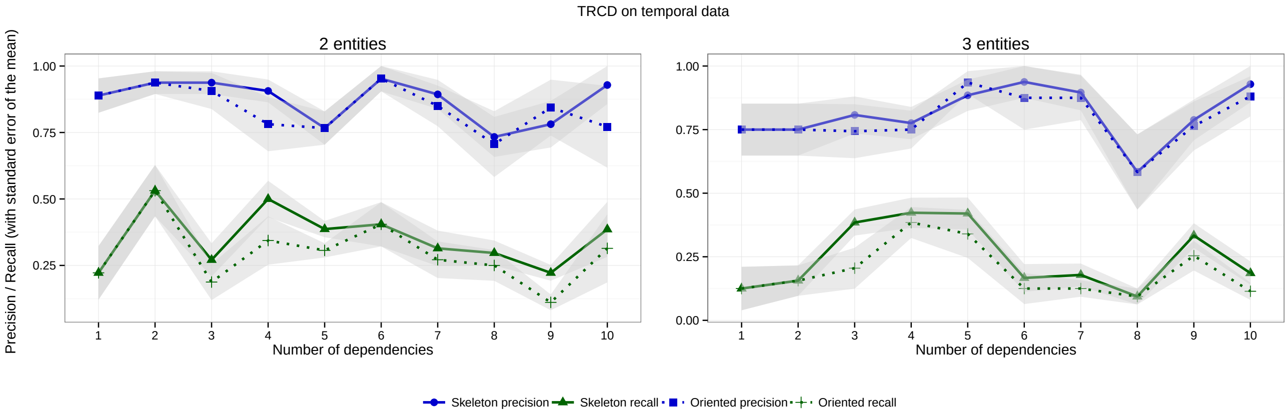


Figure 9: TRCD on synthetic temporal data.

a cause should improve the predictive accuracy of its effect, compared to predictions based solely on the effect's past values. There has been work in extending Granger causality for multivariate settings, as well as in combining Granger causality with graphical models [Eichler, 2012]. Liu *et al.* [2010] propose a regularized Hidden Markov Random Field regression to learn the structure of a temporal causal graph from multivariate time-series data. However, the methods used for learning the structure of the causal model are not constraint-based.

Another line of work in learning temporal and relational models stems from combining first-order logic with probabilistic frameworks. Logical Hidden Markov Models extend Hidden Markov Models to handle relational (non-flat data) [Kersting *et al.*, 2006]. Kersting and Raiko [2005] provide an EM-based algorithm to learn the structure of LHMMs.

In terms of representation, Manfredotti [2009] introduces relational dynamic Bayesian networks (RDBNs), a first-order logic-based extension of dynamic Bayesian networks. RDBNs are similar to the relational model we define; however, we provide an explicit characterization for the space of relational paths, and subsequently, the space of relational dependencies. To be more specific, temporal relational paths in our framework are restricted to conjunctions of predicates that correspond to possible traversals of the relational schema. This restriction, together with the domain specific hop threshold, allows us to enumerate the space of potential dependencies to learn.

10 CONCLUSIONS AND FUTURE WORK

In this paper we presented a formalization of temporal relational models, and we extended the theory of relational d-separation to the temporal domain. We presented a constraint-based algorithm, TRCD, that leverages the notion of temporal relational d-separation to learn the causal structure of temporal relational models from data. We showed that the algorithm is sound and complete, and we provided experimental evidence that showcases the correctness of TRCD. Finally, we showed the improvement that TRCD achieves compared to RCD when applied to domains with a temporal component.

TRCD makes certain simplifying assumptions. Future work could focus on relaxing some of those assumptions, specifically, allowing the structure of the causal model to change over time (change point detection for relational data). Another avenue for future research is relaxing the causal sufficiency assumption by employing techniques such as blocking [Rattigan *et al.*, 2011] for temporal relational domains. Finally, an important issue that arises when modelling time as a discrete quantity is how to choose the appropriate granularity of time points. Ribeiro *et al.* [2012] provide an analysis on how aggregating at a given time granularity affects the characterization of the underlying temporal process. Continuous time Bayesian networks [Nodelman *et al.*, 2002, 2003] provide a way around this by allowing each variable to be modeled at a different time granularity.

Acknowledgements

Funding was provided by the U.S. Army Research Office (ARO) and Defense Advanced Research Projects Agency (DARPA) under Contract Number W911NF-11-C-0088. The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Nathan Eagle and Alex (Sandy) Pentland. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, March 2006.

Michael Eichler. Graphical modelling of multivariate time series. *Probability Theory and Related Fields*, 153(1-2):233–268, 2012.

Doris Entner and Patrik Hoyer. On causal discovery from time series data using FCI. In *Proceedings of the 5th European Workshop on Probabilistic Graphical Models (PGM-2010)*, pages 121–128, 2010.

Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the structure of dynamic probabilistic networks. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 139–147, 1998.

Dan Geiger and Judea Pearl. On the logic of causal models. In *Proceedings of the Fourth Annual Conference on Uncertainty in Artificial Intelligence*, pages 136–147, 1988.

Clive W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–38, July 1969.

Kristian Kersting and Tapani Raiko. "Say EM" for selecting probabilistic models for logical sequences. In *Proceedings of the Twenty-First Conference in Uncertainty in Artificial Intelligence*, pages 300–307, 2005.

Kristian Kersting, Luc De Raedt, and Tapani Raiko. Logical hidden Markov models. *Journal of Artificial Intelligence Research*, 25:425–456, 2006.

Harri Lähdesmäki and Ilya Shmulevich. Learning the structure of dynamic Bayesian networks from time series and steady state measurements. *Machine Learning*, 71(2-3):185–217, June 2008.

Yan Liu, Alexandru Niculescu-Mizil, Aurelie C. Lozano, and Yong Lu. Learning temporal causal graphs for relational time-series analysis. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, pages 687–694, 2010.

Marc Maier, Katerina Marazopoulou, David Arbour, and David Jensen. 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, 2013.

Marc Maier, Katerina Marazopoulou, and David Jensen. Reasoning about Independence in Probabilistic Models of Relational Data. *arXiv:1302.4381*, 2013.

Cristina E. Manfredotti. *Modeling and Inference with Relational Dynamic Bayesian Networks*. PhD thesis, University of Milano, 2009.

Kevin P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

Uri Nodelman, Christian R. Shelton, and Daphne Koller. Continuous time Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 378–387, 2002.

Uri Nodelman, Christian R. Shelton, and Daphne Koller. Learning continuous time Bayesian networks. In *Proceedings of the Nineteenth International Conference on Uncertainty in Artificial Intelligence*, pages 451–458, 2003.

Judea Pearl and Thomas Verma. A theory of inferred causation. In J. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning: Proceeding of the Second International Conference*, pages 441–452. Morgan Kaufmann, 1991.

Matthew J.H. Rattigan, Marc Maier, and David Jensen. Relational blocking for causal discovery. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 145–151, 2011.

Bruno F. Ribeiro, Nicola Perra, and Andrea Baronchelli. Quantifying the effect of temporal resolution in time-varying network. *CoRR*, abs/1211.7052, 2012.

Joshua W. Robinson and Alexander J. Hartemink. Learning non-stationary dynamic Bayesian networks. *Journal of Machine Learning Research*, 11:3647–3680, December 2010.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction and Search*. MIT Press, Cambridge, MA, 2nd edition, 2000.

Thomas Verma and Judea Pearl. Causal networks: Semantics and expressiveness. In *Proceedings of the Fourth Annual Conference on Uncertainty in Artificial Intelligence*, pages 352–359, 1988.

Mark Voortman, Denver Dash, and Marek J. Druzdzel. Learning why things change: The difference-based causality learner. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pages 641–650, 2010.