Non-intuitive conditional independence facts hold in models of network data

Marc Maier  Katerina Marazopoulou  David Jensen
School of Computer Science
University of Massachusetts Amherst
{maier, kmarazo, jensen}@cs.umass.edu

1. INTRODUCTION

Many social scientists and researchers across a wide range of fields focus on analyzing a single causal dependency or a conditional model of some outcome variable. However, to reason about interventions or conditional independence, it is useful to construct a joint model of a domain. Researchers in computer science, statistics, and philosophy have developed representations (e.g., Bayesian networks), theory, and learning algorithms for causal discovery of joint models from observational data. Bayesian networks are graphical models that encode joint probability distributions over a system of variables, and they can be interpreted causally under a few assumptions [7, 8].

The rules of d-separation—a set of graphical criteria—are the foundation for algorithmic derivation of the conditional independence facts implied by the structure of Bayesian networks [1]. This theory connects causal structure with conditional independence and can be leveraged to learn causal models from observational data. Accurate reasoning about conditional independence facts is the basis for constraint-based algorithms that learn the structure of Bayesian networks (e.g., PC [8]).

Bayesian networks, as well as most analytic methods for causal analysis (e.g., linear regression), assume that data instances are independent and identically distributed (IID). However, many real-world systems involve multiple types of interacting entities with probabilistic dependencies among the variables on those entities. For example, citation data involve researchers collaborating on scholarly papers that cite prior work. Over the past 15 years, researchers in statistics and computer science have devised more expressive classes of directed graphical models, such as probabilistic relational models (PRMs), which remove the assumptions of IID data to model network, or relational, data [2]. Relational models more closely represent the real-world domains that many social scientists and other researchers investigate. To successfully learn causal models from observational, network data, we need a similar theory for deriving conditional independence from relational models.

In this paper, we explain why d-separation does not correctly produce conditional independence facts when applied to the structure of relational models. We show that non-intuitive conditional independence facts hold in models of network data through relationally d-connecting paths that only manifest in ground graphs. We describe the abstract ground graph, a lifted representation recently introduced by Maier et al. [4], that enables a sound, complete, and computationally efficient method for deriving conditional independence facts from relational models. This approach, which solves the problem of relational d-separation, forms the basis for RCD—a recent sound and complete constraint-based algorithm that learns causal models from network data [3].

2. BACKGROUND ON D-SEPARATION

A common assumption in statistics, machine learning, and causal discovery is that data instances are independent and identically distributed. IID data (also referred to as propositional data) are effectively represented as a single table, where rows correspond to independent instances and columns are attributes of those instances. Bayesian networks are widely used probabilistic graphical models of propositional data [6]. The model consists of two parts: (1) a structural component—a directed acyclic graph $G = (V, E)$, where $V$ is a set of vertices corresponding to random variables and $E \subseteq V \times V$ is a set of directed edges that encode the conditional independencies among the variables; and (2) the parameters—a conditional probability distribution $P(V | \text{parents}(V))$ for each variable $V \in V$, where $\text{parents}(V) \subseteq V \setminus \{V\}$ is the set of parent variables for $V$.

The joint probability distribution can be factored using the conditional distributions: $P(V) = \prod_{V \in V} P(V | \text{parents}(V))$.

The graphical rules of d-separation entail the same set of conditional independencies that are implied by the Markov condition on $G$ (i.e., every variable $V \in V$ is conditionally independent of its non-descendants given its parents) [5]. In the following definition, a path in $G$ is a sequence of vertices

![Figure 1: Graphical patterns of d-separating and d-connecting path elements among disjoint sets of variables X and Y given Z. Paths for which there exists a non-collider in Z or a collider not in Z are d-separating. Paths for which all non-colliders are not in Z and all colliders (or a descendant of colliders) are in Z are d-connecting.](image-url)
following edges in either direction. A variable \( V \) is a collider on a path \( p \) if the two arrowheads point at each other (col-lide) at \( V \); otherwise, \( V \) is a non-collider on \( p \).

**Definition 1 (d-separation)** Let \( X, Y, \) and \( Z \) be disjoint sets of variables in directed acyclic graph \( G \). A path from some \( X \in X \) to some \( Y \in Y \) is \( d \)-connected given \( Z \) if and only if every collider \( W \) on the path, or a descendant of \( W \), is a member of \( Z \), and there are no non-colliders in \( Z \). Then, say that \( X \) and \( Y \) are \( d \)-separated by \( Z \) if and only if there are no \( d \)-connecting paths between \( X \) and \( Y \) given \( Z \).

Figure 1 depicts the graphical patterns found along paths that lead to \( d \)-separation or \( d \)-connection based on Definition 1. In practice, \( d \)-separation queries can be answered with a linear-time algorithm based on breadth-first search and reachability on \( G \) \[^1\].

### 3. RELATIONALLY D-CONNECTING PATHS

Consider a corporate analyst who was hired to identify which products and employees are effective and productive for some organization. If the company is structured as a pure project-based organization, the analyst may collect data as described by the relational schema in Figure 2(a) (without the dependencies). The schema denotes that employees can collaborate and work on multiple products, each of which is funded by a specific business unit. The analyst has also obtained variables on each entity—competence of employees, the success of each product, and the revenue of business units. In this example, the organization consists of two employees, two products, and a single business unit, shown in the relational skeleton \(^2\) (in gray) in Figure 2(b).

The analyst may believe that the organization operates under the model depicted in Figure 2(a). The competence of an employee affects the success of products they develop, and the revenue of a business unit is influenced by the success of products that it funds. The analyst then needs to verify the model structure in order to accurately advise executive decisions, such as determining which business units should have increased funding. Perhaps the analyst has experience in graphical models and decides to check if the conditional independencies encoded by the model are reflected in the data, assuming the faithfulness condition (i.e., all conditional independencies that hold in the data are implied by \( d \)-separation). The analyst then naively applies \( d \)-separation to the model structure in an attempt to derive these conditional independencies.

Applying \( d \)-separation to the model in Figure 2(a) suggests that employee competence is conditionally independent of business unit revenue given product success. To see why this approach is flawed and why this particular conditional independence statement is not generally true, we must consider the ground graph. A necessary precondition for inference is to apply a model to a data instantiation, yielding a ground graph to which \( d \)-separation can be applied. For a Bayesian network, a ground graph consists of replicates of the model structure for each data instance. In contrast, a relational model defines a template that results in ground graphs with varying structure depending on the data.

Figure 2(b) shows the ground graph for the relational model in Figure 2(a) applied to the relational skeleton corresponding to checking on the class dependency graph, which fails to reflect the data.

A relational skeleton is oftentimes referred to as a data graph since it can be represented as a network.

---

\[^1\] A relational skeleton is oftentimes referred to as a data graph since it can be represented as a network.

\[^2\] The indirect effect attributed to a relationally \( d \)-connecting path is often referred to as interference, a spillover effect, or a violation of SUTVA because the treatment of one instance affects the outcome of another.
to capture these paths. In the next section, we describe the abstract ground graph, a lifted representation that captures every potential relationally $d$-connecting path.

4. SOLUTION

The theory of $d$-separation is only a useful theory because the derived conditional independence facts hold for all distributions represented by a Bayesian network. Thus, the implications of relational $d$-separation should analogously hold for all distributions of variables for the space of all possible ground graphs. However, in contrast to Bayesian networks, relational models produce ground graphs that vary with the relational structure of the underlying data (e.g., different products are developed by varying numbers of employees). The example in Section 3 also illustrates how relationally $d$-connecting paths only manifest in ground graphs, not the model representation. These problems indicate that an alternative representation is required.

The abstract ground graph, introduced by Maier et al. [4], is a lifted representation that supports relational $d$-separation. A relational model has a corresponding set of abstract ground graphs, one for each perspective (i.e., entity or relationship class in its underlying schema). As their name suggests, abstract ground graphs abstract all paths of dependence in any ground graph of a relational model. Note that abstract ground graphs generalize Bayesian networks: For a propositional model, they are structurally equivalent. Abstract ground graphs are also the underlying representation in a recent constraint-based causal discovery algorithm for network data [3].

Abstract ground graphs are provably sound and complete in this abstraction, supported by two main innovations: (1) The dependencies in the model are translated across perspectives and among arbitrary pairs of relational variables; and (2) they explicitly represent the potential intersection between pairs of relational variables. These facets effectively enable abstract ground graphs to represent any potential relationally $d$-connecting path in any ground graph.

Figure 3 shows a fragment of an abstract ground graph from the employee perspective for the model in Figure 2(a). The nodes are depicted with their intuitive meaning rather than their actual syntax to support understanding. For example, when choosing Roger as a base instance, the node labeled “Employee’s competence” includes Roger. Competence, the node “Success of products developed by an employee” includes Laptop.Success, and the node with the intersection includes Tablet.Success. The example relationally $d$-connecting path from the previous section is encoded by this abstract ground graph.

5. DISCUSSION

The results of this paper imply potential flaws in the design and analysis of some real-world studies. If researchers of social or economic systems choose inappropriate data and model representations, then their analyses may omit important classes of dependencies. Specifically, choosing a propositional representation from an inherently relational domain may lead to serious errors. An abstract ground graph from a given perspective defines the exact set of variables that must be included in any propositionalization. The absence of any relational variable (including intersection variables) may unnecessarily introduce latent common causes, which could result in the inference of a causal dependency where conditional independence was not detected. The potential presence of relationally $d$-connecting paths suggests that researchers should carefully consider how to represent their domains in order to accurately reason about conditional independence.

6. REFERENCES