A Kernel Conditional Independence Test for Relational Data

Sanghack Lee and **Vasant Honavar**
Artificial Intelligence Research Laboratory
College of Information Sciences and Technology
The Pennsylvania State University, University Park, PA 16802
{sxl439, vhonavar}@ist.psu.edu

Abstract

Conditional independence (CI) tests play a central role in statistical inference, machine learning, and causal discovery. Most existing CI tests assume that the samples are independently and identically distributed (i.i.d.). However, this assumption often does not hold in the case of *relational data*. We define *Relational Conditional Independence* (RCI), a generalization of CI to the relational setting. We show how, under a set of structural assumptions, we can test for RCI by reducing the task of testing for RCI on non-i.i.d. data to the problem of testing for CI on several data sets each of which consists of i.i.d. samples. We develop *Kernel Relational CI test* (KRCIT), a nonparametric test as a practical approach to testing for RCI by relaxing the structural assumptions used in our analysis of RCI. We describe results of experiments with synthetic relational data that show the benefits of KRCIT relative to traditional CI tests that don't account for the non-i.i.d. nature of relational data.

1 INTRODUCTION

Observational and experimental data represent systematic interactions among a set of random variables of interest. Conditional independence (CI) tests constitute essential tools for understanding such interactions. Random variables X and Y are said to be conditionally independent given Z, denoted by $X \perp\!\!\!\perp Y \mid Z$, if and only if the joint distribution P_{xyz} can be factorized as $\mathsf{P}_{x|z}\mathsf{P}_{y|z}\mathsf{P}_{z}$. The notion of CI plays a central role in statistical inference (Dawid, 1979), probabilistic graphical models (Koller and Friedman, 2009), and causal discovery (Pearl, 2000; Spirtes et al., 2000). A variety of methods including, in particular, non-parametric methods (Fukumizu et al., 2008; Zhang et al., 2011; Doran et al., 2014; Lee and Honavar, 2017) have been developed to test for CI in settings where the parametric form of the underlying distribution is unknown but a measure of closeness between data samples can be defined, e.g., using a kernel function. However, these methods implicitly or explicitly assume that the data samples are independently and identically distributed (i.i.d.).

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

Against this background, inspired by the notion of relational d-separation (Maier et al., 2013), which generalizes a graphical criterion for CI to a specific model of relational data, we (i) Formalize Relational Conditional Independence (RCI), the relational counterpart of CI. (ii) Examine the dependence and heterogeneity of relational variables in terms of the underlying relational structure. (iii) Based on the preceding analyses, devise a Kernel Relational CI Test (KRCIT) that, to the best of our knowledge, offers the first practical method for testing for RCI.

(iv) Describe results of experiments with synthetic relational data that show the benefits of KRCIT relative to traditional CI tests that don't account for the non-i.i.d. nature of relational data. RCI and KRCIT offer new ways to understand dependencies in relational data across a broad range of practical applications.

2 PRELIMINARIES

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

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

We use a simplified version of Entity-Relationship (ER) model (Chen, 1976) to describe relational data (Friedman et al., 1999; Heckerman et al., 2007; Maier et al., 2013). A relational schema $\mathcal{S} = \langle \mathbf{E}, \mathbf{R}, \mathbf{A} \rangle$ describes the relational domain of interest with a set of entity classes $\mathbf{E}$ (e.g., *person*, *student*), relationship classes $\mathbf{R}$ (e.g., *friend-of*, *son-of*), and attribute classes $\mathbf{A}$ (e.g., *gender*, *income*). We refer to the union $\mathbf{E} \cup \mathbf{R}$ as the set of item classes $\mathbf{I}$. In general, a relationship class $R \in \mathbf{R}$ can be n-ary where $n \geq 2$ (e.g., *contract* is a set of ternary relationships involving products, buyers, and sellers). We use $\mathbf{A}(I)$ to denote an item class I's attribute classes, and $\mathbf{A}^{-1}(X)$ by to denote the item class of X.

A relational skeleton $\sigma \in \Sigma_{\mathcal{S}}$ is an instantiation of a given relational schema, which can be viewed as an undirected bipartite graph. We denote by $\sigma(I)$ a set of items of an item class $I \in \mathbf{I}$. Given an item $i \in \sigma(I)$, we use $i.X$ to denote the item i's attribute class X. Note that $i.X$ is a random variable which takes a value $i.x \in \mathcal{X}$. An edge $(i, r) \in \sigma$ represents the participation of an entity i in a relationship r. For simplicity, we represent a skeleton, whose relationship classes are binary, as an undirected graph of entities. We use "relational structure" to mean the the graphical structure of a relational skeleton, and "relational data" to mean a set of data that conform to a given relational schema. We use letters such as i and j to stand for integers or items (e.g., the letters i and j to refer two items in $\sigma(I)$).

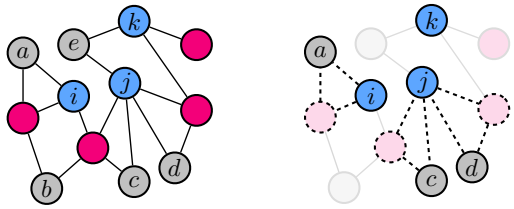

Figure 1: (**Left**) a small relational skeleton σ as an undirected graph of entities of three classes, Blue, Magenta, and Gray; and (**Right**) an example of a relational variable V where $V(\ell)$ refers to the multiset of attribute X of a set of Gray items forming a triangle with the given Blue item ℓ and a Magenta item. Hence, $V(i) = \{a.X\}$; $V(j) = \{c.X, d.X\}$; and $V(k) = \emptyset$;

3 CI TEST WITH RELATIONAL DATA

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

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

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

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

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

As a simple but concrete example, a relational variable V, 'smoking status of one's neighbors', is defined with

I, J both being the 'Person' item class, X corresponding to the attribute class 'smoking', where $j \in \sigma(J)$ where $j.X \in V(i)$ is i's neighbor for every person $i \in \sigma(I)$. Given a person Arthur who has two neighbors Sally and Sean, $V(\text{arthur}) = \{\text{sally}.Smoke\, \text{sean}.Smoke\}$ results a 2-dimensional random variable. In Fig. 1, we illustrate another example of relational variables involving three entity classes.

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

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

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

It is easy to see that this definition of relational CI (RCI) generalizes traditional CI where i.i.d. samples are drawn from P_{xyz}. Let U be "smoking status of oneself" and W be "smoking status of one's parents", e.g., $U(\text{arthur}) = \{\text{arthur}.Smoke\}$. We might ask $(U \perp\!\!\!\perp V \mid W)_\sigma$: "Is one's smoking status independent of one's neighbors' smoking status given one's parents' smoking status?" While, in the real world, answering such question can be quite difficult because of the complexity and partial observability of interactions among people, the notion of relational CI, once operationalized, can help extract useful insights from relational data.

3.1 RELATIONAL VARIABLES AND PARAMETER TYING

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

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

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

Proposition 1 (Identically Distributed Random Variables)**.** *Let $\mathcal{G}$ be a directed acyclic graph representing a conditional independence structure of item attributes where each item attribute, e.g., $k.X$, is labeled with its attribute class, e.g., X. Given that the parameters are tied, random variables $i.X$ and $j.X$ in $\mathcal{G}$ are identically distributed if $\mathcal{G}\left[an^{\mathcal{G}}(i.X)\right] \cong_{i.X,j.X} \mathcal{G}\left[an^{\mathcal{G}}(j.X)\right]$.*

Proposition 2 (Independent Random Variables)**.** *Two random variables $i.X$ and $j.X$ are independent if $\mathcal{G}\left[an^{\mathcal{G}}(i.X)\right] \cap \mathcal{G}\left[an^{\mathcal{G}}(j.X)\right] = \emptyset$.*

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

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

Assumption 3. *Let $\mathcal{S} = \langle \mathbf{E}, \mathbf{R}, \mathbf{A} \rangle$ be a relational schema and σ be a relational skeleton of $\mathcal{S}$. Let i and j be items in σ and $X \in \mathbf{A}(I)$ of $I \in \mathbf{I}$. Then, random variables $i.X$ and $j.X$ are independent and identically distributed if*

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

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

3.2 HANDLING NON-I.I.D. VARIABLES

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

Our solution to this problem is to perform CI tests by decomposing, with respect to a given RCI query, the set of items $\sigma(I)$ into subsets of items such that each subset yields a set of i.i.d. observations under the above assumption. Consider a function $id : \sigma(I) \mapsto \mathbb{Z}$ such that for $id(i) = id(j)$ only if $(u_i, v_i, w_i) \overset{d}{=} (u_j, v_j, w_j)$ and $(u_i, v_i, w_i) \perp\!\!\!\perp (u_j, v_j, w_j)$ (i.e., $\mathrm{CC}_i^\sigma \cong_{i,j} \mathrm{CC}_j^\sigma$ and $\mathrm{CC}_i^\sigma \neq \mathrm{CC}_j^\sigma$ given that U, V, and W are *isomorphism-invariant*, see Appendix for definition). Then, a traditional CI test, treating U, V, and W as random variables,

$$U \perp\!\!\!\perp V \mid W, id$$

will remove bias introduced by the relational structure provided we have large enough samples per condition (i.e., a large number of CCs per isomorphic class). Such a naive solution, however, has severe limitations in practice: i) It is possible, in the worst case, that all items in relational data are connected and ii) Each connected component might be non-isomorphic to others.

4 A KERNEL RELATIONAL CI TEST

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

4.1 KERNEL FOR RELATIONAL VARIABLES

We provide kernels for relational variables that reflect our understanding of relational structure as in Section 3.1. Consider a relational variable U, associated with the attribute class X. A kernel function for U, k_U, measures *similarity* between two instantiations of U where each instantiation, e.g., $u_i \in \mathcal{X}^{|U_i|}$, consists of a set of item attributes. We illustrate our approach using the R-convolution kernel (Haussler, 1999), which computes the kernel over two sets as the sum of kernel values for every pair of elements from two sets. Thus, we define k_U as

$$k_U\left(u_i, u_j; k_{\mathrm{IA}}^X\right) = \sum_{a.X \in U_i} \sum_{b.X \in U_j} k_{\mathrm{IA}}^X\left(a.X, b.X\right) \tag{1}$$

where the base kernel $k_{\mathrm{IA}}^X\left(a.X, b.X\right)$ measures the similarity between two item attributes. Based on the analysis in Section 3.1, U_i and U_j do not necessarily yield identically-distributed item attributes. Hence, we design $k_{\mathrm{IA}}^X\left(a.X, b.X\right)$ by taking both homogeneity and attribute values into consideration: k_{IA}^X is defined as a product kernel of the kernel for homogeneity k_σ and a kernel for attribute values k_x:

$$k_{\mathrm{IA}}^X\left(a.X, b.X\right) = k_\sigma\left(a, b\right) k_x\left(a.x, b.x\right)$$

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

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

We use h-hop neighbors of an item in σ to induce a context of the item for practicality since a connected component, e.g., CC_a^σ, can be as large as the given relational skeleton σ. Thus, we design the kernel for homogeneity $k_\sigma\left(a, b\right)$ as a graph kernel between two labeled graphs $\sigma\left[ne_h^\sigma\left(a\right)\right]$ and $\sigma\left[ne_h^\sigma\left(b\right)\right]$ where $ne_h^\sigma\left(a\right)$ is a set of items in σ that are reachable in no more than h hops from item a in σ. Each item in a context, e.g., $\sigma\left[ne_h^\sigma\left(a\right)\right]$, is labeled with its item class except the item a, which is assigned to a special label allowing a graph kernel between $\sigma\left[ne_h^\sigma\left(a\right)\right]$ and $\sigma\left[ne_h^\sigma\left(b\right)\right]$ to match a and b. This reflects '$\approx_{i,j}$', graph isomorphism with an additional constraint (Assumption 3).

We choose to exploit an existing graph kernel for labeled graphs. For example a shortest-path kernel (Borgwardt and Kriegel, 2005) is given by

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

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

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

$$k_G(i,j) = k_U(u_i,u_j;k_\sigma)\, k_V(v_i,v_j;k_\sigma)\, k_W(w_i,w_j;k_\sigma) \tag{2}$$

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

4.2 TREATING DEPENDENT OBSERVATIONS

Now we briefly discuss how we can handle dependent observations. In relational data, the dependencies among observations can arise for different reasons. In previous section, we showed that two item attributes become dependent if they share the same ancestors (Proposition 2). However, in some settings, we can ignore some types of dependence among observations. For instance, consider a hidden Markov model with hidden variables X and observed variables Y where $X_{t-1} \perp\!\!\!\perp Y_t \mid X_t$ and $\mathsf{P}(Y_t \mid X_t)$ and $\mathsf{P}(X_t \mid X_{t-1})$ are time-invariant. Simply running a traditional CI test on a sample $\{(x_{t-1}, y_t, x_t)\}_{t=2}^n$ would likely result in the null hypothesis not being rejected in spite of correlations among observations, e.g., (X_{t-1}, Y_t, X_t) and $(X_{t-2}, Y_{t-1}, X_{t-1})$. Variables like X_t and X_{t-1} are naturally represented as (temporally) related variables, the dependencies among which can be broken by conditioning on an appropriate set of variables. In this regard, we treat 'dependent observations' to be resolved explicitly through conditionals instead of being implicitly removed, e.g., non-i.i.d. CI test for *autocorrelated* data (Flaxman et al., 2016).

Algorithm 1 KRCIT

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

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

4.3 VALIDITY OF FLATTENING APPROACH

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

Condition 4. $U_i, V_i \perp\!\!\!\perp U_j, V_j \mid W_i$ and $(w_i, g_i = w_j, g_j) \rightarrow (U_i, V_i) \stackrel{d}{=} (U_j, V_j)$ for every $i \neq j \in \sigma(I)$.

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

Condition 5. $V_i \perp\!\!\!\perp V_j \mid W_i$ and $(w_i, g_i = w_j, g_j) \rightarrow V_i \stackrel{d}{=} V_j$ for every $i \neq j \in \sigma(I)$.

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

4.4 ALGORITHM

By supplying customized kernels for U, V, W, and G (Equation 1 and 2), a kernel CI test in an i.i.d. setting will decide whether to accept or reject the null hypothesis given a flattened sample Ω. We illustrate the pseudocode of the kernel relational conditional independence test, KRCIT, in Algorithm 1. If the given query is unconditional such that W is undefined, then $\mathbf{K}_w = 1$.

We considered the following two kernel-based conditional independence tests as a base CI test for KRCIT:

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

5 EMPIRICAL EVALUATION

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

5.1 METHODS

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

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

We list methods to be compared in our experiments. **(Naive)**: We use Hilbert-Schmidt Independence Criterion (HSIC, Gretton et al., 2005) for unconditional cases which uses the eigenspectrum of covariance operators in RKHS. Otherwise, either KCIT or SDCIT is used. **(Residualized)**: We residualize values of a given relational skeleton based on contexts (i.e., replace values to its residuals). Then, aforementioned naive tests are used. We append a prefix 'R-' to denote 'residualized'. **(Residual Kernel)**: Residuals are computed in RKHSs and the kernel for residuals is obtained as described above. Then, naive tests are used where computing kernel for two values (e.g., an RBF kernel) is replaced by looking up the residual kernel matrix. We append a prefix 'RK-'. **(KRCIT)**: We use a postfix '-K' or '-SD' to denote KRCIT with KCIT or with SDCIT, respectively.

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

5.2 A SIMPLE EXAMPLE FOR VISUALIZATION

We start with a simple example that explains how and why KRCIT is better for relational CI testing than standard CI tests that assume i.i.d. data. For the sake of readability, we omit unnecessary details (detailed descriptions are provided in Appendix). We considered a relational schema where attribute class X and Y associates with entity class 'circle' and 'square', respectively. There are other item classes and attribute classes in the schema. A relational skeleton σ consists of CCs of four differ-

[1] Codes are available online at `https://github.com/sanghack81/KRCIT` and `https://github.com/sanghack81/SDCIT`.

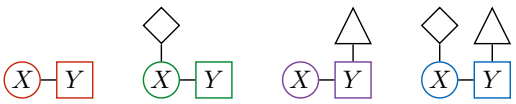

Figure 2: Four connected components composing a relational skeleton.

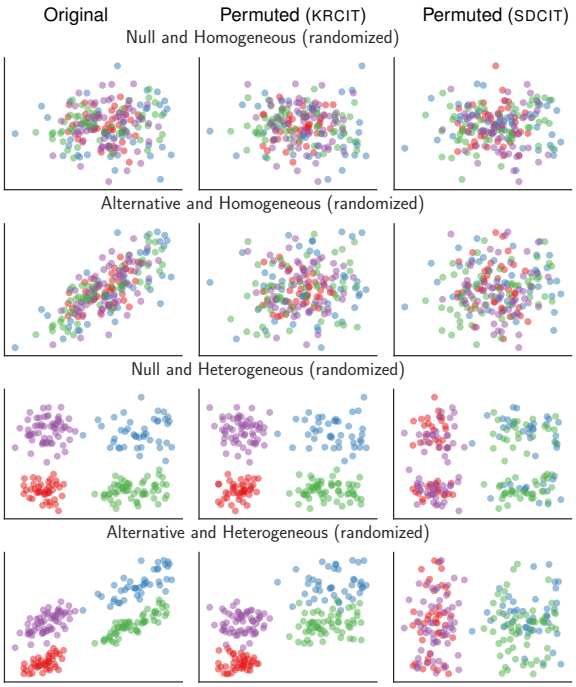

Figure 3: Comparisons of the given data (**left**) and two samples under the null hypothesis using KRCIT with SDCIT (**center**) and by SDCIT (**right**), respectively, with randomized relationships

ent structures as shown in Fig. 2. We designed a generative model such that X is a function of the value in an adjacent 'rhombus' item and Y is a function of X in an adjacent circle and the value in an adjacent 'triangle' item. That is, we make sure that $i.X$ and $j.X$ are not identically-distributed if i is adjacent to a rhombus item while j is not. The same idea also applies to Ys.

We controlled *randomness* of relationships between circles and squares: non-random relationship represents that a resulting relational skeleton consists only circles adjacent to a rhombus are connected to squares adjacent to a triangle (1st and 4th components in Fig. 2) while randomized relationship exhibits a relational skeleton where all four components are balanced. We also controlled *heterogeneity*: the extent to which distributions of (X, Y) of different components diverges from each other.

In the **left** column of Fig. 3, we visualize four relational data based on the combinations of underlying hypothe-

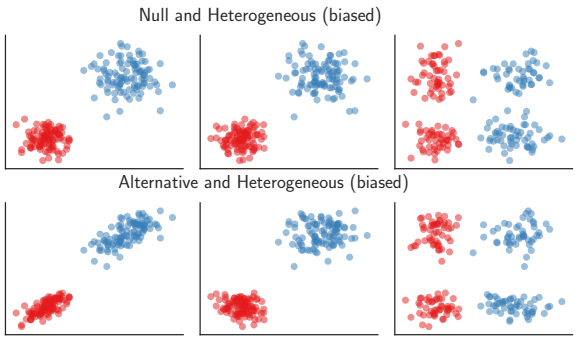

Figure 4: Comparisons with fully biased relationship

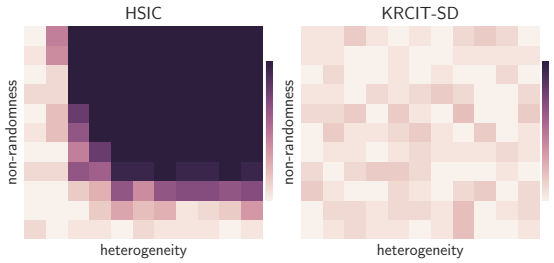

Figure 5: Type-I errors (with 20 trials) varying both non-randomness and heterogeneity

sis and heterogeneity where color codes correspond to the types of structure which associates with the value (x, y) in Fig. 2. Utilizing *permuted* samples, we visualize how KRCIT (with SDCIT) and SDCIT, which is a permutation-based test, produce data consistent with the null hypothesis. KRCIT permutes Ys conditioning on G, which corresponds to 'color', while SDCIT simply shuffles Ys. The **center** and **right** columns correspond to the permuted sample under KRCIT and a naive test, respectively. When contexts more strongly correlate with values, we can more clearly observe the difference between the permuted samples by KRCIT and by SDCIT.

For each row, if the **center** plot is significantly different from its corresponding **left** plot, KRCIT would reject the null hypothesis. For example, KRCIT correctly rejects the null hypothesis for the samples from the alternative hypothesis (row **2nd** and **4th**). Interestingly, SDCIT also correctly rejects samples from the alternative hypothesis. In Fig. 4, we plot null samples when relational skeletons exhibit biased relationships. We can similarly observe difference between two tests. Fig. 5 illustrates type-I errors (given $\alpha = 0.05$) of a naive test (HSIC) and KRCIT-SD based on relational skeletons generated with various degrees of heterogeneity and non-randomness of relationship. KRCIT-SD is robust to heterogeneity and non-random relationships while HSIC is not. Note that residualization approaches utilizing contexts perform similar

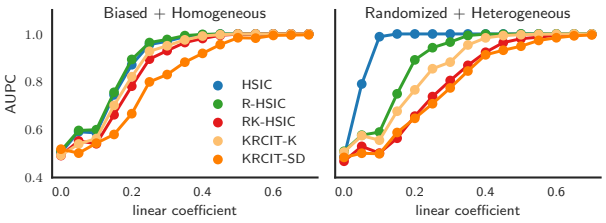

Figure 6: AUPCs with varying dependency where relational skeletons are generated with homogeneous and randomized relationships (**left**) and with heterogeneous and biased relationships (**right**).

to KRCIT (see Appendix).

Finally, we illustrate the changes of power of different tests as the strength of dependency between X and Y is increased (Fig. 6). Tests that use contexts *consistently* estimates dependency without regard to the underlying conditions while HSIC over-rejects samples from weak dependence in certain conditions where rejection comes partially from other than linear dependence (**4th** row in Fig. 3). In summary, whenever contexts provide sufficient information to infer (non-)identically distributed observations, KRCIT is able to eliminate suspicious dependencies due to such heterogeneity.

5.3 MORE COMPLICATED STRUCTURES

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

For tests for the null hypothesis, we obtained similar results as in the previous section. However, the kernel-based residualization approach (RK-HSIC) shows higher type-I errors than expected when larger contexts are employed (see Appendix). KRCIT performed as desired even with larger contexts. For tests for alternative hypothesis, we report AUPC in Fig. 7. With properly-sized contexts (hop=1), both residualization-based methods perform well. However, they are sensitive to the choice of contexts – the power of both R-HSIC and RK-HSIC drops. Both these KRCIT methods are relatively weaker than any other tests when dependence between relational variables is not strong enough. However, KRCIT with SDCIT seems

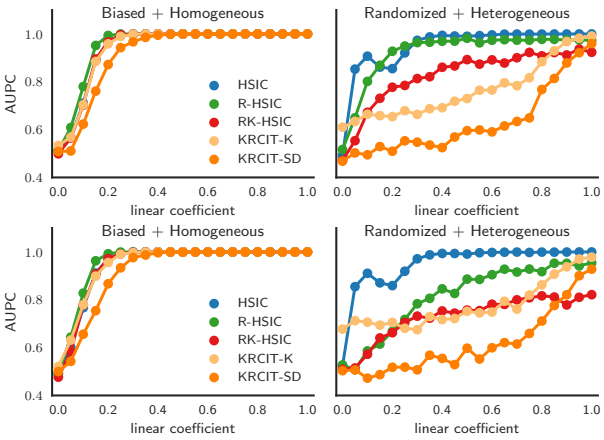

Figure 7: AUPCs with hop=1 (**top**) and hop=2 (**bottom**) under two different settings for relational skeletons

robust to the choice of contexts and achieves high AUPC as the dependence increases.

5.4 CONDITIONAL TESTS

We investigated whether KRCIT would be able to discover the causal structure of synthetic relational data generated from an RCM. Thus, we focus on testing 1) relational version of Markov condition, which is essential to learn the undirected causal structure, and 2) conditional dependence, which is critical to infer the orientation of undirected causal relationships. We constructed a set of relational skeletons of 3 entity classes and 3 relationship classes between every pair of entity classes. We controlled for the maximum number of neighbors of the same item class of an item (e.g., a circle item can have at most three square neighbors) and the number of entities per entity class. We generate relational skeletons to exhibit correlation among different relationships (see Appendix for details). Then, values are generated based on two hypotheses. For the null hypothesis, we generate values, roughly, in a causal order $X \to Z \to Y$ or $X \leftarrow Z \to Y$. For the alternative hypothesis, a relational data is generated based on a causal order $X \to Z \leftarrow Y$. We test $(U \perp\!\!\!\perp V \mid W)_\sigma$ where U, V, and W associates with X, Y, and Z, respectively.

In Fig. 8, we plot AUPCs and type-I error rates under different sizes of flattened sample, maximum cardinalities of relationships, and the sizes of contexts in terms of hops. The **left** plots demonstrate that the power of all tests increases as larger sample is used (with max 3 relationships and hop 1). However, KRCIT with KCIT as a base CI test suffers high type-I error rates. The **center** plots depict the negative effect of more complex structure on both power and type-I errors. This implies the general

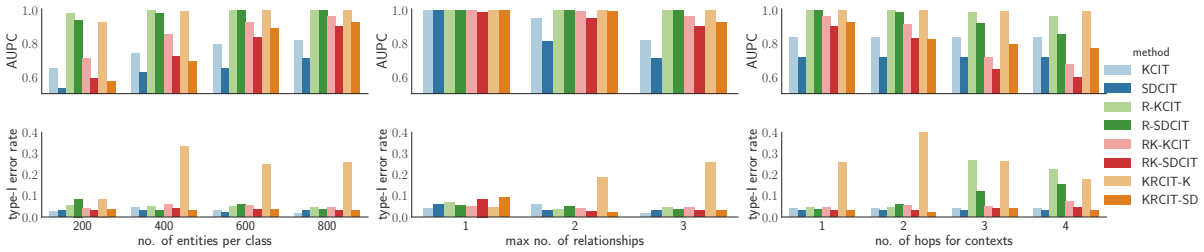

Figure 8: AUPCs and type-I error rates of various tests with relational skeletons generated with different settings and tests employed different size of subgraphs as contexts.

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

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

6 DISCUSSION

Maier et al. (2013) considered relational d-separation, a problem closely related to RCI. However, they relied on a traditional CI test by simply flattening and aggregating relational data (i.e., average) without incorporating structural information. As we have shown, such an approach biases the results of independence tests (Section 5.2). An independence test on two non-i.i.d. data sets is addressed by Zhang et al. (2009), who considered generalizing HSIC to the setting of structured and interdependent observations. However, their work focused on explicitly known CI structures which can be represented as undirected graphical models utilizing factorization provided by the exponential family.

In this paper, we have defined CI in the relational setting and provided an effective approach to testing for RCI. Our definition makes use of a definition of a relational variable that subsumes the notions of *slot chains* in probabilistic relational models (Friedman et al., 1999), *relational path* in relational causal models with *bridge burning semantics* (Maier et al., 2013) and *path semantics* (Lee and Honavar, 2016), and *first order expression* in DAPER (Heckerman et al., 2007).

7 SUMMARY AND FUTURE WORK

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

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

Acknowledgements

The authors are grateful to UAI 2017 anonymous reviewers for their thorough reviews. This research was supported by the Edward Frymoyer Endowed Professorship, the Center for Big Data Analytics and Discovery Informatics at the Pennsylvania State University, and the Sudha Murty Distinguished Visiting Chair in Neurocomputing and Data Science at the Indian Institute of Science.

References

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

Chen, P. P.-S. (1976). The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36.

Chwialkowski, K., Sejdinovic, D., and Gretton, A. (2014). A Wild Bootstrap for Degenerate Kernel Tests. In *Advances in Neural Information Processing Systems 27*, pages 3608–3616.

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

Doran, G., Muandet, K., Zhang, K., and Schölkopf, B. (2014). A Permutation-Based Kernel Conditional Independence Test. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 132–141.

Flaxman, S. R., Neill, D. B., and Smola, A. J. (2016). Gaussian Processes for Independence Tests with Non-iid Data in Causal Inference. *ACM Transactions on Intelligent Systems and Technology*, 7(2):1–23.

Friedman, N., Getoor, L., Koller, D., and Pfeffer, A. (1999). Learning Probabilistic Relational Models. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1300–1309.

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

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

Haussler, D. (1999). Convolution Kernels on Discrete Structures. Technical report, University of California, Santa Cruz.

Heckerman, D., Meek, C., and Koller, D. (2007). Probabilistic Entity-Relationship Models, PRMs, and Plate Models. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning*, pages 201–238.

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

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

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

Lee, S. and Honavar, V. (2016). A Characterization of Markov Equivalence Classes of Relational Causal Models under Path Semantics. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*, pages 387–396.

Lee, S. and Honavar, V. (2017). Self-Discrepancy Conditional Independence Test. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*. (to appear).

Maier, M., Marazopoulou, K., Arbour, D., and Jensen, D. (2013). A Sound and Complete Algorithm for Learning Causal Models from Relational Data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 371–380.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition.

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

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT Press, 2nd edition.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. (2011). Kernel-based Conditional Independence Test and Application in Causal Discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 804–813.

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