Propensity Score Matching for Causal Inference with Relational Data

David Arbour **Katerina Marazopoulou** **Dan Garant** **David Jensen**
School of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
{darbour, kmarazo, dgarant, jensen}@cs.umass.edu

Abstract

Propensity score matching (PSM) is a widely used method for performing causal inference with observational data. PSM requires fully specifying the set of confounding variables of treatment and outcome. In the case of relational data, this set may include non-intuitive relational variables, i.e., variables derived from the relational structure of the data. In this work, we provide an automated method to derive these relational variables based on the relational structure and a set of naive confounders. This automatic construction includes two unusual classes of variables: relational degree and entity identifiers. We provide experimental evidence that demonstrates the utility of these variables in accounting for certain latent confounders. Finally, through a set of synthetic experiments, we show that our method improves the performance of PSM for causal inference with relational data.

1 INTRODUCTION

Propensity score matching (PSM) [Rosenbaum and Rubin, 1983] is a widely used tool for determining causal effects from observational data. Propensity scores summarize the effects of a potentially large number of confounding variables by creating a *predictive* model of treatment. The computation of a propensity score requires specifying a set of potentially confounding variables. This task is relatively straightforward for propositional (i.i.d.) data. However, many causal analyses consider data in which treatment, outcome, and potential confounders can arise from the interactions among multiple types of interrelated entities. Propensity score matching becomes substantially more challenging in such relational data.

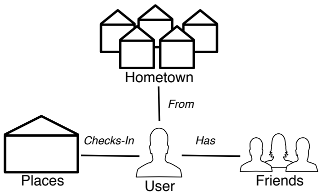

Figure 1: Example of relational data: users are friends with other users, each user comes from a hometown, and users check-in at places.

To illustrate this, consider the example domain shown in Figure 1, depicting a plausible relational domain. Foursquare is an example of a real system that could produce this sort of data. Suppose a researcher is interested in using data from this domain to assess whether smoking causes a user to gain weight. One approach would be to construct a propensity score model with user attributes that the researcher believes could be causes of whether a user smokes and the user's weight, such as alcohol consumption and ethnicity:

$$[User].Smokes \sim [User].Drinks + [User].Ethnicity.$$

While this accounts for attributes associated with the user, it fails to account for possible confounders derived from relational variables. For example, it is plausible that the alcohol consumption of a user's friends is a common cause of $[User].Weight$ and $[User].Smokes$. To account for these effects, the corresponding relational variables should be included in the propensity score model.

It is not difficult to envision more complicated relational variables having an effect. In fact, as previous work has shown [Maier et al., 2013b], the number of relational variables can be arbitrarily large depending on how many entity and relationship types exist in the network, the size of the network, and the length of the longest path (the largest degree of separation) in the network where direct dependence exists.

An additional level of complexity introduced by relational data is that relational structures may result in multiple instances of a given variable. For example, a user with multiple friends could be influenced by the drinking behaviour of each of those friends. Typically, an aggregation function, such as `mean`, is used to combine this set of values into a single value. Properly conditioning on a relational variable entails choosing the correct set of aggregation functions to represent the distribution of values contained in the set. For example, in order to condition on a relational variable, it may be necessary to condition on multiple aspects of the distribution of those values, such as the `mean` and the standard deviation (`stdev`).

To address these issues, we introduce relational propensity score matching (RPSM), a method that applies propensity score matching to relational domains. RPSM leverages the framework of relational models [Getoor and Taskar, 2007, Maier et al., 2013b] to automatically construct the set of possible relational confounders given a simpler specification of the assumed dependency structure. RPSM also identifies opportunities to use relational degree variables and entity identifiers, which, as we show empirically, can reduce the bias arising from latent relational confounders. We evaluate RPSM via a set of synthetic experiments using the relational structure of a real-world relational domain, Foursquare.

2 BACKGROUND

In this section we provide a brief overview of matching methods and propensity scores. We then introduce the relational concepts necessary to formalize RPSM.

2.1 MATCHING

In the framework of potential outcomes [Rubin, 1974], estimating the causal effect of treatment T on variable Y is formalized as a comparison of potential outcomes. More formally, let T_i be a binary treatment variable for unit i and let Y_i be the outcome variable for unit i, where $i \in \{1, \ldots, n\}$. $Y_i(T_i = 0)$ denotes the value of Y_i that would be observed if no treatment was applied to unit i. Similarly, $Y_i(T_i = 1)$ is the value of Y_i that would be observed if unit i had received treatment. The causal effect of T on Y is estimated by comparing the difference $Y_i(T_i = 1) - Y_i(T_i = 0)$ across all units i.

In practice, a specific unit either receives treatment or not. Therefore, for a given value of i we never know both $Y_i(T = 1)$ and $Y_i(T = 0)$. Experimental studies often randomly assign units to treatment and control groups, so that the expected distribution of the covariates in these groups is identical. In observational studies, where randomization is not possible, *matching* can be used to pair similar samples from the treated and the control groups. Matching can be generally defined as a method that aims to approximate random assignment by equating the distribution of covariates in the treated and control group [Stuart, 2010].

Matching requires a measure quantifying how similar two individuals are. This is achieved by (1) selecting a set of features to be used in the computation of similarity, and (2) choosing a similarity function to apply on those features (for example Mahalanobis distance, propensity score, etc.). Once a similarity measure has been chosen, individuals are matched based on this measure. There are multiple methods for performing matching (see Stuart [2010] and Ho et al. [2007] for a survey of matching methods). In this paper, we employ full matching [Hansen and Klopfer, 2006], which creates a collection of matched sets (the size of the collection is chosen automatically). Each matched set contains at least one treated and one control unit. Full matching has been shown to be optimal with respect to similarity within matched sets [Rosenbaum, 1991].

Matching methods make the assumption of *ignorable treatment assignment*, i.e., treatment assignment is independent of the outcome given the observed covariates. This assumption guides the selection of appropriate covariates for the computation of similarity.

2.2 PROPENSITY SCORE

The propensity score [Rosenbaum and Rubin, 1983] is the probability of receiving treatment, given the observed covariates $\mathbf{X}_i$

$$e_i(\mathbf{X}_i) = P(T_i = 1 | \mathbf{X}_i).$$

Propensity scores are a form of dimensionality reduction that projects the original covariates down to a single value which preserves distance with respect to the likelihood of treatment. Matching can then be performed on the propensity score, as opposed to the covariates directly. The prevailing explanation for why propensity scores are appropriate for matching is that they are balancing scores (given the value of the propensity score, the treatment and control groups have the same distribution of covariates), and they preserve ignorability of treatment assignment (if treatment assignment is ignorable given the covariates, then treatment assignment is also ignorable given the propensity score) [Stuart, 2010].

Any method that models the conditional probability of a binary variable given a set of predictors can be used to estimate a propensity score. In this work, we employ logistic regression, a widely used method for obtaining a propensity score. However, other models (such as boosted trees, support vector machines,

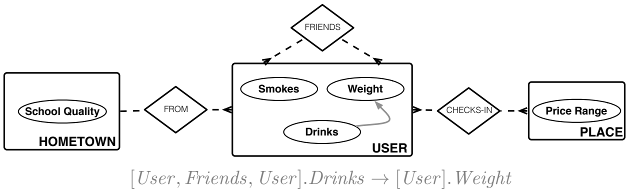

Figure 2: Relational model for the Foursquare domain. The underlying relational schema (ER diagram) is shown in black. The attributes on the entities are fictional. The relational dependency is shown in gray.

and neural networks) have been explored in the literature [Westreich et al., 2010, McCaffrey et al., 2004, Lee et al., 2010].

A key advantage of propensity scores is their robustness to model misspecification [Drake, 1993], i.e., including irrelevant variables[1] in the calculation of the propensity score. Because the propensity score model is built upon a *predictive* rather than causal model of treatment, many of the issues that arise with traditional regression modeling, such as multicollinearity, are no longer a threat to validity. Further, in contrast to matching directly on the covariates, propensity scores can down-weight or disregard variables that are not associated with treatment and have been erroneously included in the propensity model. However, as Pearl [2009] has observed, common effects of the treatment and outcome must not be included in the propensity score model. In general, the set of *d*-connecting paths between treatment and outcome needs to be considered. The propensity score model must include a (not necessarily minimal) separating set of treatment and outcome. One approach to eliminating variables that are potential common effects of treatment and outcome is the injunction of Rosenbaum and Rubin [1983] to restrict the set of covariates to pretreatment variables (variables whose values are measured prior to treatment).

2.3 RELATIONAL CONCEPTS

Propositional representations, such as Bayesian networks, describe domains with a single entity type. However, many real-world systems involve multiple types of entities that interact with each other. Data produced by such systems are called *relational* or *network data*. In this section, we introduce the basic relational concepts, following the notation and terminology of Maier et al. [2013b].

A *relational schema* $\mathcal{S} = (\mathcal{E}, \mathcal{R}, \mathcal{A}, card)$ specifies the set of entity, relationship, and attribute classes of a domain. It includes a cardinality function that imposes constraints on the number of times an entity instance can participate in a relationship. A relational schema can be graphically represented with an Entity-Relationship (ER) diagram. Figure 2 shows the ER diagram for the Foursquare domain. In this example, there are three entity classes (*User*, *Place*, *Hometown*), and three relationship classes, (*Friends*, *CheckssIn*, *From*). The entity class *User* has three attributes: *Smokes*, *Weight*, and *Drinks*. The cardinality constraints are depicted using crow's feet notation. For example, the cardinality of the *From* relationship is one-to-many, indicating that one user has one hometown, but many users can be from the same hometown.

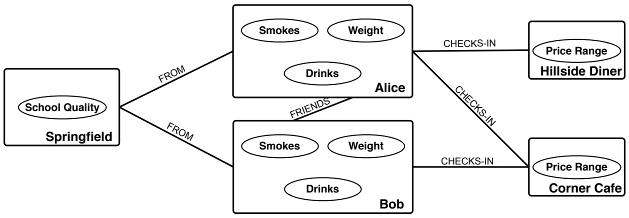

Figure 3: Example relational skeleton for the Foursquare domain. This could be a small fragment of a (potentially) larger skeleton.

A *relational skeleton* is a partial instantiation of a relational schema that specifies the set of entity and relationship instances that exist in the domain. Figure 3 depicts an example relational skeleton for the Foursquare domain. The network consists of two *User* instances, Alice and Bob, who are friends with each other and come from the same hometown. There are two *Place* instances, Hillside Diner and Corner Cafe.

Given a relational schema, one can specify *relational paths*, which intuitively correspond to possible ways of traversing the schema (see Maier et al. [2013b] for a formal definition). For the schema shown in Figure 2, possible paths include $[User, Friends, User]$ (a person's friends), and $[User, Friends, User, From, Hometown]$ (the hometowns of a person's friends). *Relational variables* consist of a relational path and an attribute that can be reached through that path. For example, the relational variable $[User, Friends, User].Drinks$ corresponds to the alcohol consumption of a person's friends. Probabilistic dependencies can be defined between relational variables. In this work, we consider dependencies where the path of the outcome relational variable is a single item. In this case, the path of the treatment relational variable describes how dependence is induced. For example, the *relational dependency*

$$[User, Friends, User].Drinks \rightarrow [User].Weight$$

[1] Variables that are marginally independent of treatment or outcome.

states that the alcohol consumption of a user's friends affects that user's weight.

A *relational model* $\mathcal{M} = (\mathcal{S}, \mathcal{D}, \Theta)$ is a collection of relational dependencies $\mathcal{D}$ defined over a relational schema along with their parameterizations Θ (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 2, and labeling each arrow with the dependency it corresponds to. If labels are omitted, the resulting graphical representation is known as a *class-dependency graph*.

Recent work by Maier et al. [2013b] provides a framework that enables reasoning about *d*-separation in relational models. Toward that end, they introduce *abstract ground graphs* (AGGs), a graphical structure that captures relational dependencies and can be used to answer relational *d*-separation queries. Abstract ground graphs are defined from a given perspective, the base item of the analysis, and include nodes that correspond to relational variables. For practical applications, the size of abstract ground graphs is limited by a (domain dependent) hop-threshold, which constrains the length of relational paths that will be considered. Intuitively, the hop-threshold corresponds to the relational "distance" of a cause from its effect.

2.4 NEW TYPES OF VARIABLES

In this section we present the new types of variables that are enabled by relational domains: (1) Relational variables (a way of defining a larger number of potential confounders) and aggregation; (2) Degree variables (a type of confounder not available without relational data); (3) Entity identifiers (which enable blocking, a way to account for latent confounders only available within relational data). Those types of variables are used in the calculation of relational propensity scores and are referred to as *relational covariates*.

2.4.1 Aggregation Functions

A fundamental characteristic of relational data is the heterogeneity of the underlying relational structure. For example, a person can have many friends, different people have different sets of friends, and those sets can overlap to varying degrees. This implies that when constructing relational variables for a specific individual, the construction process will often return a set of values rather than a single value. For instance, the relational variable "friends' age" for a person consists of a set of values containing the age of each one of that person's friends. In the field of statistical relational learning, aggregation functions are commonly used to summarize the values of related instances into a single value, representative of the distribution. Common aggregation functions include `mean`, `stdev`, `mode`, `count`, `sum`, `min`, `max`, and `median`. Researchers have also defined more complex aggregation methods [Perlich and Provost, 2006].

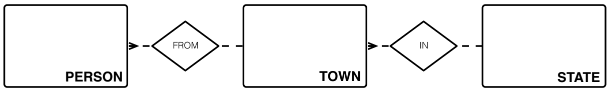

Figure 4: Relational schema that depicts a hierarchy. A state has many towns, but each town is in one state, and many people are from the same town, but each person is from one town.

2.4.2 Degree Variables

Other work has pointed out that variation in the size of the set of values for a relational variable can strongly affect the distribution of the observed values of many aggregation functions [Jensen et al., 2003]. Jensen et al. call the size of this set the "degree" and it is equivalent, in the terminology of Maier et al. [2013b], to the size of the terminal set of a relational path. To account for the effects of degree on aggregated values, RPSM includes degree variables in the calculation of propensity scores.

2.4.3 Entity Identifiers

Blocking designs are widely used in experimental studies to account for latent confounders [Fisher, 1935]. Rattigan et al. [2011] formalized *relational blocking* as an operator that can be used to infer causal dependence in observational data expressed in a relational representation. By blocking on the identifier of an entity, relational blocking accounts for the effect of latent variables associated with that entity. Blocking is uniquely available for relational data. Moreover, since blocking on an entity appears to avoid inducing dependence due to colliders on that entity, blocking may partially alleviate a key threat to validity noted by Pearl [2009].

In this work, we incorporate relational blocking with propensity scores by including entity identifiers as covariates in the calculation of propensity scores. We restrict the use of blocking to hierarchies, i.e., parts of the relational schema that are connected through a series of many-to-one relationships. An example hierarchy is shown in Figure 4. In this case, blocking on the identifier for towns (i.e., grouping users based on their hometown) accounts for the effect of latent variables associated with *Hometown*, and for the effect of latent variables associated with the *State* within which each town is located. More generally, blocking on the identifier of an entity in a hierarchy accounts for the effect of latent confounders that reside in that entity and in entities that appear higher up in the hierarchy.

3 RELATIONAL PROPENSITY SCORE MATCHING

We consider the following problem: given an entity E and two attributes on that entity, treatment $[E].T$ and outcome $[E].O$, we seek to decide between $[E].T \to [E].O$ and $[E].T \not\to [E].O$. For notational convenience, we restrict our attention to cases where the treatment and outcome are on the same entity. In practice, RPSM can be applied to any treatment and outcome lying on entities that are connected through one-to-one relationships. We assume that the relational skeleton has been given *a priori*, i.e., all entity and relationship instances have been fully and correctly specified. Additionally, we assume that the effects of all latent variables can be accounted for by using relational blocking (in other words, latent variables exists only on paths that can be blocked on).

Relational propensity score matching (RPSM) provides an automatic method for constructing the set of aggregated relational variables, degree variables and entity identifiers (i.e., the *relational covariates*) to perform propensity score matching on relational data. The procedure for RPSM is described in Algorithm 1. RPSM takes as input a data-set $\mathcal{X}$, a relational schema, the treatment and outcome attributes, a set of possible confounding attributes, a set of aggregation functions, and a hop-threshold h. The algorithm constructs the set of relational covariates based on the confounding attributes, the aggregation functions, and hop-threshold (line 2, discussed below in detail). The propensity score of the *treatment* given the *covariates* is then computed (line 3) and matching is performed based on the propensity score (line 4).

The construction of relational covariates is presented in Algorithm 2. The algorithm first constructs all potential relational variables for the confounding attributes from the given perspective, up to the specified hop-threshold (line 1).[2] This is the set of *relational confounders*. Then, for each relational confounder, it creates the appropriate relational covariates by applying the given aggregation functions (lines 7-8). A degree variable is then added for the paths of the relational confounders (line 9). Finally, the algorithm identifies parts of the schema that form a hierarchy and adds identifier variables for the schema item lowest in the hierarchy to perform blocking (lines 10-14). Relational covariates that were constructed from relational variables that are now determined by the blocking path are removed from the list of covariates (line 13).

Algorithm 1: `RPSM` ($\mathcal{X}$, *schema*, *treatment*, *outcome*, *confoundingAttrs*, *aggrFunctions*, h)

```
1  $perspective \leftarrow$ item class of $treatment$, $outcome$
2  $covariates \leftarrow$ GetRelationalCovariates ($schema$,
   $perspective$, $confoundingAttrs$, $aggrFunctions$, $h$)
3  $propensityScore \leftarrow$ Calculate propensity score for
   $treatment \sim covariates$ using $\mathcal{X}$
4  $matches \leftarrow$ Match ($propensityScore$, $treatment$, $\mathcal{X}$)
5  return $matches$
```

Algorithm 2: `GetRelationalCovariates` (*schema*, *perspective*, *confoundingAttrs*, *aggrFunctions*, h)

```
1  $relationalConfounders \leftarrow$ relational variables with
   attributes in $confoundingAttrs$ from perspective
   $perspective$ up to hop-threshold $h$
2  $relCovariates \leftarrow \emptyset$
3  for $P.X$ in $relationalConfounders$ do
4      if $P == [perspective]$ then
5          $relCovariates \leftarrow relCovariates \cup P.X$
6      else
7          for $agg$ in $aggrFunctions$ do
8              $relCovariates \leftarrow relCovariates \cup agg(P.X)$
9          $relCovariates \leftarrow relCovariates \cup degree(P)$
10 for $P.X$ in $relationalConfounders$ do
11     if $P$ is valid blocking choice for perspective then
12         $controlled \leftarrow$ relational variables that $P$
           controls for
13         $relCovariates \leftarrow relCovariates \setminus controlled$
14         $relCovariates \leftarrow relCovariates \cup P.id$
15 return $relCovariates$
```

Example 3.1. Consider our earlier scenario of assessing the effect of smoking on a user's weight. The treatment is *User.Smokes* and the outcome is *User.Weight* (the perspective of the analysis is the *User* entity class). If *Drinks* is given as a possible confounding attribute and the hop-threshold is 4, the algorithm will add the following relational variables to the set of relational confounders:

$$[User].Drinks$$
$$[User, Friends, User].Drinks$$
$$[User, Friends, User, Friends, User].Drinks$$
$$[User, CheckIn, Place, CheckIn, User].Drinks$$
$$[User, From, Hometown, From, User].Drinks$$

The next step is to create relational covariates based on the above relational variables. First, relational variables that only involve the *User* entity, in this case $[User].Drinks$, are added to the set of relational co-

[2] The algorithm can be trivially extended to exclude certain relational paths. For example, if the user has domain knowledge that would exclude specific relational paths or relational variables from the list of potential confounders.

variates. Because these covariates are propositional, aggregation functions are not applied.

The aggregation functions are then applied to relational variables that cross the boundaries of the *User* entity. If the set of aggregation functions is {`mean`}, the algorithm will add the following to the set of relational covariates:

$$\begin{aligned}
&\texttt{mean}\big([\mathit{User}, \mathit{Friends}, \mathit{User}].\mathit{Drinks}\big),\\
&\texttt{mean}\big([\mathit{User}, \mathit{Friends}, \mathit{User}, \mathit{Friends}, \mathit{User}].\mathit{Drinks}\big),\\
&\texttt{mean}\big([\mathit{User}, \mathit{ChecksIn}, \mathit{Place}, \mathit{ChecksIn}, \mathit{User}].\mathit{Drinks}\big),\\
&\texttt{mean}\big([\mathit{User}, \mathit{From}, \mathit{Hometown}, \mathit{From}, \mathit{User}].\mathit{Drinks}\big)
\end{aligned}$$

The set of relational covariates is augmented by including the degree of the relational paths that involve more than one entity classes:

$$\begin{aligned}
&\texttt{degree}\big([\mathit{User}, \mathit{Friends}, \mathit{User}]\big),\\
&\texttt{degree}\big([\mathit{User}, \mathit{Friends}, \mathit{User}, \mathit{Friends}, \mathit{User}]\big),\\
&\texttt{degree}\big([\mathit{User}, \mathit{ChecksIn}, \mathit{Place}, \mathit{ChecksIn}, \mathit{User}]\big),\\
&\texttt{degree}\big([\mathit{User}, \mathit{From}, \mathit{Hometown}, \mathit{From}, \mathit{User}]\big)
\end{aligned}$$

Finally, *id* variables are added to the relational paths. In this case, there exists a hierarchy expressed by the relational path $[\mathit{User}, \mathit{From}, \mathit{Hometown}]$. Therefore, the algorithm adds the following relational covariate:

$$[\mathit{User}, \mathit{From}, \mathit{Hometown}].\mathit{id}$$

In practice, the hop-threshold should be chosen on a case by case basis, using expert knowledge of the application domain. The choice of aggregation functions can be guided by an analysis of each variable's marginal distribution from the perspective of the treatment and outcome.

4 SYNTHETIC EXPERIMENTS

To evaluate the performance of RPSM we examine the following hypotheses:

1. Propensity score matching models that are limited to simplistic relational attributes ($h = 2$) fail to fully account for confounding network effects ($h = 4$) (Section 4.1).
2. Traditional aggregates for relational data, such as `mean`, when used in isolation do not sufficiently condition on the *distribution* of confounding relational variables (Section 4.2).
3. The inclusion of identifiers for entities that lie along valid blocking paths accounts for latent confounders on those entities as well on entities connected to them. That is, including entity identifiers in the propensity model performs an implicit causal blocking design (Section 4.3).

For all experiments we used the structure derived from a sample of a real-world network, Foursquare [Gao et al., 2012], augmented with synthetic attributes on the entities. This allows for controlling the dependencies between attributes as well as the marginal and conditional distributions, while leveraging relationships from a real-network. The relational schema for the Foursquare network is shown in Figure 2. The relational skeleton consists of 9,599 users, 47,164 friendships, 182,968 locations where users "checked-in" via the mobile application, 1,360,123 check-ins, and the users' hometowns. Aggregate statistics for the network are shown in Table 1.

Table 1: Descriptive statistics for the Foursquare relational skeleton used in the synthetic experiments.

Aggregate	Friends	Check-Ins
mean	9.45	120.09
median	5	73
min	1	1
max	3674	2477

For our experiments we generated data from multiple models to test each hypothesis individually. In all experiments, the treatment is $[\mathit{User}].\mathit{Smokes}$ and the outcome $[\mathit{User}].\mathit{Weight}$. Each model was parameterized as follows: The value of the treatment was drawn from a logistic model parametrized using coefficients drawn from $\mathcal{U}(-2, 2)$ and interaction terms increasing in degree from 1 (no interaction) to 10 (up to 10 interacting covariates, not necessarily distinct, per term). We refer to this varying degree as "covariate complexity". The value of outcome was drawn from a linear model with coefficients drawn from $\mathcal{U}(-2, 2)$ and an error distribution drawn from $\mathcal{N}(0, 1)$. Marginal distributions for each variable were drawn from $\mathcal{N}(\mu, \sigma)$, with μ and σ sampled for each variable individually from $\mathcal{U}(0, 5)$ and $\mathcal{U}(1, 3)$, respectively.

We used logistic regression to calculate the propensity score and then performed full matching using the `optmatch` package [Hansen and Klopfer, 2006]. A linear model was applied using treatment and matching assignment as covariates and outcome as the response variable to assess statistical significance, with an α value of 0.01 for determining dependence. In this setting, we would expect a low error rate for linear log-odds functions (covariate complexity is 1), given the perfect correspondence between the generating models and the estimation methods when the set of covariates is correctly specified (no interaction terms). Adding interaction terms renders the models progressively less appropriate. We report Type I and Type II errors. Type I error corresponds to cases where a valid causal dependence exists between treatment and outcome and RPSM incorrectly concludes that there exists no such dependence. Type II error corresponds

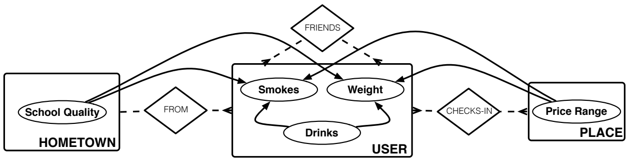

Figure 5: Class-dependency graph for the models used to evaluate the effect of using relational variables with longer hop-thresholds as covariates.

to cases where RPSM incorrectly concludes that there exists a dependency between treatment and outcome.

4.1 SIMPLE NETWORK DEPENDENCIES

We examine the first hypothesis, whether propensity score models limited to simplistic relational attributes fail to fully account for confounding network effects, by generating data from two models. Both models have the same class-dependency graph, shown in Figure 5, but differ in the length of the longest true dependency.

In the first model (World2), the true relational confounders are at most two hops away from the treatment and outcome entity. This corresponds to dependencies that can be read directly from the class dependency graph, e.g., the places a user checks in to. The set of true relational confounders for the model is:

$$[User].Drinks$$
$$[User, From, Hometown].SchoolQuality$$
$$[User, ChecksIn, Place].PriceRange$$

In the second model (World4), the set of true confounders is extended to include relational variables up to four hops away, e.g., other users that check in to the same places as a user. The set of confounders includes all of the confounders of the first model as well as:

$$[User, Friends, User].Drinks$$
$$[User, ChecksIn, Place, ChecksIn, User].Drinks$$
$$[User, From, Hometown, From, User].Drinks$$
$$[User, Friends, User, Friends, User].Drinks$$
$$[User, Friends, User, ChecksIn, Place].PriceRange$$
$$[User, Friends, User, From, Hometown].SchoolQuality$$

Using the above procedure we ran 100 trials. For each trial we considered two cases, one in which treatment and outcome are conditionally independent and one in which there is a direct effect between them. We then compared two methods for creating the relational covariates for propensity score matching:

1. RPSM using `mean`, `stdev`, `max`, `min` as aggregation functions and $h = 2$ without blocking or degree variables (RPSM2)
2. RPSM using `mean`, `stdev`, `max`, `min` as aggregation functions and $h = 4$ without blocking or degree variables (RPSM4)

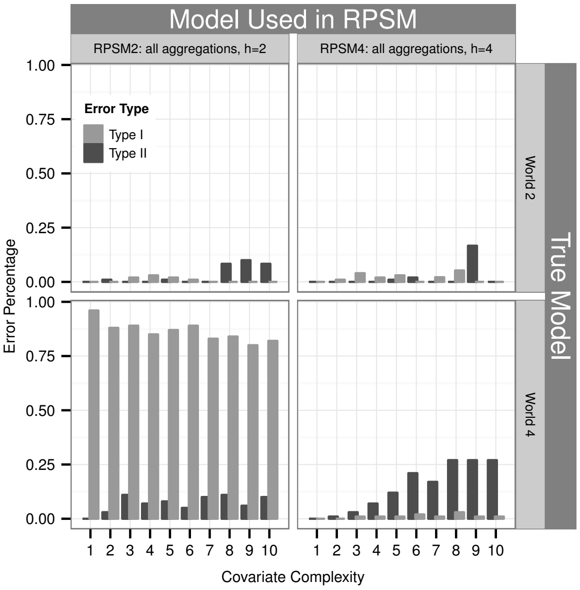

Figure 6: Percentage of Type I and II error when RPSM2 and RPSM4 are applied to data generated by World2 and World4 models with increasing covariate complexity, averaged over 100 trials.

The results are shown in Figure 6. Along the diagonal the RPSM model is consistent with the world configuration. When models are over-specified, for instance RPSM4 in the World2 configuration, RPSM4 achieves comparable performance to RPSM2. However, when models are underspecified, for instance RPSM2 in the World 4 configuration, a spurious effect is inferred between treatment and outcome in the conditionally independent case. These results also demonstrate a case in which RPSM can successfully tolerate large numbers of irrelevant covariates.

4.2 COMPLEX NETWORK DEPENDENCIES

In this section, we examine the second hypothesis regarding the effect of using complex aggregation function in the construction of relational covariates. We generated data from models with the same class-dependency graph as in Section 4.1. We used World2 and World4, as before, and two simplified models which consider *only* `mean` as an aggregate, with hop-thresholds of 2 (World2-) and 4 (World4-). We then used the RPSM2 and RPSM4 methods for constructing relational covariates and two simpler propensity score models that only include `mean` as an aggregate

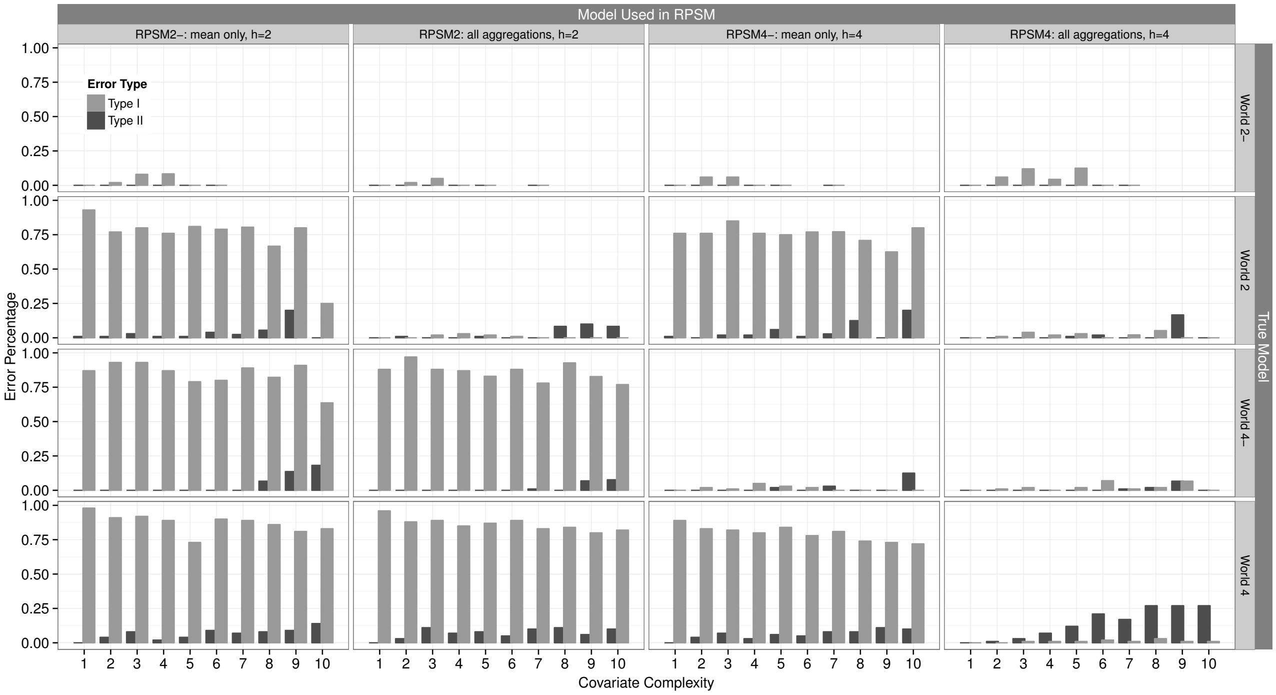

Figure 7: Type I and Type II error when RPSM2-, RPSM2, RPSM4-, and RPSM4 are applied to data generated by World2-, World2, World4-, and World4 models with increasing covariate complexity, averaged over 100 trials.

(RPSM2- with $h=2$ and RPSM4- with $h=4$).

The results are shown in Figure 7. Along the main diagonal, the assumptions of the RPSM model are consistent with the true world configuration. In cases where the employed model uses `mean` as the sole aggregation function but distributional dependencies are more complex, RPSM commits a large number of Type I errors. However, the over-specified models (e.g., RPSM4 in World2) maintain accuracy levels that are consistent with the most efficient RPSM configuration.

4.3 ENTITY IDENTIFIERS

The final experiment examines the third hypothesis regarding the effect of including entity identifiers in the relational propensity score model. We generated data from a model similar to that of Figure 5, with an additional latent confounder on the *Hometown* entity. We then created relational covariates using four strategies:

1. Use all observed variables and hop-threshold of 2 (RPSM2) and 4 (RPSM4).
2. Use degree variables and entity-identifiers for all eligible blocking paths with either $h = 2$ or $h = 4$ (RPSM2+ and RPSM4+ respectively).

The results are shown in Figure 8. RPSM2 and RPSM4 perform poorly, because of the bias induced by unconditioned confounders. RPSM2+ performs well when true relational dependencies are limited to $h = 2$. RPSM4+ performs well in all cases. This is an indication that including the entity identifiers in the propensity model performs blocking, producing effects similar to the explicit conditioning performed by Rattigan et al. [2011]. This also strengthens the connection between relational blocking and a conjecture made by Perlich and Provost [2006] that the inclusion of identifier variables in a non-causal setting can be used to create a relational fixed or random effects model. Given these results, the ability to automatically identify and utilize entity identifiers provides a strong argument for using RPSM as opposed to a propositional approach. While blocking accounts for a relatively small subset of all possible confounders, it provides a substantial improvement over the alternative of assuming no latent confounders.

5 RELATED WORK

Multi-level propensity score models [Hong and Raudenbush, 2006, Li et al., 2013] provide a method for accounting for group or cluster level effects. This corresponds to a one-to-many relationship in a relational schema. RPSM can be seen as an extension of the multi-level setting, capturing not only one-to-many group level effects, but also many-to-many effects. There has also been significant progress in un-

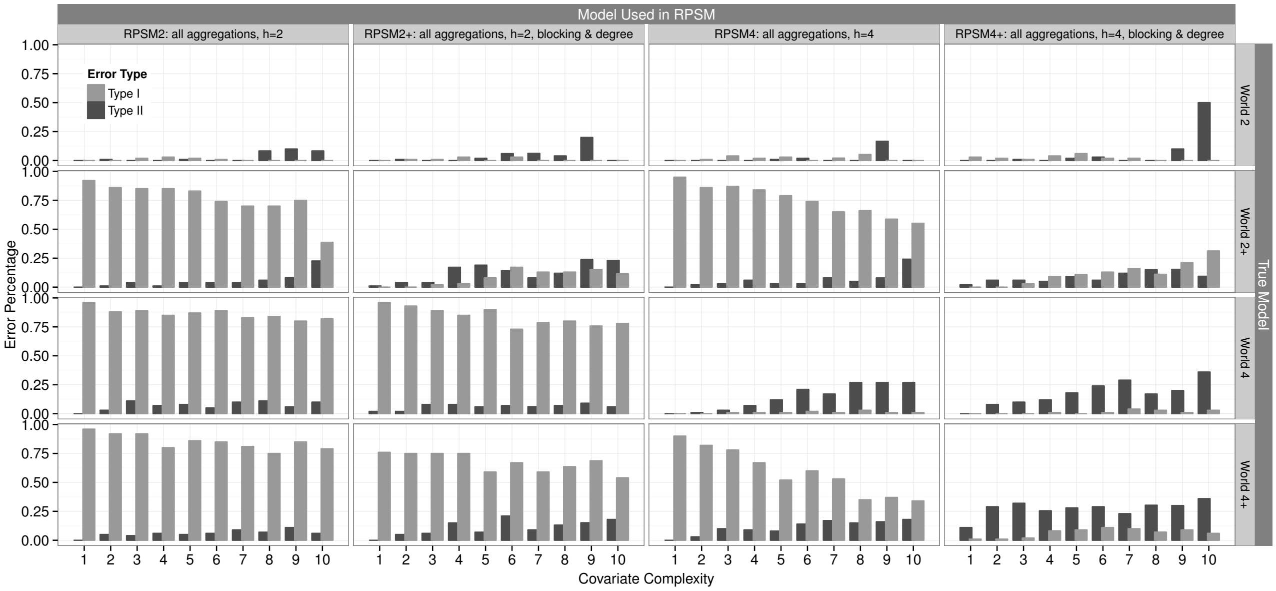

Figure 8: Type I and Type II error when RPSM2, RPSM2+, RPSM4, and RPSM4+ are applied to data generated by World2, World2+, World4, and World4+ with increasing covariate complexity, averaged over 100 trials.

derstanding the consequences of the stable unit treatment value assumption (SUTVA)[3] for matching and propensity models in the fields of statistics, epidemiology and econometrics [Hudgens and Halloran, 2008, Tchetgen and VanderWeele, 2012, Manski, 2013]. This work does not address SUTVA violations, but extensions to that setting are a focus of future work.

Perlich and Provost [2006] introduced relational fixed and random effects models using identifier attributes as features in the ACORA framework. RPSM differs in two important aspects. First, the aim of the aforementioned work is predictive, rather than causal. Second, RPSM incorporates degree variables and provides an algorithm for deciding *which* relational variables should be included, rather than assuming the correct set of relational variables and aggregating.

In the area of relational causal discovery, Maier et al. [2013a] introduced a constraint-based algorithm, RCD, that leverages relational d-separation [Maier et al., 2013b] to learn causal models from relational data. RCD learns a joint causal model of a relational domain and abstracts away the mechanics of performing individual tests of conditional independence, while RPSM focuses on evaluating a single causal dependence and the conditioning mechanism.

6 FUTURE WORK

We plan on examining RPSM further, using more complex synthetic data and real-world data. An interesting avenue for future research is extending RPSM to the case where the treatment or outcome lies along a one-to-many relational path (e.g., the effect of a treatment performed on an individual on an aggregate attribute of the individual's friends). There are also a number of methods for performing matching without a propensity score, such as matching on the full set of covariates [Stuart, 2010], coarsened exact matching [Iacus et al., 2012], and entropy balancing [Hainmueller, 2012]. Extending these methods to the relational setting would allow practitioners flexibility in terms of the set of assumptions required for a given causal analysis.

7 CONCLUSIONS

Propensity score matching provides a powerful and robust method for causal inference on propositional data. However, naively applying PSM to relational data ignores both new challenges and opportunities presented by this richer type of data. RPSM automatically constructs the set of relational covariates to be used in the propensity score model given a set of confounding attributes, a set of aggregation functions, and a hop threshold. Further, it exploits the relational structure by identifying degree variables and entity identifiers, which can account for latent relational confounders. We evaluate its efficacy via synthetic experiments that leverage a real-world relational skeleton.

[3] SUTVA states that the outcome of an individual is independent of the treatment status of other individuals.

References

C. Drake. Effects of misspecification of the propensity score on estimators of treatment effect. *Biometrics*, 49(4):1231–1236, 1993.

R. A. Fisher. *The Design of Experiments*. Oliver and Boyd, Edinburgh, 1935.

H. Gao, J. Tang, and H. Liu. gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1582–1586. ACM, 2012.

L. Getoor and B. Taskar. *Introduction to statistical relational learning*. MIT press, 2007.

J. Hainmueller. Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political Analysis*, 20(1):25–46, 2012.

B. B. Hansen and S. O. Klopfer. Optimal full matching and related designs via network flows. *Journal of Computational and Graphical Statistics*, 15(3):609–627, 2006.

D. E. Ho, K. Imai, G. King, and E. A. Stuart. Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference. *Political Analysis*, 15(3):199–236, 2007.

G. Hong and S. W. Raudenbush. Evaluating kindergarten retention policy. *Journal of the American Statistical Association*, 101(475), 2006.

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

S. M. Iacus, G. King, and G. Porro. Causal inference without balance checking: Coarsened exact matching. *Political Analysis*, 20(1):1–24, 2012.

D. D. Jensen, J. Neville, and M. Hay. Avoiding bias when aggregating relational data with degree disparity. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 274–281. AAAI Press, 2003.

B. K. Lee, J. Lessler, and E. A. Stuart. Improving propensity score weighting using machine learning. *Statistics in Medicine*, 29(3):337–346, 2010.

F. Li, A. M. Zaslavsky, and M. B. Landrum. Propensity score weighting with multilevel data. *Statistics in Medicine*, 32(19):3373–3387, 2013.

M. Maier, K. Marazopoulou, D. Arbour, and D. 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, 2013a.

M. Maier, K. Marazopoulou, and D. Jensen. Reasoning about independence in probabilistic models of relational data. *arXiv preprint arXiv:1302.4381*, 2013b.

C. F. Manski. Identification of treatment response with social interactions. *The Econometrics Journal*, 16(1):S1–S23, 2013.

D. F. McCaffrey, G. Ridgeway, and A. R. Morral. Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological Methods*, 9(4):403, 2004.

J. Pearl. Remarks on the method of propensity score. *Statistics in Medicine*, 28(9):1415–1416, 2009.

C. Perlich and F. Provost. Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning*, 62(1-2):65–105, February 2006.

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

P. R. Rosenbaum. A characterization of optimal designs for observational studies. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3): 597–610, 1991.

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

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

E. A. Stuart. Matching methods for causal inference: A review and a look forward. *Statistical Science*, 25 (1):1–21, 2010.

E. J. T. Tchetgen and T. J. VanderWeele. On causal inference in the presence of interference. *Statistical Methods in Medical Research*, 21(1):55–75, 2012.

D. Westreich, J. Lessler, and M. J. Funk. Propensity score estimation: Neural networks, support vector machines, decision trees (cart), and meta-classifiers as alternatives to logistic regression. *Journal of Clinical Epidemiology*, 63(8):826–833, 2010.